EXPANDABLE HARDWARE FOR COMPUTING CORTICAL FEATURE MAPS

Bertram E. Shi, Eric K. C. Tsang, Stanley Y. M. Lam, Yicong Meng

Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, email: {eebert, eeeric, eelym, eyicong}@ust.hk

ABSTRACT

We describe an expandable hardware architecture for the rapid simulation of feature maps inspired by the visual cortex. Feature maps are retinotopically organized arrays of neurons selective to different combinations of visual features. The responses of these maps are believed to be important for the brain to merge information from different visual cues. This architecture is based on a custom-designed board containing DSP and FPGA chips. It is modular in the sense that additional boards can be integrated into the system to accommodate cortical models of increasing complexity.

1. INTRODUCTION

In primates, visual information is processed in a number of different areas of the brain, collectively referred to as the visual cortex. Each of these areas contains neurons tuned to combinations of visual features, such as retinal position, ocularity, orientation, spatial frequency, temporal frequency and disparity, color and direction of motion [1][2]. These feature dimensions have natural geometries associated with them, and neurons in these areas are often arranged topographically, i.e., neighboring neurons are tuned to similar feature combinations. If we consider the distribution of neuronal responses across these feature dimensions, we obtain a high-dimensional topographical map. It appears that the visual system performs its task of perceiving the external environment by first decomposing the visual input across a multidimensional feature space.

We seek to build computational models of this multi-dimensional topographical map. Our motivation for this work arises from increasing evidence that by integrating complementary information about the environment, these maps account for the observed behavioral robustness of biological systems. Biological systems integrate information from different visual cues very early on in processing. For example, cells in the primary visual cortex are simultaneously selective to orientation, motion direction, speed and binocular disparity, which may be used to solve the binocular matching problem [3].

This paper describes a hardware system that accelerates the computation of cortical feature maps. We had several objectives in the development of this board. First, computation should be flexible enough to support different types of maps. Second, computations must be performed in real time to enable the system to control an active binocular vision head in response to visual input. Third, the system must be expandable so that it can accommodate cortical models with increasing complexity without altering the timing of computations in previously developed processing stages. To achieve these objectives, our system is modular, consisting of several identical boards. The computation on each board is performed by a DSP chip, which provides flexible real-time processing. This board has been designed with a flexible communication architecture that enables the results of its computations to be rapidly communicated to other boards. Thus, this board can be used to construct a system whose processing capability and memory expands with the complexity of the model.

The remainder of the paper starts by describing a model of visual cortical neurons that is implemented by these boards. Following that, we detail the architecture and implementation of the boards, as well as their functionality. Finally, we compare our boards with existing neuromorphic systems.

2. A MODEL OF VISUAL CORTICAL NEURONS

We use the energy model to model the selectivity of visual cortical neurons to spatial frequency, orientation, disparity, temporal frequency and direction of motion. Its basic components are linear filtering by pairs of phase quadrature filters followed by squaring and summing of the two filter responses. The linear filter, whose kernel is often equated with the receptive field (RF) profile of the neuron, establishes the stimulus selectivity. In the following, we describe how spatial filtering establishes orientation selectivity. The spatial receptive field of V1 simple cells can be accurately modelled by a Gabor function [4][5][6], which accounts for neuronal selectivity along the dimensions of retinal position, spatial frequency and orientation. For vertical orientations, the Gabor function has the form

\[ g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \cos(\Omega_x x + \phi) \]  

The standard deviations of the Gaussian envelope, \( \sigma_x \) and \( \sigma_y \), control the width of the receptive field and therefore the selectivity to spatial position. The parameter \( \Omega_x \) determines the peak spatial frequency tuning, and \( \phi \) is a phase shift parameter. By the uncertainty relationship, there is an inverse relationship between selectivity to spatial position and selectivity to spatial frequency. Other orientations can be obtained by rotating this function.

The response of an orientation selective simple cell whose receptive field is centered at retinal position \((x, y)\) can be modelled by

\[ r(x, y) = \left( \int \int g(x-x', y-y') dxdy \right)^2 \]  

The response of an orientation selective simple cell whose receptive field is centered at retinal position \((x, y)\) can be modelled by
where \( I(x, y) \) represents the input image intensity and \( |x|^+ = \max\{0, x\} \) is the positive half-wave rectifying nonlinearity.

The response of a complex cell can be modelled as the sum of four simple cell responses whose phase parameters differ by \( \pi/2 \). Since filter kernels that differ in phase by \( \pi \) are equivalent, except for a change in sign, the response is equivalent to squaring and summing the non-rectified responses of two filters: one with \( \phi = 0 \) and the other with \( \phi = \pi/2 \). This sum of squared filter outputs is called the orientation energy, and is maximized for input stimuli whose orientation matches the orientation of the filter kernels.

