SpiNNaker-based Visual Systems

End-of-first-year report

Garibaldi Pineda García

Supervisor: Steve Furber
Co-supervisor: Dave Lester

Advanced Processing Technologies Group
School of Computer Science
University of Manchester
United Kingdom

University of Manchester
APT Group

Contents

Contents

Abstract

3d reconstruction
slam, current approaches too expensive
neural representations can reduce computations
study of the brain and state-of-the-art in neural and classic vision
first years work is input

Acknowledgements

CONACYT/SEP

Chapter 1

Introduction

For most animals, an important part of perception is done through visual input.

This kind of input has given humans the possibility of culture through reading and writing, cognitive development.

We might take vision for granted, but we are reminded of its importance when we hear an unusual noise in a dark room.

An important aspect of vision is our ability to create mental maps of our current or, even past, locations.

The goal of the project is to create a 3D environment reconstruction.

There has been work on this field on "classic" computer vision, but for real-time they rely on high-performance power-hungry devices. Something that limits the actual utility of such systems for mobile applications. It would be a bit inconvenient to carry around a couple of car batteries in ones pocket.

We are able to do it with a highly-parallel 20-watts neural blob. There must be a more efficient way of doing this. A brief description of the brain and its function can be found in Chapter 2. We delve into the components of human vision in Chapter 3.

SpiNNaker provides a massively-parallel high-efficiency computing platform, inspired by the brain. It's an excellent choice for neuroscience research, particularly to study spiking neural networks. Its software stack has many ready-to-use neural models and development of new models can be performed in a straight forward manner. Chapter 4 has a more detailed description of this and other neuromorphic hardware.

Input for spiking neural networks has to be in *spike trains*, which are a series of spikes emitted by a neuron in a given time slot. In order to use video sources they need conversion. Few solutions which, mostly, require the use of custom hardware which is expensive and has low availability. We propose a parallel software based encoding.

This years work consisted in creating an input system for our spiking neural networks; details of this can be found in Chapter 6.

While there are some examples of hardware based retinas, they are still expensive or they have limited availability. Implementing a retina model using consumer hardware is of great help for people that are unable to obtain a silicon retina.

Of special interest are mobile applications, if we can provide a low-power solution to a silicon retina emulator, we could enable millions of phones, tablets or computers to work as an input to neural computations (QUALCOMM CHIP, SPINNAKER) and keep the traditional camera functionality.

1.1 Problem description

1.2 Objectives

3D environment reconstruction is a very active field of research. Advances in depth perception (KINECT) have made real-time simultaneous localization and mapping (SLAM) a possibility. One way of achieving is to use high-performance GPUs and solve the problem using raw power. Another is to use a mix of KINECT and RETINA, not fully neural??? We propose using an exclusively neural networks approach using SpiNNaker hardware.

1.3 Plan

Steps:

Image recognition

Time-based encoding, learning, classification, deep belief networks comparison, hierarchical structures

3D object recognition

Correlation in space and time, spiking neural networks should make an excellent match for this.

Depth perception

Binocular, depth-from-defocus, other sensors? Optic flow to infer motion?

Orientation and localization

Even more sensors? Make statistics/probabilistic models of past data?

Reconstruction

Get a top down approach? Interface 2 nets? Deconvolutional Networks

Chapter 2

A look into the brain

2.1 The brain

The brain is an exquisite piece of evidence of energy-efficient biological computation.

Neural systems in animals are different, from the simple ones found in insects to more complex ones in reptiles, birds and mammals. After millions of years of evolution, great apes, humans in particular, have one of the most intricate nervous systems. The human brain acts as regulator of this system and performs high level cognitive tasks.

It can perform the most diverse activities, from bird spotting to mathematics to art. All of this with about 20 watts of energy spread across many small computational units called *neurons*.

It consists of around 10^{12} individual neurons which are interconnected through about 10^{15} synapses.

Most of this neurons are arranged in thin sheet of about 1100 cm^2 area and a 2 to 4 mm. thickness.

