

Exploration of spatial-temporal dynamic phenomena in a 32×32-cells stored program 2-layer CNN Universal Machine Chip Prototype

István Petrás¹, Csaba Rekeczky¹, Tamás Roska¹, Ricardo Carmona², Francisco Jiménez-Garrido²,
Angel Rodríguez-Vázquez²

¹ *Analogical and Neural Computing Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u.11, Budapest, 1111 - Hungary*

² *Instituto de Microelectrónica de Sevilla-CNM-CSIC
Avda. Reina Mercedes s/n, 41012 Sevilla (SPAIN)
Tel. +34955056666 Fax. +34955056686*

Abstract. This paper describes a full-custom mixed-signal chip that embeds digitally programmable analog parallel processing and distributed image memory on a common silicon substrate. The chip was designed and fabricated in a standard 0.5 μ m CMOS technology and contains approximately 500,000 transistors. It consists of 1024 processing units arranged into a 32×32 grid. Each processing element contains two coupled CNN cores, thus, constituting two parallel layers of 32×32 nodes. The functional features of the chip are in accordance with the 2nd Order Complex Cell CNN-UM architecture. It is composed of two CNN layers with programmable inter- and intra-layer connections between cells. Other features are: cellular, spatial-invariant array architecture; randomly selectable memory of instructions; random storage and retrieval of intermediate images. The chip is capable of completing algorithmic image processing tasks controlled by the user-selected stored instructions. The internal analog circuitry is designed to operate with 7-bits equivalent accuracy. The physical implementation of a CNN containing second order cells allows real time experiments of complex dynamics and active wave phenomena. Such well-known phenomena from the reaction-diffusion equations are traveling waves, autowaves, and spiral-waves. All of these active waves are demonstrated on-chip. Moreover this chip was specifically designed to be suitable for the computation of biologically inspired retina models. These computational experiments have been carried out in a development environment designed for testing and programming the analogic (*analog-and-logic*) programmable array processors.

1 Introduction

Various phenomena of the world around us are the result of local interactions of particles. This is true for both micro and macro world, for molecules, cells and even for members of a population. Two large classes of such infinitely many phenomena are the pattern formation in biology (e.g. spots of animals or plants) or in chemistry and active wave propagation (e.g. action potential in the nervous system, propagation of cardiac muscle excitation, blood coagulation process). Fig. 1 shows examples of pattern formation and active wave phenomena.

The expression “active wave” means that the wave propagation occurs in an “energetically active” medium, i.e. the energy conservation law does not hold because the media can inject energy into the system, therefore the amplitude and the waveform are preserved during the propagation. These are the most important differences when compared with classical waves. To make any reasonable complex pattern formation, we need the local activity of the cells [34].

Up to now the typical way to carry out experiments related to these phenomena was either laboratory work (e.g. with chemicals) or through time-consuming computer simulation. Other option is to build dedicated hardware circuitry. Although it is fast, its disadvantage is the lack of easy programmability.

The Complex Cell CNN programmable array computer [5][11] is an extension of the CNN Universal Machine [1]-[4]. The multilayer or higher order elementary cells are or-

ganized into a 32×32 square grid. Each cell has second order dynamics and local interconnections to its neighbors. This structure is especially suitable for computing a certain set of ordinal differential equations. Simple PDEs can be transformed so that they can be easily programmed on the array computer. With the programmable Complex Cell CNN Universal Machine scientists are provided with a unique opportunity to study active-wave propagation and other reaction-diffusion differential equation based phenomena in a programmable manner. Exploiting the inherent programmability of the CNN architecture, complex wave-computing analogic algorithms can be designed.

In this paper we show how to transform the mathematical model of the phenomena outlined above into a form suitable for parallel processing on the Complex Cell CNN chip and we present the first measurement results.

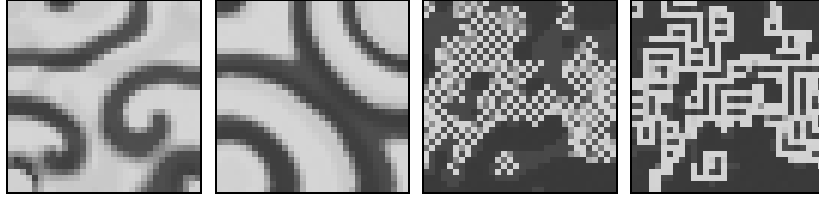


Fig. 1: spiral- and autowaves and patterns measured on the CACE1k chip (image size: 32×32)

The following section describes the mathematical model of the core of the complex cell chip. In Section 3 the prototype chip is introduced. Section 4 describes the relation between the reaction-diffusion equations and the structure of the complex cell chip. Finally, Section 5 contains the real-time chip measurements. We should emphasize that no simulation results are presented here. This chip was designed also for computing biologically inspired retina modeling. The implementation of the main effect of a retina model will be published in the near future elsewhere.

2 Computational architecture

The architecture of the chip follows the design of the first order CNN-UM chip [10], but its first order cell core is replaced by a second order one [5]. With this change we are able to reproduce basic active wave phenomena (See Fig. 1). Fig. 2 shows the functional architecture of the 2nd order, two layer CACE1k core.

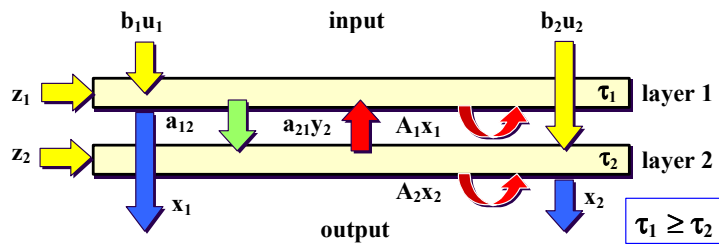


Fig. 2: Functional diagram of the CACE1k chip

The evolution law of the complex cell CNN is the following differential equation system:

$$\begin{aligned}
\tau_1 \frac{dx_{1,ij}(t)}{dt} &= -g(x_{1,ij}(t)) + \sum_{kl \in N_1} A_{1,kl} x_{1,kl}(t) + a_{12} x_{2,ij}(t) + b_1 u_{ij} + z_{1,ij} \\
\tau_2 \frac{dx_{2,ij}(t)}{dt} &= -g(x_{2,ij}(t)) + \sum_{kl \in N_1} A_{2,kl} x_{2,kl}(t) + a_{21} x_{1,ij}(t) + b_2 u_{ij} + z_{2,ij} \\
1 \leq i \leq M, 1 \leq j \leq M, 0 \leq |u_{ij}| \leq 1, 0 \leq |x_{ij}| \leq 1
\end{aligned} \tag{2.1}$$

