Translating image sequences into spike patterns for cortical neuro-stimulation

Francisco J. Pelayo[1], Samuel Romero[1], Christian A. Morillas[1], Antonio Martínez[1], Eduardo Ros[1], Eduardo Fernández[2]

[1]*Dept. of Computer Architecture and Technology, University of Granada, Spain*
[2]*Dept. of Histology and Institute of Bioengineering, University Miguel Hernandez, Alicante, Spain*

Abstract

This paper describes a bio-inspired preprocessing and coding system devised for producing optimal multielectrode stimulation at the cortical level, starting from image sequences, and working at video rates.
The system is implemented as a hybrid platform with software and reconfigurable hardware. It produces as output a continuously varying stream of pulses or spike patterns. The main objective of this work is to build, in a near future, a portable system for a visual neuro-prosthesis able to stimulate efficiently an array of intra-cortical implanted microelectrodes. Thus, a set of parameters can be adjusted in the designed processing and spike coding modules in order to trade-off their technology constraints with the biological plausibility of their functional features. Spike timing and possible temporal correlation among different microelectrodes are preserved as important cues to improve the stimulation efficiency and/or to initiate plasticity processes at the receiver neurons in the primary visual cortex.

Key words: neural processing and coding, spiking neurons, artificial retinas, visual neuro-prostheses, bioinspired systems, reconfigurable computing.

1. Introduction

The work here presented has been carried out within the EC project CORTIVIS (QLK6-CT-2001-00279), which has as a main objective the development of a cortical visual neuro-prosthesis for the blind [1]. Also, different modules of the processing and coding system here described are being evaluated in the framework of the EC project SpikeFORCE (IST-2001-35271), in this case to be applied for the construction of artificial visuo-motor and multi-modal sensory integration systems.
The system addresses two objectives: (a) building a fully programmable and parameterized test platform for experimentation in the field of multielectrode neural stimulation, specially for visually evoked stimulation of microelectrode matrices, such as the Utah microelectrode array [2,3]; and (b), producing specific customized prototypes, integrated on ASIC (*Aplication Specific Integrated Circuit*) or in FPGA (*Field Programmable Gate Array*) chips, able to provide compact and low-power solutions for portable visual prosthetic systems.

Among the tasks planned in CORTIVIS, there is the implementation of a bioinspired visual-information processing system that must work in real time (that is, at a rate that can provide almost continuous visual perception), and able to produce neural stimuli similar to those received by the primary visual cortex of a sighted person. This system will be initially used for animal experimentation and for the development of the whole electronics required by the prosthesis. At the end, it must be capable of being integrated into an specific chip, customizable for each patient.
Figure 1 shows a schematic block diagram or the whole prosthetic system, in which participate different partners of the project [1]. It is composed by the bio-inspired visual pre-processing module (we call *retina model*); and a spike coding block that interfaces the retina model with a radio-frequency (RF) link. The RF module provides a wireless transfer of power and signal to a sub-cutaneous implanted receiver. This implanted circuit must decode the signals, identifying the target electrode of each impulse (decoder), and conforming the voltage shape (D/A converter) of the spike to be applied to the corresponding intra-cortical electrode. A number of similar prosthetic approaches are being developed by different labs world-around; all of them trying to cope with the challenge of providing some functional vision to blind subjects.