So far, several functional units in the brain have been identified; but the exact mechanisms of how they perform is still unknown.

One of the most consuming tasks is vision, about 30% of the cerebral cortex is used in visual perception.

2.2 Neurons and responses

Different theories for the composition of the nervous system. Golgi thought neurons where fused into a continuous nerve network. Ramón y Cajal demonstrated that neural centres consisted of individual cells, with directed communicating through different parts of the cellular anatomy.

The nervous system is composed of specialized cells called *neurons*. Their area of expertise is long-range communication. While most cells in the body can "talk" to their neighbours, neurons have structures that allow them to communicate for up to XX cm (in the human case).

Neurons are composed of a soma (body), this part has similar components to other cells in the body. One of the specialized communication structures is called the *axon*, through it the neuron outputs a signal. *Dendrites* are at the other end of the information exchange and receive messages that other neurons sent through their axon, thus they can be seen as inputs for the cell.

Theories suggest they perform some kind of calculation, most of the time modelled as a threshold activation function

Some neurons use analog/continuous signals to transmit information, though they are mostly act as an interface to the exterior world.

Latest evidence suggests that the complex cognitive functions are performed using spikes, on-off responses, as a means for communication.

The place where axons and dendrites meet, is called the SSS, synapses

When one neuron's output elicits another neuron to spike,

Neuron models

From the very detailed Hudxley

2.3 Different languages

Neurons communicate using different "languages", spike-codes

Spike rate

How many spikes where produced in a time slot. Not much information can be encoded this way. Easy to transfer previous neural net work. Not entirely biologically plausible, specially for high cognitive tasks.

Spike timing

The precise time a spike was emitted. Lots of information, but still difficult to use. Polychronization might be the answer to learning/training.

Rank-order

Only the order of spike events are important, not the particular timing. Might be more robust than spike-timing but can encode less information. Some problems on training as well.

Input from sensors is most likely rate-based, though processing time and energy consumption in the brain suggests a different one is used for further processing

2.4 Artificial neural networks

First modelled as an on-off threshold gate, perceptron

Multi-layered networks and feedback, Hebbian learning

Spiking neural networks, third gen, include time as a factor, more accurate, more powerful mathematical properties,

learning studied using Hebbian back-prop, stdp, bcm

still work to be done on time-based learning

2.5 Conclusions

conclusions the brain

Chapter 3

Vision

Vision is one of the most important senses for animals; humans use it extensively for all kinds of tasks. Hunting, assessing danger, reading, driving, drawing, predicting rain from grey clouds, etc., these are all tasks that involve *seeing.*

There is a vast collection of knowledge about the components of vision, though a unified theory of vision (or the answer to *How do we see?*) has not yet been achieved.

Vision starts at the eye, which transforms electromagnetic radiation that assembles an image, into voltage pulses that our brain may interpret. This encoded images are sent to the posterior region of the brain through the optic nerves. The cortex then performs many computations that result in our ability to see.

3.1 The eye and the retina

Our everyday experience might lead us to believe that the eyes are sensory organs developed completely separate from the brain but, in fact, the retina is an extension of the brain that performs spatio-temporal compression of a continuous flow of "images" of the world.

The eye is composed of many parts that resemble a camera (LENS, CAMERA OSCURA, FILM)

After light has been transformed into an electrical representation, the retina takes over and computes a representation of it.

Photoreceptors have the task of transforming light into an electrical signal. Colour is perceived by special type of receptors *cones.* For low-light conditions and higher contrast sensitivity, we use *rods.* Vertebrates have both rods and cones. Evolutionary adaptation has made eyes in different animals have special ratios of cones and rods. Reptiles and fish have more cones, most likely because they "live" on daytime for a lack of worm blood.

Many mammals have retinas with more rods than cones. For primates the retina has has two almost dual sensor zones. Most of the photosensitive area has more rods than cones; a tiny region called the *foveal pit* has almost no rods, is densely packed with cones for high-resolution vision and is virtually blind when there is not enough light.