Where:

$$g(x_{n,ij}) = \lim_{m \rightarrow \infty} \begin{cases} mx_{n,ij} & \text{if } x_{n,ij} > 1 \\ x_{n,ij} & \text{if } |x_{n,ij}| \leq 1 \\ -mx_{n,ij} & \text{if } x_{n,ij} < -1 \end{cases} \tag{2.2}$$

Function $g(\cdot)$ regulates the state so that it stays within the +1..-1 interval. Variables u_1, u_2 are the independent inputs, b_1, b_2 are their weight factors respectively; $z_{1,ij}, z_{2,ij}$ are space variant bias maps.. Variables x_1, x_2 denotes the state variables of the layers. Each x_{ij} corresponds to one cell; it is one pixel of a M by N image if we consider the system's output as a picture. A_1, A_2 are the weights of the intra-layer couplings, a_1, a_2 are the inter-layer weights. Equation (2.1) utilizes the so-called *full-range model* (FSR)[7], where the voltage of the state variable is always the same as the output.

3 Chip description

3.1 Architecture of the chip and basic processor structure

The prototype chip consists of an analog programmable array processor of 32×32 identical cells (Fig. 3), surrounded by the boundary conditions for the CNN dynamics. There is also an I/O interface, a timing and control unit and a program memory. The interface consists of a serializing-deserializing analog multiplexor. The program memory is composed of the analog program memory, and the logic program memory and the switch configuration registers. The analog instructions and reference signals need to be transmitted to every cell in the network in the form of analog voltages. Thus, a bank of D/A converters interfaces the analog program memory with the processing array. Distributing analog references across large distances within a chip is not a trivial task. Apart from the problems caused by electromagnetic interference, voltage drops in long metal lines carrying currents can be quite noticeable. Thus, in order to avoid this signal buffering and low-resistance paths must be provided. Finally, the timing unit is made of an internal clock/counter and a set of FSMs that generate the internal signals which control the processes of images up/downloading and program memory accesses.

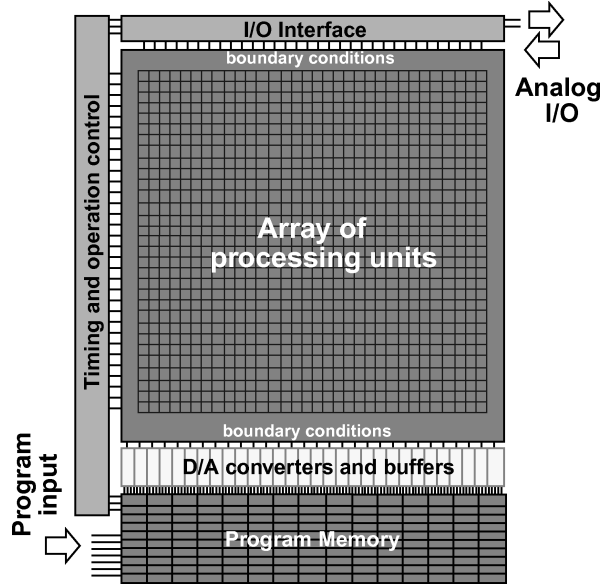


Fig. 3: Architecture of the chip

Every elementary processor of the Complex Cell CNN programmable array chip includes two coupled continuous-time CNN cores (Fig. 4(a)) belonging to each of the two different layers of the network. The synaptic connections between processing elements of the same or different layer are represented by arrows in the diagram. The basic processor also contains a programmable local logic unit (LLU) and local analog and logic memories (LAMs and LLMs) to store intermediate results. All the blocks in the cell communicate via an intra-cell data bus, which is multiplexed to the array I/O interface. Control bits and switch configuration are passed to the cell directly from the global programming unit.

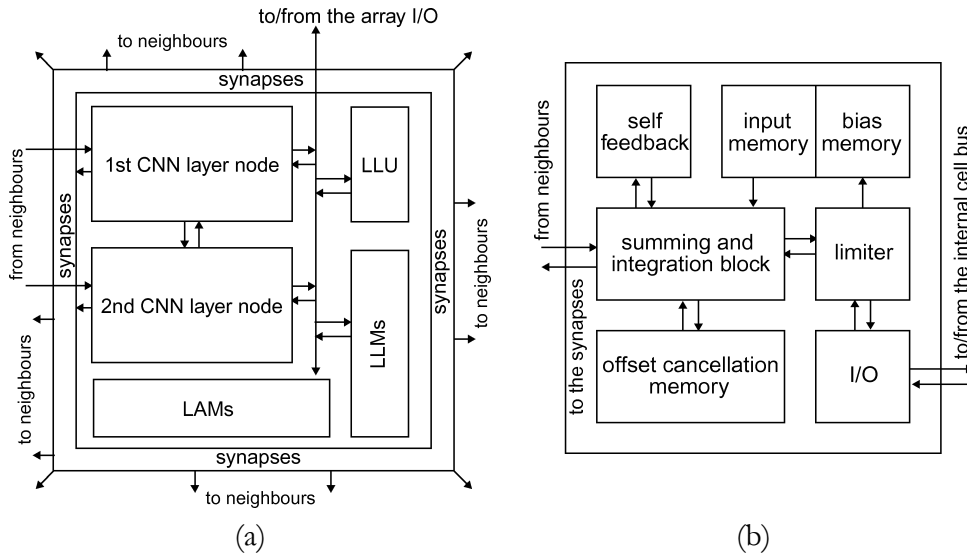


Fig. 4: Conceptual diagram of the (a) basic cell and (b) CNN layer node

The internal structure of each CNN core is depicted in Fig. 4(b). Each core receives contributions from the rest of the processing nodes in the neighborhood which are summed and integrated in the state capacitor. The two layers differ in that the first layer has a scalable time constant, controlled by the appropriate binary code, while the second layer has a fixed time constant. The evolution of the state variable is also driven by self-feedback and by the feed forward action of the stored input and bias patterns. There is a voltage

limiter for implementing the FSR CNN model [7]. The state variable is transmitted in voltage form to the synaptic blocks, in the periphery of the cell, where weighted contributions to the neighbors' are generated. There is also a current memory that will be employed for cancellation of the offset of the synaptic blocks. Initialization of the state, input and/or bias voltages is done through a mesh of multiplexing analog switches that connect to the cell's internal data bus.

