Real-Time Implementation of Retinal Models

Garibaldi Pineda García
Supervisor: Steve Furber, Co-Supervisor: Dave Lester
School of Computer Science, University of Manchester, U.K.

***Abstract*—Vision systems in biological entities are among the most complex sensory inputs in nature. If we want to simulate them, it would require incredible amounts of computing power and, traditionally, different algorithms to perform each individual task. A parallel computation platform is the best way to go while attempting to solve this problem, since neural structures in the brain compute in this way.**

SpiNNaker is one of such platforms, a network of low-powered processing units, each of which can simulate several neurons. Given that the SpiNNaker platform resembles this natural neural structures, computer vision algorithms need to be developed in a completely different manner.

The aim of this project is to develop algorithms in the realm of computer vision but using a spiking neural networks approach. In particular we'll study time-based spike codes and how to process them. This algorithms should be able to cooperate and share their interpretation of the input data to gain a more robust understanding of images.

I. Introduction

In recent years neuromorphic (i.e. one that mimics the brain) hardware has risen attention as a different way of computing. One key aspect is the high parallelism found in the infrastructure of the brain. Platforms such as *SpiNNaker* [1] emulate such parallelism; furthermore it does so while maintaining low power consumption. The *SpiNNaker* platform can also give neural simulations the flexibility of software models and keep them running in biological real-time.

Converting conventional images or video into spike based representation is a must-do step for further studies, in section IV we report on the work done so far towards this goal. We use off-the-shelf hardware to encode video, in particular we use Graphics Processing Units (GPU) due to their parallel architecture.

In the final sections of this paper we present the plans we have to develop learning and classification algorithms. Furthermore this algorithms may lead to an implementation of vision tasks such as registration or optical flow.

II. Research Aims and Contribution

This research aims to develop computer vision algorithms using SpiNNaker. This is to be achieved by modelling biological vision, using spiking neural networks, on SpiNNaker. Several stages of vision would need modelling and/or implementation, the latter has been the goal for this year's work. We hypothesize that a better understanding of vision in biology will lead to a unified computer vision framework. Using neural networks should translate in gaining an insight to the meaning of elements in a scene and, thus, a relation between different images of the same scenario.

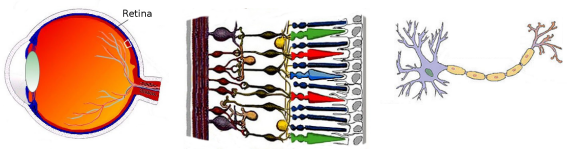
Bio-inspired vision algorithms using SpiNNaker hardware could be used on robotics, security or transportation applications. The research on learning and classification could lead into a theory of learning and memory in the brain.

III. Previous Work

In order to process visual input from frame based imaging devices on a spiking neural network (SNN) a transformation is needed. The most common way is to simply encode using Poisson spiking with a rate that is proportional to pixel intensity. This is just modelling the photoreceptor layer in the retina as other cell layers react to changes in intensity[2] and perform other computation before emitting actual spikes. One of the most accurate retinal models was developed by Wohrer and Kornprobst in [3]. A special category is hardware based bio-inspired retinas. First reported on [4]. New devices have been developed and reported in [5], [6], this are splendid real-time, low-powered, high-dynamic-range event-based cameras; though they have limited availability.

IV. Project Progress to Date

The literature review is advancing at a very good pace. Our objective this year is to generate a video-to-spike train encoder using of-the-shelf components. For this, the first approach was to use a biologically plausible functional model [7] that results in images being transformed into rank-ordered spikes [8].

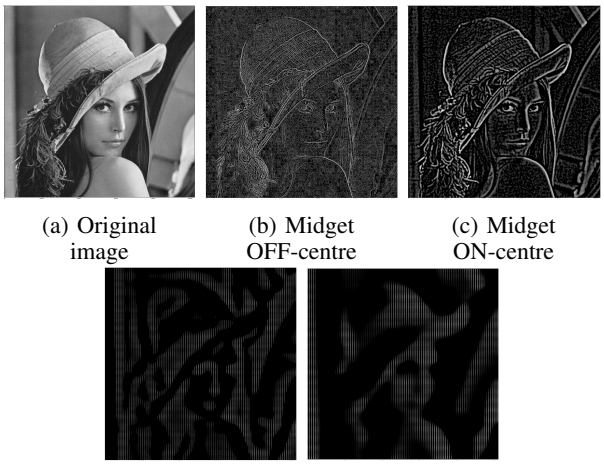


(a) Eye schematics (b) Retina (c) Neuron

Figure 1: Anatomy of the (human) eye

The retina is a thin layer of neural cells located in the eye (Fig. 1), it is responsible for the sensing, processing and transmitting visual input[2]. At its deepest layer, the retina has a millions of cells known as photoreceptors (right of Fig. 1b), they are in charge of transforming

light into electrical signals. After this step there are three layers of neurons that perform different computations such as lateral inhibition or on/off centre-off/on surround behaviour (left of Fig. 1b) [2], [7]. A small area at the centre of the retina has very few obstacles to obtain light and has high resolution, this area is known as the *foveal pit* (small depression on the right of Fig. 1a).