Horizontal cells average spatially (surround), input from photoreceptors; output to bipolar and to photoreceptors (adapt to different light conditions)

Bipolar cells, centre behaviour, input from horizontal and photoreceptors

IMAGE OF CENTRE SURROUND!!!!

First layers (photoreceptors, bipolar, horizontal cells) use analog signals, ganglion cells use spike trains.

Most authors agree that ganglion cells can be modelled by a *Difference of Gaussians* due to its centre-surround behaviour.

Ganglion cells extend to the Lateral Geniculate Nucleus, where information is relayed and organized so that the cortex can interpret it.

Organization makes left visual field sent to right hemisphere, right field to left hemisphere.

Redundancy keeps things working even if some neurons/receptors die out. To avoid saturation of nerve fibres and over-representation lateral inhibition might play a big role. It's specially useful for spike-timing encoding, since sensors give a rate based output that needs to be re-encoded.

3.2 The visual cortex

The portion of the cortex that is involved with visual processing has been estimated to about 30%.

It has been studied and areas have been labelled due to their function.

V1, V2, V...

3.3 Conclusions

Chapter 4

Neuromorphic hardware

4.1 Classical computing

classical computing

4.2 Neuromorphic trends

neuromorphic hardware trends

4.3 SpiNNaker

spinnaker info

4.4 Event-based model

event-based programming/infrastructure

4.5 Conclusions

conclusions neuro hardware

Chapter 5

3D environment reconstruction

Environment reconstruction has been receiving a lot of attention from research community, specially with things like Simultaneous Localization and Mapping (SLAM). Typically performed using cameras

5.1 Simultaneous localization and mapping

Examples of SLAM
High performance hardware and/or exotic sensors (laser/kinect-like)
Mix of Kinect + DVS
Rat neuro based SLAM

5.2 Visual cortex models

Lowe's work inspired by neuro
Hierarchical has been shown to provide geometric transformation invariance
Hierarchical neural networks for image interpretation
Hierarchical temporal memory

5.3 Conclusions

Chapter 6

From video to spikes

Spiking neural networks require inputs encoded as spike trains. The most common way to do this is to perform a continuous value to frequency transformation using Poisson sources. For most of sensory input in the body, this might be good enough but the retina performs spatio-temporal compression before feeding any information into the cortex.

There are some video-to-spike encoders but have some issues. Real-time encoders require custom hardware and are hard to come by. For off-line encoding, applications are limited to certain type of research, that is no real-time experiments could be performed. Our objective this year was to generate a real-time video-to-spike encoder using of-the-shelf components.

6.1 Real-time encoding

For mobile and robotics applications a real-time encoder is needed. Hardware based real-time video encoders are expensive and not massively produced, thus their availability is limited. Creating one with off-the-shelf components opens the potential users. Two different models chosen whose computational cost was low enough to keep them operating at real-time and had kept biological plausibility constraints.

General purpose computing in the graphics processor unit

GPU History GPU programming OpenCL Memory hierarchy

The foveal pit model

The highest resolution area of the eye is the foveal pit (see section 3.1). A functional model for this region of the retina was developed by Sen and Furber, they called the implementation the *Filter-overlap Correction algorithm* (FoCal)[1]. It's based on the response and physiology of the fovea. The authors concluded that using four different layers of ganglion cells, most of the visually relevant information could be recovered after encoding. Furthermore, the encoder outputs a collection of rank-ordered spike trains.

The ganglion cells themselves where modelled using Difference of Gaussians (DoG), Equation 6.1.

$$DoG_w(x,y) = \pm\frac{1}{2\pi\sigma_{w,c}^2}e^{\frac{-(x^2+y^2)}{2\sigma_{w,c}^2}} \mp \frac{1}{2\pi\sigma_{w,s}^2}e^{\frac{-(x^2+y^2)}{2\sigma_{w,s}^2}} \tag{6.1}$$

The size of the receptive field of the simulated cells depends on the layer they belong to, this is reflected in the convolution kernel's width and parameters. Variables $\sigma_{w,c}$ and $\sigma_{w,s}$ are the standard deviation for the centre and surround components of the DoGs for layer w. The signs for the equation will be ($-$,+) if the ganglion cell is *off-centre* and (+,$-$) if it is *on-centre*. The parameters for this equation can be found in table 6.1.