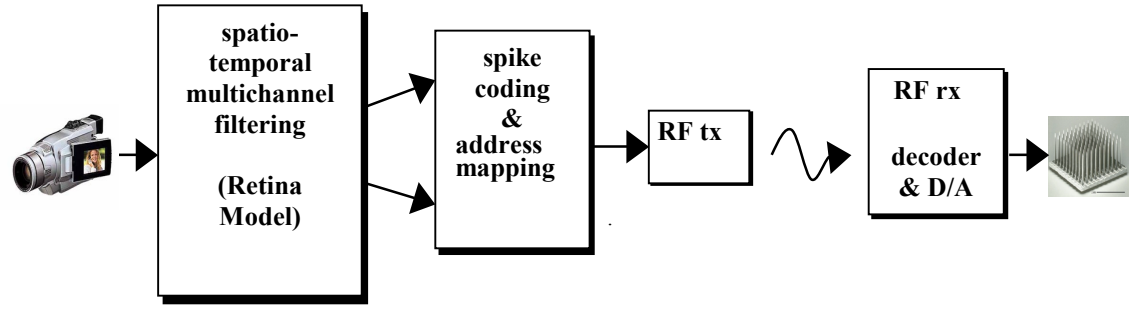

Fig.1: Schematic view of the main blocks of the neuroprosthesis under development. It includes a bioinspired visual information pre-processing block (retina model), a coding block that translates the retina output into pulses modulated with different cadences and temporal lags, and whose projection onto the electrode matrix must be fully reconfigurable. Finally, an integrated telemetry system will deliver the electric pulses to the electrode stimulator.

The first block in Fig.1, summarized in Section 2, has been described in [4]. The work here presented is a part of the second block in Fig.1, which performs a continuous conversion of activity values into pulses according to the firing neuron model described in Section 3.

2. Retinal pre-processing module

The images arriving to the pulse coding module in Fig.1 are previously processed by a retina model with the functional organization depicted in Figure 2. It includes a bank of spatial filters that model different color and intensity opponent channels [4]. The output of this filters are weighted are combined to produce a unique intensity matrix. An additional step in this processing allows to assign or map the intensity matrix to a limited number of electrodes, by performing a quantization or pixelization that defines the equivalent receptive field for each electrode. Different sizes, shapes, and spatial distribution of this receptive fields are also programmable in the present MATLAB implementation of the retina model.

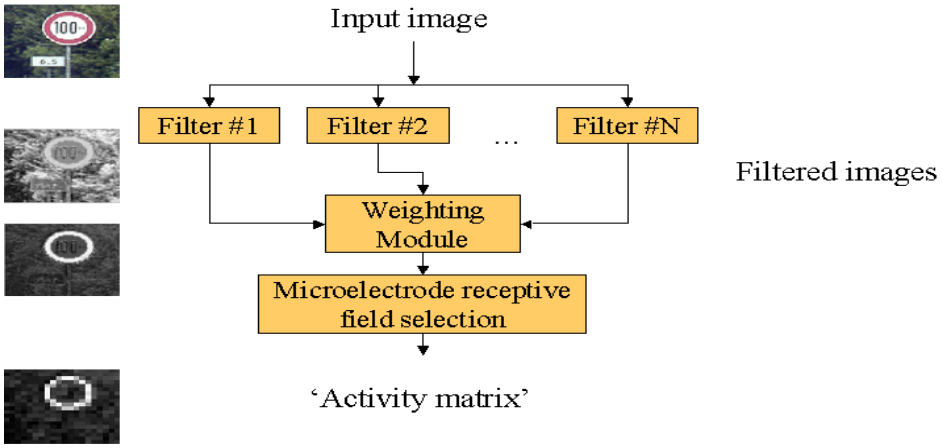

Fig. 2: Functional organization of the retinal model.

3. Leaky integrate-and-fire neuron model

The continuous-varying components of the activity matrix produced by the retina module are converted to neuromorphic pulse-coded signals, through a circuit that resembles spiking neurons. Thus, while the retina model described in the previous Section approximates the function of the photoreceptor, horizontal, and bipolar layers in a biological retina; these spiking neurons emulate the function of the retinal ganglion cells.

Our design is based on a leaky integrate-and-fire model of spiking neurons (see for example [5]). This abstraction models a neuron by using a capacitor and a resistor in parallel, and the inputs values (in this case, the components of the activity matrix) are represented by currents. The incoming current from the image matrix charges the capacitor until a threshold is reached, discharging the capacitor and producing a pulse (spike). The higher is the incoming current, the higher will the spike frequency be. The capacitor leaks its charge so that in absence of stimulus, the accumulated value will fall to a resting potential.

Although this model is essentially analog, we have chosen a digital implementation to have a more flexible and standard approach, that could be in the future easily customized for each implanted device.

A preliminary digital implementation of the spiking neuron model, described in terms of the Xilinx block set for Simulink [6], is shown in Fig.3. The neuron accumulates input values coming from its receptive field into a register, until the value stored reaches a programmable threshold. Then it fires and discharges the accumulated value, resetting the register to zero (or to a given programmable value). The value of the leakage factor, also configurable by de user, makes the accumulated value diminish for zero or very low input values.