(a) Original image (b) Midget OFF-centre (c) Midget ON-centre

(d) Parasol OFF-centre (e) Parasol ON-centre

Figure 2: Results of simulating ganglion cells (convoluted images are enhanced for better contrast)

This algorithm models the foveal pit, first we perform a two-dimensional discrete *convolution* of the current frame with four *kernels* (Table I) which represent four types of ganglion cells; each cell type is modelled using a Difference of Gaussian (DoG, Eq. 1).

$$DoG_w(x,y) = \pm\frac{1}{2\pi\sigma_{w,c}^2}e^{\frac{-(x^2+y^2)}{2\sigma_{w,c}^2}} \mp \frac{1}{2\pi\sigma_{w,s}^2}e^{\frac{-(x^2+y^2)}{2\sigma_{w,s}^2}} \quad (1)$$

where $\sigma_{w,c}$ and $\sigma_{w,s}$ are the standard deviation for the centre and surround components of the DoG at scale w (cell type). The signs will be $(-,+)$ if the ganglion cell is OFF-centre and $(+,-)$ if it is ON-centre.
Every pixel in the convoluted images (Fig. 2) represent a spike emission time, the higher the pixel value, the sooner the spike will be sent out. Different ways of performing convolutions on a GPU where implemented, the naïve does a discrete convolution with the full kernels; for the biggest kernel (243×243 elements) we were unable to fit it into one of the fast memory locations of the GPU. We decompose the DoG into two horizontal and two vertical convolution kernels to perform a separated convolution. This method works best on kernels bigger than 3×3. Last approach, *Tiled Convolution* is reported by Advanced Micro Devices (AMD) in [9]. They only do kernels of size 3×3, but we have an 11×11 convolution working; we are still developing solutions for the larger kernels. Convolution alone is a compute intensive task and we obtain about 12 frames-per-second (FPS) on videos with 640×360 8-bit grayscale pixel resolution. Encoding was carried out using a desktop computer running 64-bit GNU/Linux, with a Core i5-4570 4-core CPU @ 3.20GHz processor with 8 GBytes of 64-bit DDR3 RAM @ 1600MHz and a GeForce GT 720 GPU with 192 CUDA cores @ 797 MHz, 1 GBytes of 64-bit DDR3 RAM @ 1800 MHz.

Table I: Simulation parameters for ganglion cells

Cell type	Matrix width	Centre std. dev. (σ_c)	Surround std. dev. (σ_s)	Sampling resolution
Midget Off-centre	3	0.8	$6.7 \times \sigma_c$	columns: 1, rows: 1
Midget On-centre	11	1.04	$6.7 \times \sigma_c$	columns: 1, rows: 1
Parasol Off-centre	61	8	$4.8 \times \sigma_c$	columns: 5, rows: 3
Parasol On-centre	243	10.4	$4.8 \times \sigma_c$	columns: 5, rows: 3

Table II: Performance comparison.

	Midget Off-centre	Midget On-centre	Parasol Off-centre	Parasol On-centre
Naïve	0.0009s	0.0031s	0.0587s	N/A[1]
Separated	0.0029s	0.0055s	0.0172s	0.0472s
Tiled	0.0019s	0.0027s	N/A[2]	N/A[2]

[1] Unable to fit convolution kernel into constant memory.
[2] Coding optimizations are still in progress.

(a) Original image (b) 100% of raw spikes (c) 100% of *corrected* spikes

Figure 3: Results of reconstruction procedure

In the retina, redundancy of information is reduced via lateral inhibition prior to any ganglion cell activity. In this algorithm, we perform a correction is on the convolved images by adjusting the convoluted image's weights according to the correlation between convolution kernels. The results of using correction (Fig. 3b) or not (Fig. 3c) show that the convolution stage can only provide redundant information. Furthermore using only 30% of the corrected weights, enough visual information is transmitted to reconstruct the original image [7].

Correcting the spikes for redundancy is a highly time consuming task which might be better suited for event-based programming, such as the one found on the SpiNNaker platform. We are still working on an implementation for this approach.

A second way of encoding is to simulate the early stages of the retina, which sense changes in intensity on the photoreceptors. This is quite similar to what real

Dynamic Vison Sensors (DVS, [5], [6]) do but with limited dynamic range and lower temporal resolution. The main advantage is that no specialized hardware is needed and the operation is so fast that any recent computer should be able to do it. For this type of encoding procedure we hypothesize that the bigger the change, the sooner a cell would spike and, thus, we can obtain a spike timings given the difference of two video frames. So far we can process about 20 and 25 FPS using a Numpy and an OpenCL backend, respectively (using the same hardware set-up previously described). Although it's currently a good approximation, more research on this algorithm is needed to better approximate to biology.