Table 6.1: Simulation parameters for ganglion cells

Layer	Behaviour	Matrix width	Centre std. dev. (σ_c)	Surround std. dev. (σ_s)	Sampling resolution (cols, rows)
1	OFF-centre	3	0.8	$6.7 \times \sigma_c$	1, 1
2	ON-centre	11	1.04	$6.7 \times \sigma_c$	1, 1
3	OFF-centre	61	8	$4.8 \times \sigma_c$	5, 3
4	ON-centre	243	10.4	$4.8 \times \sigma_c$	5, 3

For each cell type, a convolution kernel must be computed and stored in a matrix (DoG_w). For each element in the matrix we use Equation 6.1, substituting parameters specified in Table 6.1 and integer valued x-y coordinates whose origin is the centre of the matrix. For example, for the 3×3 kernel (layer 1 cells), the upper-left value would be calculated as follows:

$$
\begin{aligned}
DoG_3(x,y) &= -\frac{1}{2\pi\sigma_{3,c}^2} e^{\frac{-(x^2+y^2)}{2\sigma_{3,c}^2}} + \frac{1}{2\pi\sigma_{3,s}^2} e^{\frac{-(x^2+y^2)}{2\sigma_{3,s}^2}} \\
DoG_3(-1,-1) &= -\frac{1}{2\cdot\pi\cdot 0.8^2} e^{\frac{-((-1)^2+(-1)^2)}{2\cdot 0.8^2}} + \frac{1}{2\cdot\pi\cdot 5.36^2} e^{\frac{-((-1)^2+(-1)^2)}{2\cdot 5.36^2}} \\
&= 0.27399398
\end{aligned}
\tag{6.2}
$$

The procedure to encode images can be broken into two parts. First, algorithm 1 simulates the ganglion cells. It requires four independent 2D convolutions (Eq. 6.3) using DoG kernels calculated as explained in the previous paragraph.

$$
C(x,y,w) = I * DoG_w = \sum_i \sum_j \left(I(i+x, j+y) \cdot DoG_w(i,j)\right) \tag{6.3}
$$

Algorithm 1 FoCal, Part 1

```
procedure GanglionCells(image $I$, kernels $DoG$)
    $C \leftarrow \emptyset$
    for all $w \in Layers$ do
        $C \leftarrow C \cup I * DoG_w$
    end for
    return $C$
end procedure
```