3.2 Prototype chip data

The prototype chip has been designed and fabricated using 0.5 μm single-poly triple-metal CMOS technology. Its dimensions are 9.27 x 8.45 mm² (Fig. 5). The cell density achieved is 29.24cells/mm². The time constant of the layers is around 100ns (unscaled). The programmable dynamics of the chip permit the observation of different phenomena of the type of propagation of waves, pattern generation, etc (See Section 5 for examples). Table 1 summarizes the most relevant data of the prototype chip.

<i>Technology</i>	<i>0.5 μm CMOS 1-P 3-M</i>
<i>Number of cells</i>	<i>32 x 32</i>
<i>Die area</i>	<i>9.27 x 8.45 mm²</i>
<i>Die area (w/o pads)</i>	<i>8.77 x 7.94 mm²</i>
<i>Array area</i>	<i>5.98 x 5.93 mm²</i>
<i>Package</i>	<i>ceramic PGA - 100</i>
<i>Power supply voltage</i>	<i>3.3V</i>
<i>Logic "0" / Logic "1"</i>	<i>0V / 3.3V</i>
<i>Accuracy on the weights</i>	<i>8b</i>
<i>Image samples resolution</i>	<i>7-8b</i>
<i>I/O rates</i>	<i>10Ms/s</i>
<i>CNN time constant</i>	<i>below 100ns</i>

Table 1: Characteristic parameters of the CACE1k prototype chip

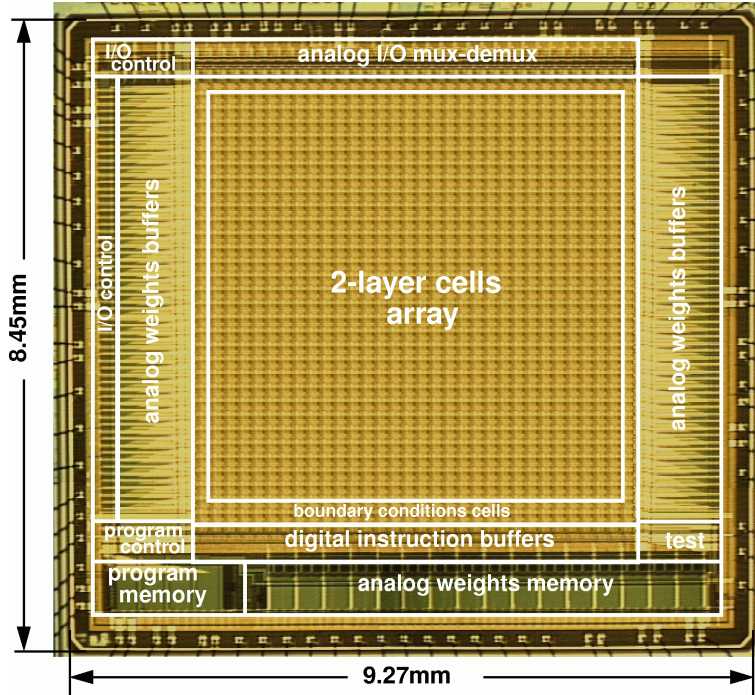


Fig. 5: Microphotograph of the prototype chip

4 Complex cell CNN-UM chip as a ‘‘PDE solver’’ – typical waves

4.1 General PDE formulation

Let us consider the following PDE formulation of a coupled vector valued nonlinear reaction-diffusion system (examples for nonlinear PDEs could be found in [26]- [32]):

$$\begin{aligned}
\frac{d}{dt}\phi_1(\bar{x},t) - \text{div}[c_1(\bar{x},t) \text{grad}(\phi_1(\bar{x},t))] &= \alpha_1(\phi_1(\bar{x},t)) + \beta_1(\phi_1(\bar{x},t_0)) + \gamma_1(\phi_2(\bar{x},t)) \\
\frac{d}{dt}\phi_2(\bar{x},t) - \text{div}[c_2(\bar{x},t) \text{grad}(\phi_2(\bar{x},t))] &= \alpha_2(\phi_2(\bar{x},t)) + \beta_2(\phi_2(\bar{x},t_0)) + \gamma_2(\phi_1(\bar{x},t)) \\
\phi_1(\bar{x},t_0) &= \phi_{10}(\bar{x}); \quad c_1(\bar{x},t) = g(|\text{grad}(\phi_1(\bar{x},t))|) \\
\phi_2(\bar{y},t_0) &= \phi_{20}(\bar{x}); \quad c_2(\bar{y},t) = g(|\text{grad}(\phi_2(\bar{x},t))|)
\end{aligned} \tag{4.1}$$

where $[\phi_1(\bar{x},t) \phi_2(\bar{x},t)]$ can be interpreted as the time evolution of a vector valued image intensity ($[\phi_{10}(\bar{x}) \phi_{20}(\bar{x})]$ are the initial images), the vector \bar{x} represents the spatial coordinates, the time variable t can also be interpreted as the scaling parameter and c is the conductance parameter in the *diffusion term*. The right hand side of the equations consists of three *reaction terms*: (i) $\alpha(\cdot)$ is the ‘‘self-reacting term’’; (ii) $\beta(\cdot)$ is a ‘‘spatial constraint’’ (calculated from the initial data); and (iii) $\gamma(\cdot)$ is the ‘‘cross-coupling’’ term. Some proposed functions for $g(\cdot)$ are:

$$\begin{aligned}
g_1 &= \exp\left(-|\text{grad}(I(\bar{x},t))|^2 / K^2\right), \quad K > 0 \\
g_2 &= \left(1 + (|\text{grad}(I(\bar{x},t))| / K)^{1+\alpha}\right)^{-1}, \quad \alpha > 0, \quad K > 0
\end{aligned} \tag{4.2}$$

Decoupling the two equations ($\gamma_1 = \gamma_2 = 0$) and choosing $\alpha(\xi) = -\xi$ the PDE formulation is given in the form of the so-called ‘‘biased’’ nonlinear anisotropic diffusion equation (Nordström [27]) an extended version of the Perona-Malik formulation [26]. Gerig et. al [29] first proposed the application of coupled nonlinear diffusion systems for vector-valued image processing that has been further studied by many others [32]. For our purposes – motivated by silicon implementation – we focus on a simplified version of (4.1) fixing the diffusion parameter to a constant value.