The basic architecture above can be extended to model neuronal selectivity along other dimensions. Adding temporal filtering establishes selectivity to temporal frequency and direction of motion. Linearly combining the spatio-temporal filter outputs from two eyes establishes selectivity to binocular disparity. Although inconsistencies between the energy model and biological data have been noted, this model is still a good first order approximation to the responses of visual cortical neurons, and thus is an appropriate starting point for constructing a neuromorphic model of them.

### 3. ACCELERATOR BOARD

As described above, a key component of the model is the implementation of linear filters modelling the receptive field profiles of neurons, as well as subsequent nonlinearities. In this section, we describe the board we have developed to accelerate these computations.

Figure 1(a) shows the system level architecture of the board. It can be roughly partitioned into two parts: the communication controller and the neural array simulator.

The communication controller is responsible for routing data between boards through low voltage differential signalling (LVDS) links, interfacing with external components such as cameras and active vision heads through the general purpose input-output (GPIO) ports, and routing inputs and outputs from the neural array simulator. This communication controller is an important element in ensuring the expandability of our system.

The neural array simulator takes inputs routed to it by the communication controller, and computes the outputs of arrays of cortical neurons, which it passes back to the communication controller. Our model for the cortical array outputs is based upon the energy model described previously. Because the outputs of these energy models are sparse, we encode input and outputs using an address-event-representation (AER) like encoding. However, because our input is frame synchronous, events occur every frame interval, rather than asynchronously.

Figure 1(b) shows a photograph of the accelerator board, which was designed at HKUST. For maximum flexibility and ease of reconfiguration, the logic for the communication controller is implemented using a Xilinx Spartan III FPGA operating at 100MHz. Data for communication is buffered using 4MB of SRAM partitioned into two equal banks. Some pins of the FPGA are routed to header pins as GPIO. The LVDS input/output pairs are implemented using National Instruments LVDS Channel Link Serializers and Deserializers. Each link transmits or receives 12 bits at 50MHz, for a rate of 600Mbps. The neural array simulator is implemented using a Texas Instruments 6414 DSP chip operating at 600MHz. The 1MB of internal storage is supplemented with 32MB of off-chip SDRAM.

### 4. BOARD FUNCTIONALITY

In this section, we describe the functionality currently implemented on the board.

#### 4.1 CMOS Camera Interface

The board accepts image input from a 352x288 pixel OmniVision OV6630 image sensor through the GPIO ports. The camera interface logic is implemented in the Xilinx FPGA, and one frame is
buffered in an SRAM bank before being passed to the neural array simulator for further processing. Once a frame of 8 bit pixel data has been buffered, it requires 0.6ms to transfer to the neural array simulator.

4.2 Image filtering

Spatial image filters model the neural spatial receptive field profiles. Our system currently implements low pass, high pass and Gabor-like filters. Low pass filters can be used to model the aggregation of neural outputs from an extended spatial area. High pass filters can be used to model the responses of edge-selective neurons in the retina. Gabor-like filters can be used to model the responses of orientation selective neurons in the primary visual cortex.

For computational efficiency, we implement two-dimensional low pass spatial filtering using an X-Y separable filter, which enables us to break down the 2D filter into a cascade of 1D filters in the X (horizontal) and Y (vertical) directions. Each 1D filter is implemented using the forward backward algorithm, where a recursive infinite impulse response filter is applied first from left to right (up to down) and then from right to left (down to up). To minimize computation, we use a first order filter, which computes the output of each pixel as a weighted average of the current input and the output of the previous pixel. Using the same filter in both directions results in an even symmetric filter with zero phase shift. Startup transients at the boundaries are minimized by careful choice of the filter initial conditions, based upon the assumption that the input image outside the boundaries of the array is equal to a constant value that is the same as the edge pixel value. Because the filters applied in the X and Y direction are the same, the resulting 2D impulse responses are approximately circularly symmetric.

Edge selective high pass filtering is obtained by subtracting the original image from a low pass filtered version of it. By ensuring the DC gain of the low pass filter is equal to one, we obtain a high pass filter with perfect DC rejection.