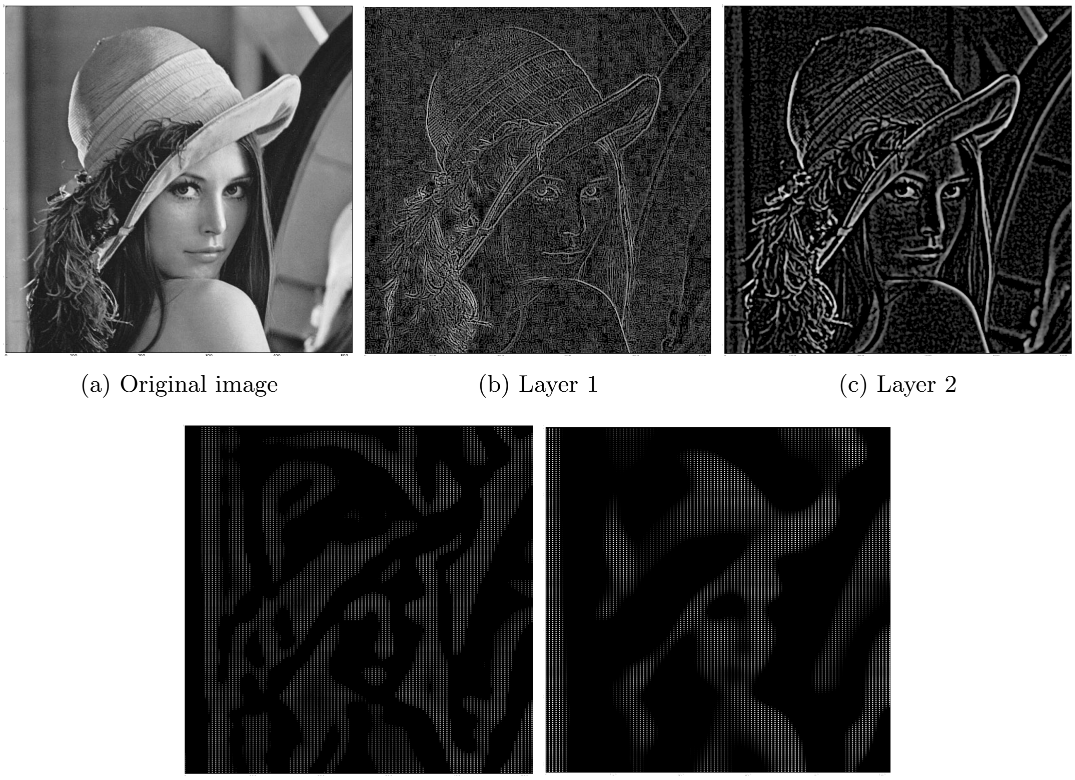
We'll call coefficients to the pixel values that come out of the convolutions (Figures 6.1b, 6.1c, 6.1d and 6.1e). This coefficients are interpreted as a quantity that is inversely proportional to the spike emission time. That is, the pixel with the largest coefficient value represents the ganglion cell that will spike first.

In order to check the validity of the generated spikes, a reconstruction procedure is employed (Equation 6.4). Each coefficient in C has an origin layer w, a value c and a position (k, l). For all coefficients in C, a DoG from it's respective layer will be weighed by it's value and be added at the it's position to the reconstructed image R. The procedure is based on the assumption that the DoG are orthogonal basis. Figure 6.2b shows the result of the image reconstruction procedure without any redundancy correction applied.

$$
R(x,y) = \sum_i \sum_j \sum_w C_w(i-x, j-y) DoG_w(i,j) \tag{6.4}
$$

The eye is unlikely to provide unnecessary information to the brain, that is redundant information is filtered somehow before it's delivered. In the retina, lateral inhibition is the most likely candidate to minimize information redundancy. It's still a matter of discussion where and by which cells is lateral inhibition performed in the retina. It is most likely to happen in layers prior to the ganglion cell layer. The DoG kernels are only an approximately orthogonal basis, thus the resulting coefficients from the convolutions in Algorithm 1 suffer from redundant information. That is, two neighbouring pixels might contain information that represent the same feature in the image. The main issue with this redundancy is that neighbouring coefficients might encode almost the same information with a similar value. Since the value provides the order of the spikes, this phenomenon will push other important coefficients into the later, less important, parts of the spike representation. In order to correct for redundancy, FoCal performs a second step (Algorithm 2).

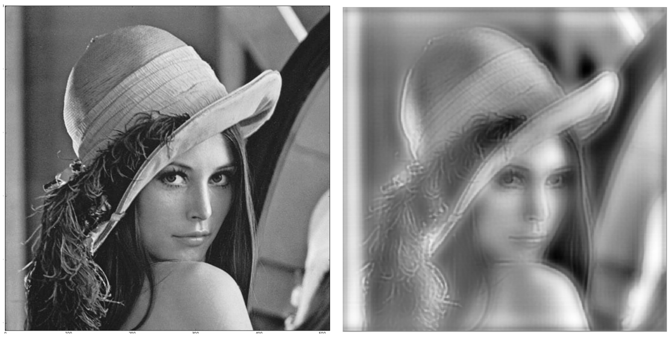
All the coefficients that where obtained from Algorithm 1, are put in a set C. For every step of the correction procedure the maximum coefficient is searched and its spatially surrounding pixels in all layers (Figure 6.3a) will be adjusted according to the correlation due to overlap between the maximum coefficient's convolution kernel and the other layer's kernels. The bold