The neuron model specification given in Figure 3 is directly synthesisable with the System Generator package of Xilinx. The synthesis results give low area consumption for the device we're employing (a 0.2% of a FPGA Virtex2000 [7]), and the timing analysis shows that the working speed (clock cycles of tens of nanoseconds) is far

superior to that needed to match the biological inter-spike delay, in the order of milliseconds.

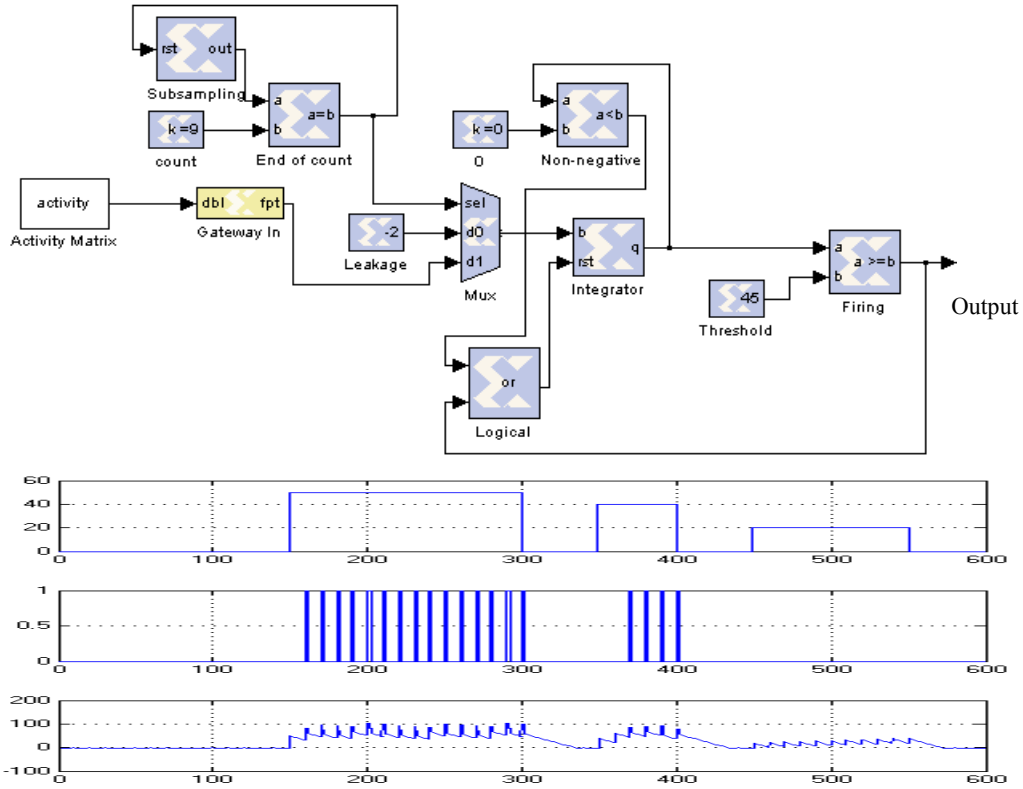

Fig. 3. (*Top*) Schematics block diagram of a single spiking neuron described in terms of the digital modules of the Xinlinx block set for Simulink. (*Bottom*) simulation results of the spiking neuron for an ideal sequence of light flashes of different duration and intensity (top trace). The middle trace shows the voltage pulses at the output of the neuron, and the bottom trace shows the accumulated activity value (i.e. the output of the integrator module). For this experiment, the firing threshold has been set to 70, and the leakage term to –2. It is observed that, when the input stimuli intensity is such that the neuron fires, the 'time to the first spike' is inversely proportional to the intensity of the input flash.

4. Extension of the model for an array of electrodes

The model for one neuron must be extended to be used with an array of microelectrodes. We will use at least a 10 by 10 electrode array. Taking the module in Fig.3 as basic building block, we have the choice of replicating it 100 times, or building a sequential architecture to multiplex and switch between electrodes; thus saving circuit resources of the FPGA.