Orientation selective Gabor-like filters are obtained using an algorithm proposed by Unser [8]. Input data is first modulated by a complex exponential whose spatial frequency and orientation corresponds to the tuned spatial frequency and orientation of the Gabor-like filter. The data is then filtered using the low pass filter previously described, and finally demodulated. The bandwidth of the low pass filter determines the bandwidth of the resulting Gabor-like filter. The advantage of this approach is that Gabor-like impulse responses with arbitrary width, spatial frequency tuning and orientation tuning can be implemented using only nearest neighbor or pixel-wise operations, and that the two Gabor kernels differing in phase by $\pi/2$ are computed simultaneously. In our implementation, the real and imaginary parts of the filter can be computed for a 352 x 288 pixel input image in 2.3ms. High pass or low pass filtering alone requires only half the time.

4.3 Half-wave rectification and event encoding

As we move farther and farther into cortex, neurons become increasingly selective to particular combinations of input parameters. Thus, although the number of retinotopic maps expands, each of these maps becomes increasingly and increasingly more sparse.

For this reason, we adopt an event-based transmission architecture for transmitting maps between different boards. We generate events by integrating the outputs of the maps until they reach either a positive or negative threshold. At each frame, we transmit only the pixels whose values exceed the threshold. This reduces the amount of data we transmit per event. In addition, because we use both a positive and negative threshold, by looking at only positive or negative events, we obtain a half-wave rectified version of the output.

![Figure 2](a) Original picture from the camera. (b) Result of high pass filtering. (c) Result of orientation selective filtering.

4.4 Inter-board communication

We route maps between different boards using a routing architecture based on split and merge circuits, similar to that adopted in a recent neuromorphic model of orientation hypercolumns [12]. As shown in Figure 1(a), there are two LVDS input/output pairs on the communication controller for communication. A split circuit can take input from the left side, and pass one copy to the neural array simulator, and another copy out the left output. Split circuits can be used to implement fanout. A merge circuit can route both the output of the neural array simulator and the input from the right side alternatively through the right output. Merge circuits can be used to route maps computed on two or more different boards for combination on another board. By appropriately connecting boards via split and merge circuits, we can implement complex architectures mimicking the organization and interconnections between different areas of visual cortex.
5. COMPARISON WITH EXISTING NEUROMORPHIC SYSTEMS

Our goal is to develop an effective platform to compute a number of retinotopic maps so as to explore the advantages of biologically inspired processing systems. Unlike the vision chip approach, where all circuits required for a task from photosensing through processing are integrated on a single die [9], we partition the cortical processing among several DSPs with the communication routings on FPGA. The ability to program both the DSP and FPGA has the advantage of accelerating the development of cortical models. However, our platform is not as power efficient as a vision chip for mobile robotic applications. However, it is still possible to control an active vision system through the GPIO ports in real time.

Our system shares the advantages of flexibility and algorithmic reprogrammability with the Cellular Neural Network Universal Machine[10] and the Bi-i System [11]. Although it cannot perform computations as rapidly and efficiently as the analog processing circuits included in the CNNUM processing core, its computational power can be expanded by adding additional boards, at the price of additional power consumption.

Our system architecture is adapted from the modular approach proposed by Choi et al. [9], which allocates the computation of retinotopic maps among different mixed signal VLSI chips linked with AER split and merge circuits. The system we describe here has higher flexibility. Multiple orientation maps with higher resolution can be computed in a single board. Once the processing exceeds the real time limit, the system can be scaled by allocating additional computation to another board. Since the activities in the cortical maps are sparse, coding them as spike events will enable use to merging activities on a single AER bus without a dramatic increase in the data bandwidth.

We choose to model biological processing at a higher level than dynamically reconfigurable spiking systems [13][14], which consist of arrays of unconnected spiking neurons whose connectivity is handled by routing address events. Rather than modelling exact details at the spiking neuron level, we abstract the functionality of groups of neurons as a population of cortical cells arranged along functional dimensions, such as orientation, disparity and motion. Our approach is more effective in exploring the way to construct and interpret those populations for the higher level perception. Indeed, Mountcastle has argued that the effective unit of operation in the brain is not the single neuron, but groups of cells with similar functional properties [2]. However, because of the abstraction, our platform is not as effective in modelling detailed synaptic behaviors, such as the facilitation, depression, and spike-time-dependent plasticity.

6. CONCLUSION

We have described an expandable hardware architecture for accelerating the simulation of cortical feature maps. While this architecture consumes more power than custom mixed-signal chips which can compute similar maps, its capability to be reconfigured rapidly enables more flexible experimentation with different system level architectures, as well as different ways of pooling the outputs of several maps. Thus, algorithms and architectures developed and tested using this hardware will guide the path for the development of more power efficient mixed signal neuromorphic chips.

ACKNOWLEDGEMENTS

This work was supported by the Hong Kong Research Grants Council.

REFERENCES