(a) Original image (b) Layer 1 (c) Layer 2

(d) Layer 3 (e) Layer 4

Figure 6.1: Results of simulating ganglion cell layers (convolved images were enhanced for better contrast)



(a) Original image (b) 100% of raw spikes

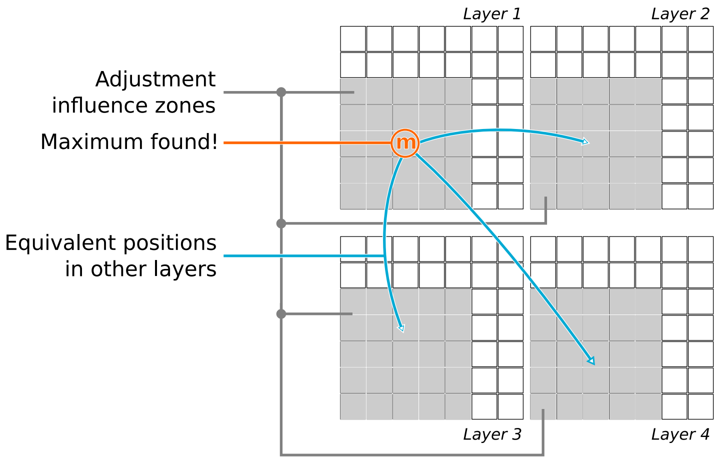
Figure 6.2: Reconstruction results without overlap correction.

square in Figure 6.3b shows the overlap of two 3 × 3 kernels of neighbouring pixels, a similar overlap is considered for the interaction between layers.

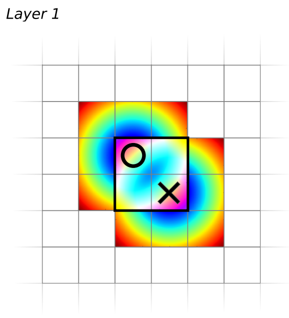
After this correction algorithm is applied only non-redundant spikes are preserved, this results in a much better reconstruction (Figure 6.4b). Not only is it more visually pleasing, but the fidelity of the reconstruction has been tested quantitatively; another interesting result is that only 10% of the rank-ordered and FoCal corrected spikes are needed to preserve 90% of the visually important information [2].

Algorithm 2 FoCal, Part 2

```
procedure Correction(coeffs $C$, correlations $Q$)
    $N \leftarrow \emptyset$                              ▷ Corrected coefficients
    repeat
        $m \leftarrow max(C)$
        $M \leftarrow M \cup m$
        $C \leftarrow C \setminus m$
        for all $c \in C$ do                              ▷ Adjust all remaining c
            if $Q(m, c) \neq 0$ then                      ▷ Adjust only near
                $c \leftarrow c - m \times Q(m, c)$
            end if
        end for
    until $C = \emptyset$
    return $M$
end procedure
```



(a) FoCal correction influence zones.



(b) Kernel overlap

Implementation details

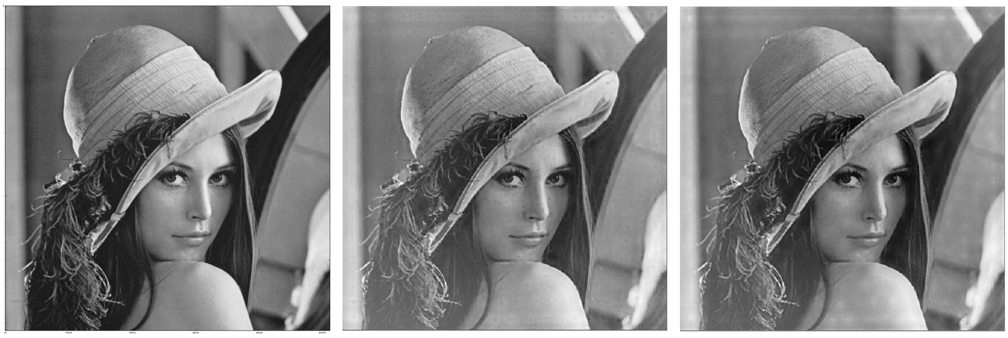
Different ways of applying convolutions to images on a GPU where implemented and evaluated. The first one, the **naïve approach**, implies a discrete convolution with the full 2D kernels. Since we are using squared kernels, this means $N^2 \times W \times H$ operations for a $W \times H$ image using a kernel of width N. As expected, performance drops quickly and the biggest problem for this approach was that biggest kernel (243×243 elements) requires more resources than the GPU's constant memory can provide (240 KBytes vs. 64 KBytes). This results in execution errors that may only be fixed using memory with greater latency to store the convolution kernel.