The first choice is easier to develop, and all the spiking neuron circuits would work in parallel. Its disadvantage is the need for chip area, at least 100 times the needed to implement a single neuron. The second choice makes a better use of the resources available in the FPGA device, but it must work serially, scanning and updating every electrode. The payoff for this saving in silicon area is time consumption. This disadvantage is overcome thanks to the speed of computing for a single neuron.

For a 10x10 microelectrode array, if we assume that all the spiking neurons must be "refreshed" within 1 millisecond, we should dedicate to each electrode less than 10

microseconds. That time supposes no problem for our design, capable of updating the neuron state (and firing a spike event if needed) in the order of tens of nanoseconds. For a clock frequency of 1 MHz, we will be capable of addressing a 1000 electrode array, while our design is able to work properly up to 40 MHz. In other words, 40000 electrodes could be stimulated with an inter-spike temporal resolution equal or lower than one millisecond.

Thus, we consider the second option, as it accomplishes well with the constraints and performance requirements of the visual prosthesis. This sequential architecture uses two RAM memories to store the current and accumulated values for every electrode, and is pipelined (see Figure 4). A counter switches to every address of these memories, in order to update the values for the electrodes.

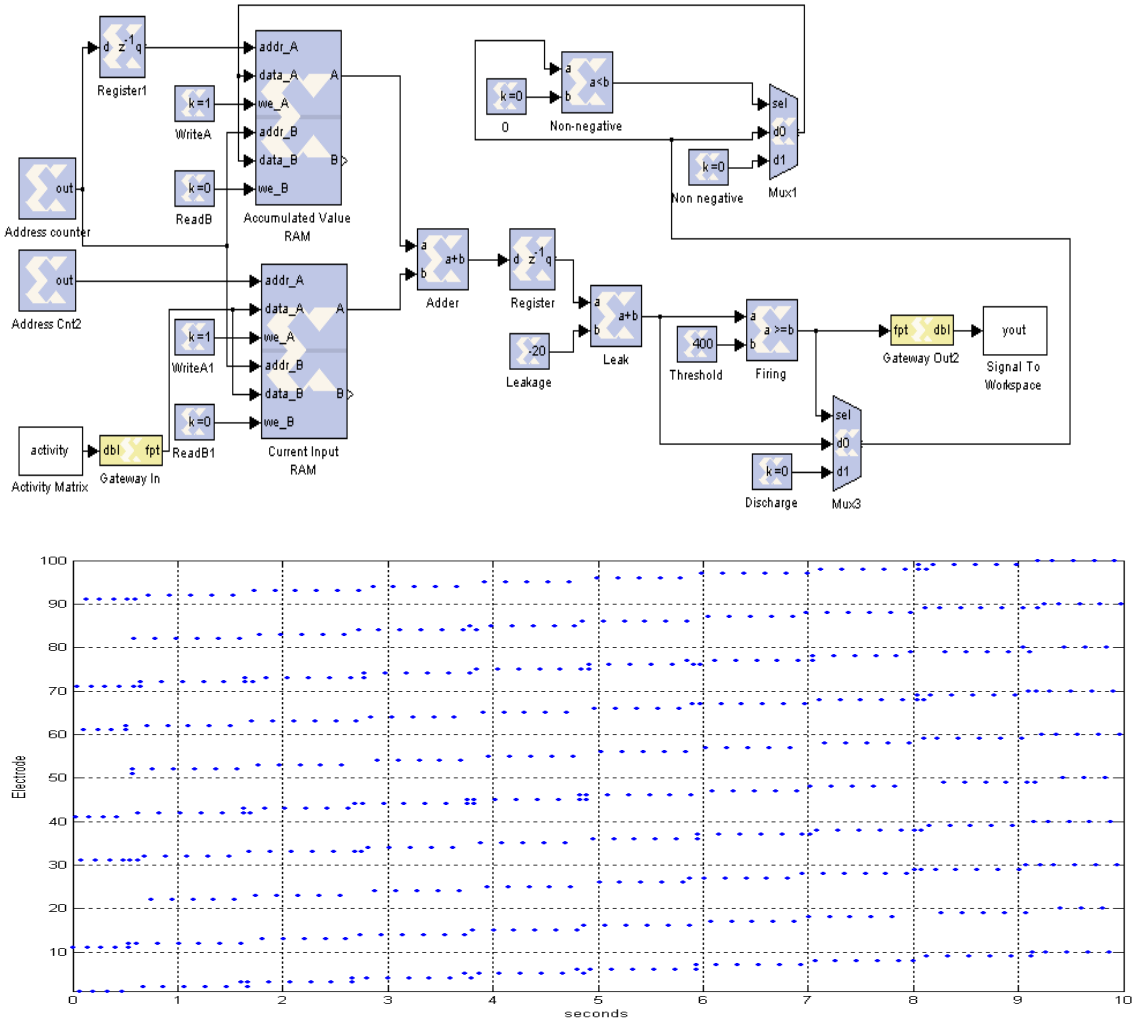

Fig.4. (*Top*) Extension of the spiking neural circuit in Fig.3 for stimulation of an array of microelectrodes (see text for details). (*Botton*) Outputs produced by the system (for stimulation of an array of 10x10 microelectrodes) in response to the horizontal displacement of a vertically oriented moving bar. The image sequence (10 seconds long, 160x120 pixels, 15 frames per second), is pre-processed by the retina model of Section 2. The bar's displacement produces the progressive firing of elements in the same row (electrodes 1 to 10, 11 to 20, etc). The firing for electrodes in the same columns (1, 11, 21, …) tend to be synchronous.

5. Conclusions

The design of a spiking neural system for visual cortex stimulation with microelectrode arrays has been presented. Its implementation in digital hardware provides a flexible design, achieving a high performance with response times several orders of magnitude lower than that of biological systems; thus allowing the temporal multiplexing of computations without degrading important features such as the inter-spike times of the firing neurons. The use of reconfigurable circuitry (FPGA) lets us to adjust or even change the spiking model easily.