4.2 Spatial discretisation

In two spatial dimensions and assuming $c = \text{const.}$ (4.1) and (4.2) reduces to:

$$\begin{aligned}
\frac{d}{dt}\phi_1(x,y,t) - c_1 \text{div}[\text{grad}(\phi_1(x,y,t))] &= \alpha_1(\phi_1(x,y,t)) + \beta_1(\phi_1(x,y,t_0)) + \gamma_1(\phi_2(x,y,t)) \\
\frac{d}{dt}\phi_2(x,y,t) - c_2 \text{div}[\text{grad}(\phi_2(x,y,t))] &= \alpha_2(\phi_2(x,y,t)) + \beta_2(\phi_2(x,y,t_0)) + \gamma_2(\phi_1(x,y,t))
\end{aligned} \tag{4.3}$$

After spatial discretisation using the finite difference approach one obtains:

$$\begin{aligned}
\frac{d\phi_{1,ij}(t)}{dt} &= \gamma_1(\phi_{2,ij}(t)) + \alpha_1(\phi_{1,ij}(t)) - c_1\phi_{1,ij}(t) + \frac{c_1}{4}(\phi_{1,i-1,j}(t) + \phi_{1,i+1,j}(t) + \phi_{1,ij-1}(t) + \phi_{1,ij+1}(t)) + \beta_1(\phi_{10,ij}) \\
\frac{d\phi_{2,ij}(t)}{dt} &= \gamma_2(\phi_{1,ij}(t)) + \alpha_2(\phi_{2,ij}(t)) - c_2\phi_{2,ij}(t) + \frac{c_2}{4}(\phi_{2,i-1,j}(t) + \phi_{2,i+1,j}(t) + \phi_{2,ij-1}(t) + \phi_{2,ij+1}(t)) + \beta_2(\phi_{20,ij})
\end{aligned} \tag{4.4}$$

Finally, with $\alpha(\cdot) = \alpha_0$ and $\beta(\cdot) = \beta_0$ a simplified form (with spatial symmetry and isotropy) of the CNN complex cell equation is derived:

$$\begin{aligned}
\frac{d\phi_{1,ij}(t)}{dt} &= \gamma_1 \phi_{2,ij}(t) + (\alpha_1 - c_1) \phi_{1,ij}(t) + \frac{c_1}{4} (\phi_{1,i-1,j}(t) + \phi_{1,i+1,j}(t) + \phi_{1,ij-1}(t) + \phi_{1,ij+1}(t)) + z_{1,ij} \\
\frac{d\phi_{2,ij}(t)}{dt} &= \gamma_2 \phi_{1,ij}(t) + (\alpha_2 - c_2) \phi_{2,ij}(t) + \frac{c_2}{4} (\phi_{2,i-1,j}(t) + \phi_{2,i+1,j}(t) + \phi_{2,ij-1}(t) + \phi_{2,ij+1}(t)) + z_{2,ij} \\
z_{1,ij} &= \beta_1 \phi_{10,ij} + z_1; \quad z_{2,ij} = \beta_2 \phi_{20,ij} + z_2
\end{aligned} \tag{4.5}$$

Making ϕ_{ij} explicitly depend on a state variable ξ_{ij} such as $\phi_{ij} = f(\xi_{ij}) = \text{sigm}(\xi_{ij})$ leads a good approximation to the full-range CNN circuit model ([7], $\text{sigm}(\cdot)$ could be a piecewise linear or monotonic continuous smooth function playing the role of a signal-limiter). The form of the corresponding 2nd order CNN template is as follows:

$$\begin{aligned}
A_{11} &= \begin{bmatrix} 0 & c_1/4 & 0 \\ c_1/4 & \alpha_1 - c_1 & c_1/4 \\ 0 & c_1/4 & 0 \end{bmatrix}; & A_{12} &= \gamma_1; & B_{11} &= \beta_1; & \tau_1 &= 1; & z_1 &= 0 \\
A_{22} &= \begin{bmatrix} 0 & c_2/4 & 0 \\ c_2/4 & \alpha_2 - c_2 & c_2/4 \\ 0 & c_2/4 & 0 \end{bmatrix}; & A_{21} &= \gamma_2; & B_{22} &= \beta_2; & \tau_2 &= 1; & z_2 &= 0
\end{aligned} \tag{4.6}$$

Analysis: It should be noted that the diagonal terms could also be added resulting in a much better spatial isotropy during the diffusion process. Symmetry and spatial isotropy of these templates are due to the fact that there are no *convection terms* in the original formulation (4.1)–(4.2). With these modifications the sign and magnitude of the template entries in (4.6) could be changed in any spatial directions. The time constant is also fixed and equal for the two layers. Having two different time variables (say $t'' \neq t'$) in (4.1) leads to a "double time-scale" in the discretized system $t'' / t' = \tau_2 / \tau_1 \neq 1$ which has practical relevance when looking at various second order models.

4.3 Typical wave classes

There are a number of parameter settings of special interest that lead to very different qualitative behaviors in this symmetric second order system. We have examined and explored primarily the following simple cases (see also Sections 5.2):

- (i) trigger-wave generation (diffusion process in "saturation" at various speeds)
 $c_1 > 0; c_2 > 0; \alpha_1 - (c_1 + 1) > 0; \alpha_2 - (c_2 + 1) > 0; \gamma_1 = \gamma_2 = 0; \beta_1 = \beta_2 = 0; \tau_2 / \tau_1 \neq 1$
- (ii) traveling-wave, spiral-wave and auto-wave generation (spatially interacting trigger-waves)
 $c_1 > 0; c_2 > 0; \alpha_1 - (c_1 + 1) > 0; \alpha_2 - (c_2 + 1) > 0; \gamma_1 \neq 0; \gamma_2 \neq 0; \beta_1 \neq 0; \beta_2 = 0; \tau_2 / \tau_1 \neq 1$
- (iii) retina effects (spatially interacting receptive fields)
 $\alpha_1 \approx c_1 > 0; \alpha_2 \approx c_2 > 0; \gamma_1 > 0; \gamma_2 < 0; \beta_1 \neq 0; \beta_2 \neq 0; \tau_2 / \tau_1 \neq 1$

5 Measurement and computation examples

In the following we present chip measurement of the different active wave phenomena: trigger wave, traveling wave, autowave, and spatio-temporal edge detection. The movie files of the measurements are available at [33]

5.1 Stored programmability

The CACE1k chip can be programmed using the so-called AMC (ANALOGIC MACHINE CODE) language. This is similar to the assembler language but it contains additional high-level instructions, such as: image loading, frame grabber instructions, etc. and built in im-

age processing functions. In the Appendix is a simple AMC program which computes the snapshots of the autowave example (Subsection 5.2.4). The actual computation is performed in line 40. Before this line is the preparation for the computation: setting up the parameters. After the computation is done, the results are read out and displayed from the chip's local analog memories.