The second approach to perform a 2D DoG convolution with an image is to rely on **kernel separability**. A convolution kernel K is said to be *separable* if $K = K_1 * K_2 \ldots K_n$. Gaussian kernels are separable (Eq. **??**). and, fortunately a DoG is merely the subtraction of them (Eq. 6.5).

$$O = I * DoG = I * G_c - I * G_s \tag{6.5}$$

Applying the algebraic properties of convolutions and the fact that Gaussian kernels are separable, the full 2D DoG convolution can be performed using four 1D separated ones (Eq. 6.6).

$$O = I * DoG = G_{v,c} * G_{h,c} * I - G_{v,s} * G_{h,s} * I \tag{6.6}$$



(a) Original image (b) 100% of *corrected* spikes (c) 30% of *corrected* spikes

Figure 6.4: Results of reconstruction procedure

The main advantage of the separated kernel approach is a reduction of the number of operations needed ($4N \times W \times H$). An exception happens for the 3×3 kernel, in this case there are 12 operations vs. the 9 needed for the naïve approach.

The last approach, *Tiled Convolution* is reported by Advanced Micro Devices (AMD) [3]. They only present 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 @ 3.20 GHz processor with 8 GBytes of 64-bit DDR3 RAM @ 1600 MHz and a GeForce GT 720 GPU with 192 CUDA cores @ 797 MHz, 1 GBytes of 64-bit DDR3 RAM @ 1800 MHz.

Table 6.2: Convolution performance comparison.

	Layer 0	Layer 1	Layer 2	Layer 3
Naïve	0.0009s	0.0031s	0.0587s	N/A[1,2]
Separated	0.0021s	0.0055s	0.0172s	0.0472s
Tiled	0.0009s	0.0044s	0.1643	N/A[2]

[1] Unable to fit convolution kernel into constant memory.

[2] Unable to compile OpenCL code.

The performance of convolution in GPUs is bound by memory transfers, even if some of the information is reused.

In the retina, redundancy of information is reduced via lateral inhibition prior to any ganglion cell activity. In this algorithm, we perform a correction on the convolved images by adjusting the pixel values according to the correlation between convolution kernels (Alg. 2). The results of using correction (Fig. 6.2b) or not (Fig. 6.4b) show that the convolution stage can only provide redundant information. Furthermore, using only 30% of the corrected weights still provides enough visual information to reconstruct the original image [1].

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.

12fps is for good most phenomenon, full image encoding

This probably happens only once every so many ms

A dynamic vision sensor emulator

Output what a DVS does but with a camera as a source

Convolution of current and past frames ? centre - current / surround - past

Per-pixel adaptive threshold keeps fast changing pixels from spiking constantly, emulates refractory period of cells.

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 Vision Sensors (DVS) do but with limited dynamic range and lower temporal resolution [4], [5]. 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 back-end, 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.

6.2 Dataset creation

dataset for article

6.3 Conclusions

conclusions rank-ordered images

Chapter 7

Conclusions

7.1 Conclusion

7.2 Further work

7.3 Plans for second and third year

Bibliography

[1] 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.

[2] B. Sen, “Information recovery from rank-order encoded images”, Doctor of Philosophy Thesis, Faculty of Engineering and Physical Sciences, University of Manchester, 2008.

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

[4] 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`.

[5] 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`.