Performing a bio-inspired pre-processing of the visual information that will be transmitted to the intra-cortical neuro-stimulator is essential, trying to have the signal reach every micro-electrode: (a) convey the biggest amount of "useful information" that is possible for visual perception and (b) exhibit the maximum independency of the illumination conditions. This pre-processing is carried out by the retina model briefly described in Section 2, and previously presented in [4], whose output feeds the spiking neural system. The retina model, complemented with the spiking neural module conform an "artificial retina" that translates image sequences into a continuous, parallel or serial, stream of short pulses representing spike events.

The analysis of multi-electrode recordings of biological retinas obtained by other partners of the CORTIVIS project, will allow us to contrast our model and tune its parameters to match the response of natural cells to a given visual stimulus.

Also, since the continuous stream of spikes generated by the artificial retina must be sent serially through the RF channel, an arbitrated address event representation (AER) scheme is being implemented [8], in order to transfer efficiently the inherent parallel visual information processed by the retina. This work is being carried out together with other partner of the project CORTIVIS (the INESC-ID of Lisbon, Portugal), and will be published elsewhere.

Acknowledgements

This work has been supported by the EC under the European projects CORTIVIS (Cortical Visual Neuro-prosthesis for the Blind, QLK6-CT-2001-00279), and SpikeFORCE (Real-time Spiking Networks for Robot Control**,** IST-2001-35271).

References

[1] CORTIVIS: http://cortivis.umh.es/

[2] Normann RA, Maynard EM, Guillory KS, Warren DJ. *Cortical implants for the blind.* IEEE Spectrum 54-59, May (1996)

[3] Maynard EM, Nordhausen CT, Normann RA. *The Utah intracortical electrode array: a recording structure for potential brain-computer interfaces.* Electroenceph. Clin. Neurophysiol. 102: 228-239 (1997).

[4] F.J. Pelayo, A. Martínez, S. Romero, Ch.A. Morillas, E. Ros, E. Fernández: *Cortical Visual Neuro-Prosthesis for the Blind: Retina-Like Software/Hardware Preprocessor.* To appear in Proc. of the IEEE-EMBS International Conference on Neural Engineering (NER-2003); Capri, 20-22 March 2003.

[5] Gerstner W, Kistler W. *Spiking Neuron Models*. Cambridge University Press, 2002.

[6] Xilinx. "Xilinx System Generator v2.1 for Simulink. User Guide. Xilinx Blockset Reference Guide".

[7] Xilinx. www.xilinx.com

[8] Mahowald M. *An Analog VLSI System for Stereoscopic Vision*. Kluwer Academic Publishers, 1994