The elementary program of the chip is the template. It contains the weight factors of the coupling between the cells and weights for the input and the bias map.

$$\mathbf{A}_1 = \begin{bmatrix} a_{-1,-1}^1 & a_{-1,0}^1 & a_{-1,1}^1 \\ a_{0,-1}^1 & a_{0,0}^1 & a_{0,1}^1 \\ a_{1,-1}^1 & a_{1,0}^1 & a_{1,1}^1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} a_{-1,-1}^2 & a_{-1,0}^2 & a_{-1,1}^2 \\ a_{0,-1}^2 & a_{0,0}^2 & a_{0,1}^2 \\ a_{1,-1}^2 & a_{1,0}^2 & a_{1,1}^2 \end{bmatrix} \quad a_{21} \quad a_{12} \quad (5.1)$$

$$b_1 \quad b_2 \quad z_1 \quad z_2 \quad \tau_1 : \tau_2 = \tau : 1, \quad \tau = 1..16$$

The operation of the array computer is completely determined by the 25 template values, the initial states and boundary conditions. In (5.1) \mathbf{A}_1 and \mathbf{A}_2 matrices denotes the weights inter layer connections of the slower and the faster layer respectively. The strength of the influence of the second layer on the first is controlled by a_{21} and a_{12} stands for the reverse case. Symbols b_1 , b_2 , z_1 and z_2 are the weights of the independent inputs and the space variant bias maps. The ratio of the time constants of the two CNN layers are controlled by τ_1 , τ_2 is fixed. An analogic algorithm is made up of combination of template executions, logic instructions and spatial arithmetic operations. The example shown in the Appendix is a very simple one, it contains only template execution operation.

5.2 Wave phenomenon

This subsection contains the on-chip measurements of the active wave phenomena.

5.2.1 Trigger wave

This example shows a very simple effect: an active (nonlinear) wave in the faster layer initiates a second wave in the slower layer through the positive interlayer coupling. The input is fed to the faster layer. Fig. 6 shows snapshots of the wave evolution.

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} \quad a_{21} = 3.3 \quad a_{12} = 0 \quad (5.2)$$

$$b_1 = 0 \quad b_2 = 3 \quad z_1 = 0.6 \quad z_2 = -1.5 \quad \tau_1 : \tau_2 = 16 : 1$$

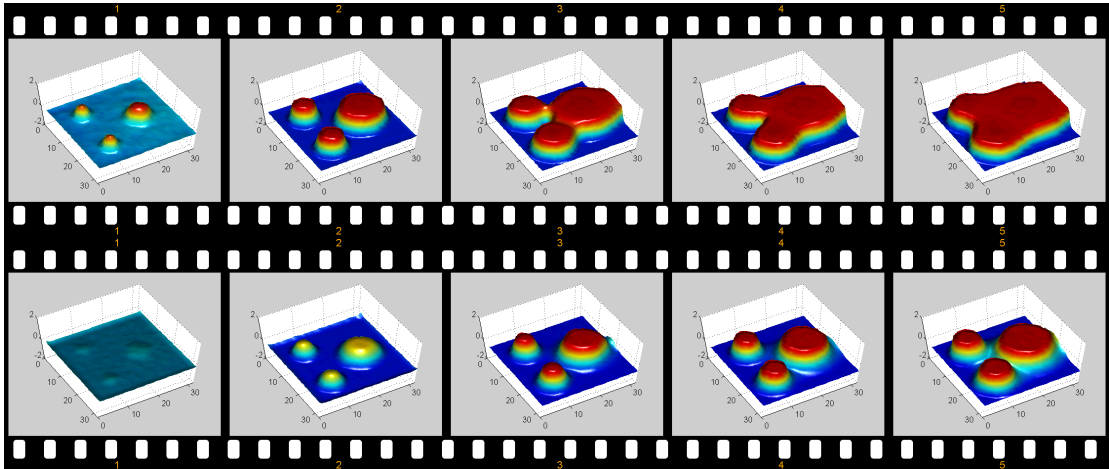


Fig. 6: Trigger wave propagation in the faster and in the slower layer. The first film shows the output of the faster layer.

5.2.2 Erasure effect

This example is an “enhancement” of the trigger wave example: a negative coupling is introduced from the slower layer back to the faster one. As a result, after a while, the trigger wave on the faster layer is erased by the trigger wave from the slower layer. As in the case of the trigger wave the input is given to the faster layer. Fig. 7 shows the snapshots of the wave evolution on the faster layer and Fig. 8 shows the evolution on the slower layer.

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} \quad a_{21} = 3.3 \quad a_{12} = -5.4 \quad (5.3)$$

$$b_1 = 0 \quad b_2 = 3 \quad z_1 = 0.6 \quad z_2 = -2.7 \quad \tau_1 : \tau_2 = 16 : 1$$

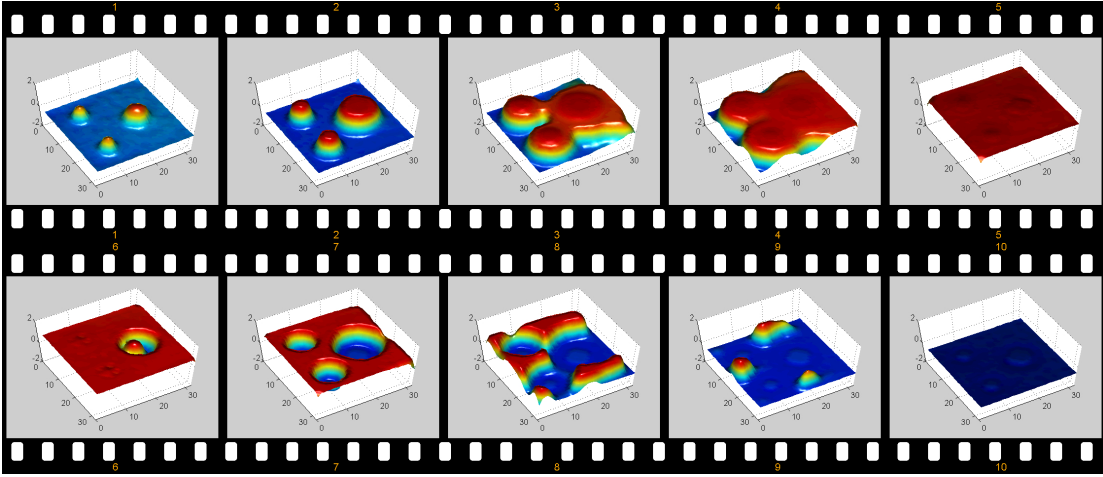


Fig. 7: Retina like wide-field erasure effect, faster layer.