V. Thesis Outline

- Abstract
- **Chapter 1**. Introduction.
  - Neural networks.
  - Spike codes in vision.
  - Inhibition.
  - Spatio-temporal patterns and learning.
  - Research objectives.
- **Chapter 2**. Background.
  - SpiNNaker platform.
  - Real-time artificial neural computations.
  - Polychronization.
  - Classification.
- **Chapter 3**. Methodology.
  - Model visual input using time-based spike codes.
  - Hierarchical networks for robust classification.
  - Feature identification.
  - Sensor fusion and image registration.
- **Chapter 4**. Results.
  - Comparison with other methods.
  - Discussion.
- **Chapter 5**. Conclusions and Further Work.
  - Conclusions.
  - Future work.
  - Publications.
- **References**.

VI. Conclusions and further work

A. Conclusions

We obtained further knowledge about the anatomy of the eye from both a detailed and a functional approach. We also have come to appreciate the importance of inhibition circuits to enable high-efficiency, low-redundancy computing in the brain and how this brings robustness to neural structure.

Real-time spike encoding, although with low temporal resolution, is achievable with a common GPU and the right combination of mathematics and engineering. Memory reads and writes in a GPU are extremely important, it is one of the biggest bottlenecks of the presented algorithms. If each convolution was to be performed by a different GPU, we expect to see much better performance, though testing is needed on this regard. We are planning on developing an FPGA based solution to this problem.

We propose timing mechanisms to emit spikes from a rank-ordered source. Possible solutions for a faster mutual inhibition algorithm might be to do it in-line as we send spikes to neuromorphic hardware; or let the neural simulation deal with mutual inhibition.

B. Future work

We plan to explore learning on spiking neural networks using time-based codes or rank-ordered ones. This is an area where work has been made but remains an open problem. After a learning mechanism is proposed, we will use this to design networks that are able to robustly classify the MNIST dataset we previously encoded (section IV). Once some understanding of how features are interpreted in the network is gained, we shall proceed to make use of this knowledge to derive vision algorithms. In some cases we will use multiple sensors, so we'll have to apply techniques like sensor fusion.

VII. Publications

Part of the work carried during this year will be published as a paper on a **Frontiers in Neuroscience** journal in a special issue *Benchmarks and Challenges for Neuromorphic Engineering*. The article will present the MNIST database encoded using different types of spike codes and propose it as a standard way of testing learning and recognition tasks.

Acknowledgements

This research is funded by the National Council of Science and Technology (CONACyT) and the Secretariat of Public Education (SEP) of México.

References

[1] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the spinnaker system architecture", *Computers, IEEE Transactions on*, vol. 62, no. 12, pp. 2454–2467, 2013.

[2] H. Kolb. (2015). Simple anatomy of the retina. M. E. Center, Ed., Webvision, [Online]. Available: http://webvision.med.utah.edu/book/part-i-foundations/simple-anatomy-of-the-retina/.

[3] A. Wohrer and P. Kornprobst, "Virtual retina: a biological retina model and simulator, with contrast gain control", English, *Journal of Computational Neuroscience*, vol. 26, no. 2, pp. 219–249, 2009, ISSN: 0929-5313. DOI: 10.1007/s10827-008-0108-4.

[4] C. A. Mead and M. Mahowald, "A silicon model of early visual processing", *Neural Networks*, vol. 1, no. 1, pp. 91 –97, 1988, ISSN: 0893-6080. DOI: 10.1016/0893-6080(88)90024-X.

[5] J. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A five-decade dynamic-range ambient-light-independent calibrated signed-spatial-contrast aer retina with 0.1-ms latency and optional time-to-first-spike mode", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 10, pp. 2632–2643, 2010, ISSN: 1549-8328. DOI: 10.1109/TCSI.2010.2046971.

[6] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120 db 15 us latency asynchronous temporal contrast vision sensor", *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 2, pp. 566–576, 2008, ISSN: 0018-9200. DOI: 10.1109/JSSC.2007.914337.

[7] B. Sen and S. Furber, "Evaluating rank-order code performance using a biologically-derived retinal model", in *Proceedings of the 2009 International Joint Conference on Neural Networks*, ser. IJCNN'09, Atlanta, Georgia, USA: IEEE Press, 2009, pp. 1835–1842, ISBN: 978-1-4244-3549-4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1704175.1704440.

[8] S. Thorpe, A. Delorme, and R. V. Rullen, "Spike-based strategies for rapid processing", *Neural Networks*, vol. 14, no. 6–7, pp. 715 –725, 2001, ISSN: 0893-6080. DOI: 10.1016/S0893-6080(01)00083-1.

[9] (2015). Tiled convolution: fast image filtering, [Online]. Available: http://developer.amd.com/resources/documentation - articles / articles - whitepapers / tiled - convolution-fast-image-filtering/.

APPENDIX

The project plan is presented on the following Gantt chart.