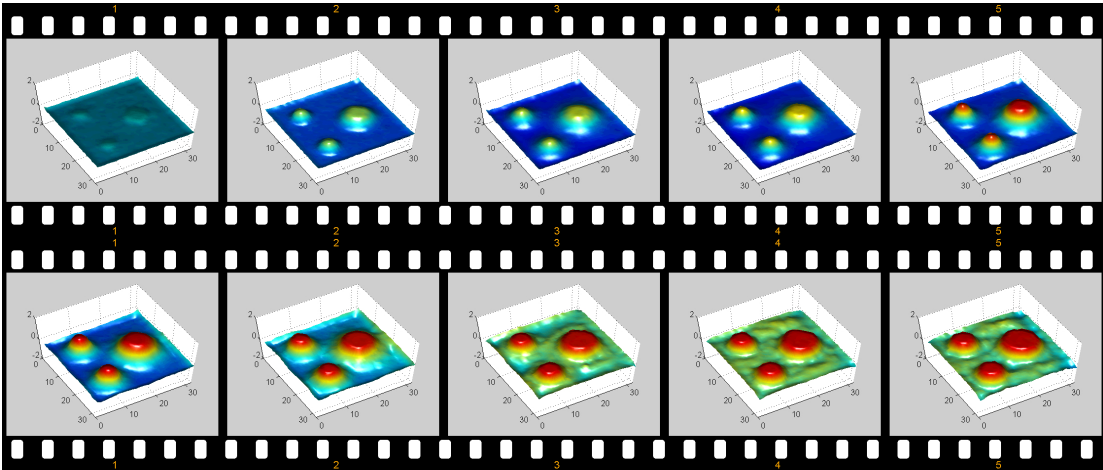


Fig. 8: Retina like wide-field erasure effect, slower layer.

5.2.3 Traveling wave

The characteristic of this wave is that a single wave front travels across the active medium. The autowave is simply formed from the trigger wave template by adding a negative intra-layer coupling from the slower to the faster layer. In Fig. 9 observe the annihilation property of this wave type.

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 4.5 & a_{12} &= -4.2 \\
b_1 &= 0 & b_2 &= 5 & z_1 &= 0.6 & z_2 &= -2.7 & \boldsymbol{\tau}_1 : \boldsymbol{\tau}_2 &= 16 : 1
\end{aligned} \tag{5.4}$$

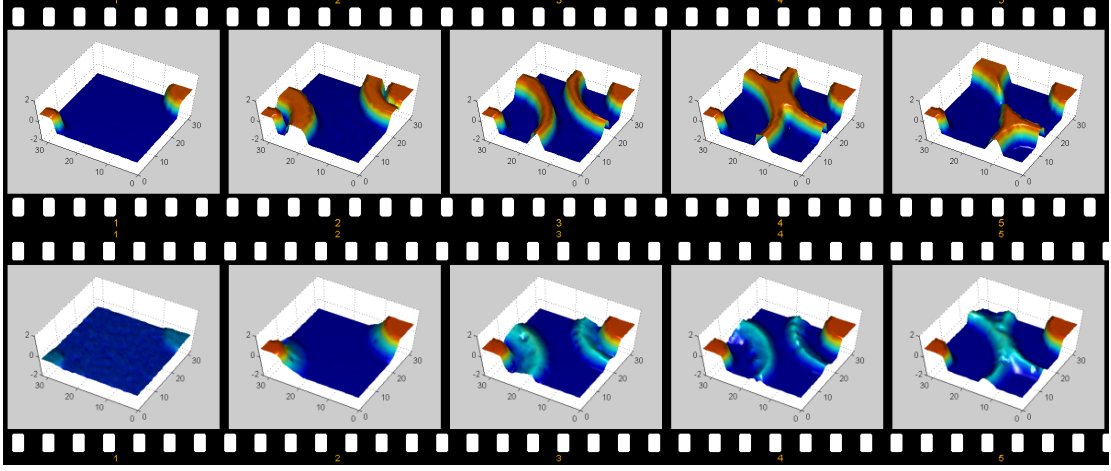


Fig. 9: Travelling wave generation.

5.2.4 Autowave

The autowaves possess all of the properties of the traveling waves; the difference is that in this case multiple wave fronts are generated. See Fig. 10.

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.9 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 5.7 & a_{12} &= -2.4 \\
b_1 &= 0 & b_2 &= 3 & z_1 &= -1.5 & z_2 &= -3.9 & \boldsymbol{\tau}_1 : \boldsymbol{\tau}_2 &= 13 : 1
\end{aligned} \tag{5.5}$$

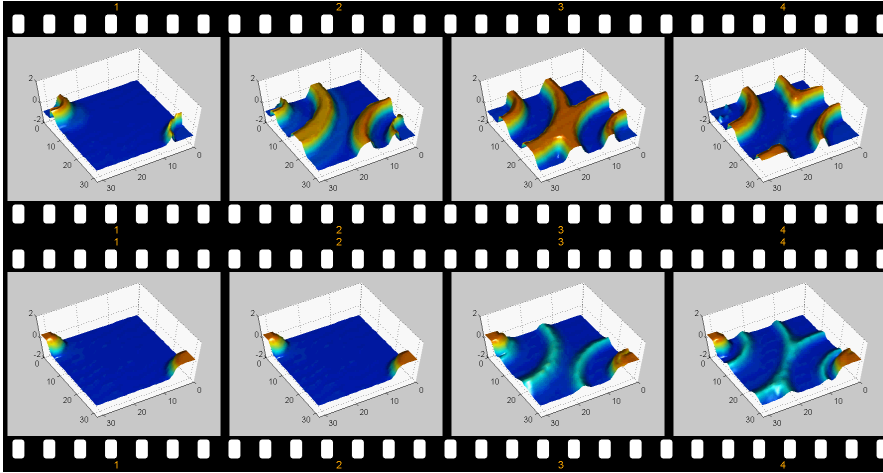


Fig. 10: Autowave propagation CACE1k chip.

5.2.5 Spiral wave

This wave type received its name from the spiral like autowave structure. Fig. 11 shows measurement results. The annihilation of two wave fronts occurs in this case as well.

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.3 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.9 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 6 & a_{12} &= -3 \\
b_1 &= 0 & b_2 &= 3.9 & z_1 &= -4.5 & z_2 &= -3.6 & \boldsymbol{\tau}_1 : \boldsymbol{\tau}_2 &= 8 : 1
\end{aligned} \tag{5.6}$$

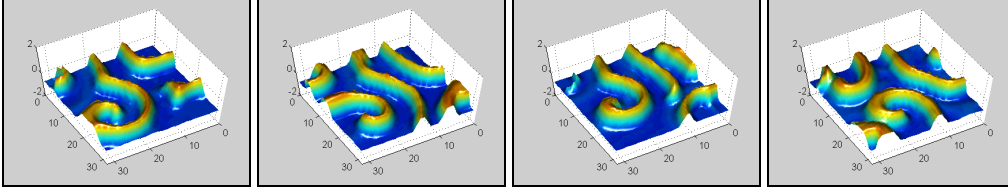


Fig. 11: Spiral wave formation in the CACE1k chip.

5.2.6 Spatio-temporal edge detection

The basic idea of edge detection in the image processing is to somehow extract or enhance the high frequency components of the image. Here this is done with two layers. The slower layer, which gets its input from the faster one, computes the diffusion. The result is subtracted from the faster layer. As a result we get the edge image of the original image. The input was fed to the faster layer. See snapshots of the transient in Fig. 12.

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix} 0.5 & 0.8 & 0.5 \\ 0.8 & -3.6 & 0.8 \\ 0.5 & 0.8 & 0.5 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & a_{21} &= 4.5 & a_{12} &= -5.7 \\
b_1 &= 0 & b_2 &= 3 & z_1 &= -9 & z_2 &= 0 & \boldsymbol{\tau}_1 : \boldsymbol{\tau}_2 &= 6 : 1
\end{aligned} \tag{5.7}$$

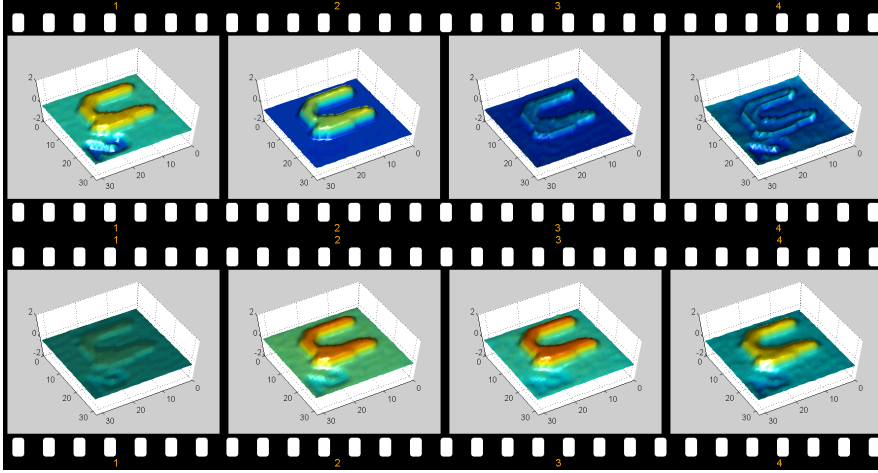


Fig. 12: Edge detection with second order dynamics on the CACE1k chip.

Acknowledgement

This project was supported by: MTA-SZTAKI - Hungarian Academy of Sciences, *Spanish CCICYT (SIVA-2, TICC99-0826), ESPIRIT V (DICTAM-II, IST-1999-19007), ONR/NICOP (N00014-00-1-0429).*

Appendix

TYPICAL AMC PROGRAM EXAMPLE

```
1      ;AUTOWAVE COMPUTING AMC PROGRAM FOR THE CACE1K COMPLEX CELL CHIP
2
3      ;INITIALIZE CYCLE VARIABLE
4      mov.gam_f.gam_f 10 gamf1
5
6      ;LOAD CLONING TEMPLATES
7      host.load.tem autowave.tem tem33 1
8      host.load.tem autowave.tem tem34 2
9
10     ;LOAD IMAGE
11     host.load.pic autowave.bmp lam5
12
13     Loop:
14     ;FILL UP THE NECESSARY MEMORIES
15         ar.layer.param 1 fill_bias -1
16         ar.layer.param 2 fill_bias -1
17         ar.layer.param 1 fill_state 0
18         ar.layer.param 2 fill_state 0
19
20     ;SET THE TIME CONSTANT RATIO OF THE TWO LAYERS (1..16)
21         ar.layer.param 1 layer_c 11
22
23     ;SET BOUNDARY CONDITIONS FOR THE TWO LAYERS
24         ar.layer.param 1 layer_boundary zeroflux
25         ar.layer.param 2 layer_boundary zeroflux
26
27     ;ASSIGN THE LOCAL ANALOG MEMORIES (LAM) TO THE INNER MEMORIES
28         ar.layer.image 1 img_input LAM1
29         ar.layer.image 2 img_input LAM1
30         ar.layer.image 1 img_output LAM3
31         ar.layer.image 2 img_output LAM4
32
33     ;ASSIGN TEMPLATE TO BE EXECUTED
34         mov.tem.tem tem33 TEM1
35
36     ;COPY INPUT IMAGE TO THE CHIP MEMORY
37         mov.lam.lam lam5 LAM1
38
39     ;COMPUTE THE TRANSIENT
40         ar.tem.run TEM1 gamf1 1
41
42     ;READ OUT THE RESULT FROM THE CHIP
43         mov.lam.lam LAM3 lam6
44         mov.lam.lam LAM4 lam7
45
46     ;DISPLAY IT
47         host.display lam6 1
48         host.display lam7 2
49
50     ;INCREASE CYCLE COUNTER
51         sc.add.gam_f 50 gamf1 gamf1
52         sc.rel.gt 3500.0 gamf1 glm1
53         jumpc glm1 Loop
54
55     end
```

References

- [1] T. Roska and L.O. Chua: "The CNN Universal Machine: An Analogic Array Computer". *IEEE Transactions on circuits and Systems-II: Analog and Digital Signal Processing*, Vol.40, No.3, pp.163-173, March 1993.
- [2] L.O. Chua and L. Yang, "Cellular Neural Networks: Theory", *IEEE Trans. on Circuits and Systems*, Vol.35. pp. 1257-1272, 1988.
- [3] L.O. Chua and L. Yang, "Cellular neural networks: Applications", *IEEE Trans. on Circuits and Systems*, Vol.35. pp. 1273-1290, 1988.
- [4] L.O. Chua and T. Roska, "The CNN paradigm", *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, Vol.40, No. 3, pp. 147-156, 1993.
- [5] Cs. Rekeczky, T. Serrano-Gotarredona, T. Roska and A. Rodríguez-Vázquez, "A Stored Program 2nd Order/3-Layer Complex Cell CNN-UM". *Proc. of the Sixth IEEE International Workshop on Cellular Neural Networks and their Applications*, pp.219-224, Catania, Italy, May 2000.
- [6] Cs. Rekeczky, B. Roska, E. Nemeth and F. Werblin, "Neuromorphic CNN Models for Spatio-Temporal Effects Measured in the Inner and Outer Retina of Tiger Salamander". *Proc. of the Sixth IEEE International Workshop on Cellular Neural Networks and their Applications*, pp.15-20, Catania, Italy, May 2000.
- [7] S. Espejo, R. Carmona, R. Domínguez-Castro and A. Rodríguez-Vázquez, "A VLSI Oriented Continuous-Time CNN Model". *International Journal of Circuit Theory and Applications*, John Wiley & Sons. Vol.24, No.3, pp.341-356, May-June 1996.
- [8] R. Carmona, *Analysis and Design of CNN-based VLSI Hardware for Real-Time Image Processing*. Ph. D. Dissertation. University of Seville, June 2002.
- [9] Richard C. Jaeger, "Tutorial: Analog Data Acquisition Technology. Part I – Digital-to-Analog Conversion". *IEEE Micro*, Vol.24, No.3, pp.20-37, May 1982.
- [10] G. Liñán, S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez, "ACE4k: An analog I/O 64×64 visual microprocessor chip with 7-bit analog accuracy". *International Journal of Circuit Theory and Applications*. Vol. 30, No. 2-3, pp. 89-116, March 2002.
- [11] R. Carmona, F. Jiménez-Garrido, R. Domínguez-Castro, S. Espejo and A. Rodríguez-Vázquez, "Bio-inspired Analog VLSI Design Realizes Programmable Complex Spatio-Temporal Dynamics on a Single Chip". *Proceedings of the Design, Automation and Test in Europe Conference (DATE'02)*, pp. 362-366. Paris, France, March 2002. Published by the IEEE Computer Society, Los Alamitos, CA, ISBN 0-7695-1471-5.
- [12] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez and R. Carmona, "Four-Quadrant One-Transistor Synapse for High Density CNN Implementations". *Proceedings of the 5th IEEE Workshop on Cellular Neural Networks and Applications (CNSA'98)*, pp. 243-248, London, UK, April 1998.
- [13] B. Roska and F. S. Werblin, "Vertical Interactions across Ten Parallel Stacked Representations in Mammalian Retina", *Nature*, Vol. 410, pp. 583-587, 2001
- [14] D. Bálya, B. Roska, T. Roska, F. S. Werblin: "A CNN framework for modeling parallel processing in a mammalian retina", *International Journal on Circuit Theory and Applications (CTA)*, Vol. 30, 2002
- [15] L. O. Chua and T. Roska: *Cellular Neural Networks and Visual Computing*, Cambridge University Press, Cambridge, UK, 2002

- [16] Cs. Rekeczky, B. Roska, E. Nemeth and F. S. Werblin “The network behind spatio-temporal patterns.” *Int. J. Circuit Theory and Applications*, Vol. 29, pp. 197-239, 2001
- [17] E. A. Kandel, J. A. Schwartz, and T. M. Jessell (editors): *Essentials of Neural Science and Behavior*, Prentice Hall International, 1995
- [18] F. S. Werblin and A. Jacobs: “Using CNN to unravel space-time processing in the vertebrate retina”, *Proc. of CNNA-94*, pp.33-40, 1994
- [19] F. S. Werblin, T. Roska and L. O. Chua “The analogic cellular neural network as a bionic eye”, *Int. J. on Circuit Theory and Applications*, Vol. 23, pp. 541-569, 1995
- [20] B. Roska, E. Nemeth, L. Orzo, F. Werblin: “Three Levels of Lateral Inhibition: A Space-time Study of the Retina of the Tiger Salamander”, *Journal of Neuroscience*: pp.1941-1951, 2000
- [21] T. Roska and L. O. Chua “The CNN universal machine: an analogic array computer”, *IEEE Trans. on Circuits and Systems II*, Vol. 40, pp. 163-173, 1993
- [22] D. Bálya, B. Roska, T. Roska, F. S. Werblin: “A CNN model framework and simulator for biologic sensory systems”, *Proc. ECCTD-2001*, 2001; pp. I-357-360.
- [23] D. Bálya, Cs. Rekeczky and T. Roska: “RMRM”, *Proc. IEEE ISCAS-2002*, pp. IV-361-364, Phoenix
- [24] Complex-cell chip ISCAS-2002 Carmona G R, Jiménez Garrido F, Domínguez Castro R, Espejo Meana S, Rodríguez Vázquez A (2002): *CACE1K User’s Guide*, Instituto de Microelectrónica de Sevilla
- [25] Roska T, Kék L, Nemes L, Zarándy A, and Szolgay P (1999) *CNN Software Library (Templates and Algorithms) Version 7.3*. Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, Hungary
- [26] [P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 629-639, July 1990.
- [27] [N. Nordström, "Biased Anisotropic Diffusion - A Unified Regularization and Diffusion Approach to Edge Detection", *Image Vision Computers*, Vol. 8, No 4, pp. 318-327, July 1990.
- [28] [M. Nitzberg and T. Shiota, "Nonlinear Image Smoothing with Edge and Corner Enhancement", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 8, August 1992.
- [29] [G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz, "Nonlinear Anisotropic Filtering of MRI Data", *IEEE Transactions on Medical Imaging*, Vol. 11, No. 2, June, 1992.
- [30] [F. Catté, P. L. Lions, J. M. Morel, and T. Coll, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion", *SIAM Journal on Numerical Analysis*, Vol. 29, No. 1, pp. 182-193, 1992.
- [31] [L. Alvarez, P. L. Lions, and J. M. Morel, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. II", *SIAM Journ. on Num. Analysis*, Vol. 29, No. 3, pp. 845-866, 1992.
- [32] B. M. H. Romeny (ed.), *Geometry-driven Diffusion in Computer Vision*, Kluwer Academic Publishers, 1994.
- [33] <http://www.sztaki.hu/~petras/cace1k>
- [34] L. O. Chua "Passivity and complexity," *IEEE Trans. Circuits Syst. I, Fundamental Th. Appl.* 46(1), 71-82, 1999.