Bio-Inspired Analog VLSI Design Realizes Programmable Complex Spatio-Temporal Dynamics on a Single Chip

R. Carmona, F. Jiménez-Garrido, R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez
Instituto de Microelectrónica de Sevilla. IMSE-CNM-CSIC
Avda. Reina Mercedes s/n 41012 Sevilla (SPAIN)
Tel.:+34955056666, Fax: +34955056686
E-mail: rcarmona@imse.cnm.es

Abstract[i]

A bio-inspired model for an analog parallel array processor (APAP), based on studies on the vertebrate retina, permits the realization of complex spatio-temporal dynamics in VLSI. This model mimics the way in which images are processed in the visual pathway what renders a feasible alternative for the implementation of early vision tasks in standard technologies. A prototype chip has been designed in $0.5\mu m$ CMOS. Design challenges, trade-offs and the building blocks of such a high-complexity system (0.5×10^6 transistors, most of them operating in analog mode) are presented in this paper.

1. Bio-inspired APAP model

1. 1. Sketch of the biological retina

The vertebrate retina has the structure displayed in Fig. 1. A first layer of photodetectors at the top, the cone cells, captures light and converts it to activation signals [1]. Bipolar cells carry them across the layers to the ganglion cells that interface the retina with the optical nerve, in a trip of several micrometers. The ganglion cells convert the continuous activation signals, proper of the retina, to pulse-like action potential signals that can be transmitted over longer distances by the nervous system. In the way to the ganglion cells, the information carried by bipolar cells is affected by the operation of horizontal and amacrine cells. They form layers in which signals are weighted and promediated to bias photodetectors and to inhibit the vertical pathway. These operations have a local scope and depend on the recent history. Once adaptation is achieved, patterns of activity are formed dynamically by the presence or absence of visual stimuli. Inhibition is and transmitted laterally through the horizontal and amacrine cells.

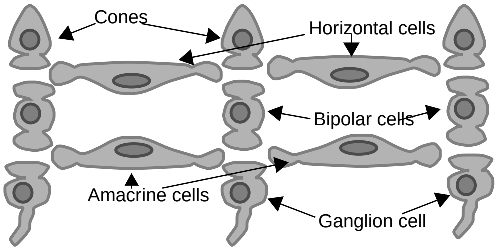


Fig. 1. Conceptual diagram of the retina.

There are, in this description, some interesting aspects of each retinal layer that markedly resemble the characteristics of the Cellular Neural Networks (CNNs) [2]: 2D aggregations of continuous signals, local connectivity between elementary nonlinear processors, analog weighted interactions between them. Motivated by these coincidences, a model for the operations of the biological retina based on CNNs have been developed.

1. 2. CNN-based analogy

Based on measurements of the response of the inner and outter plexiform layers of the retina, a complex-cell CNN-based chip has been proposed [3]. This 2nd-order 3-layer CNN cell consists of 2 CNN layers coupled by some inter-layer weights and an additional layer incorporating analog arithmetics to combine the outputs of the dynamically linked layers (Fig. 2). The cells in the two first layers have a first order core, while the third layer, that can be also modeled in this way, has much faster dynamics

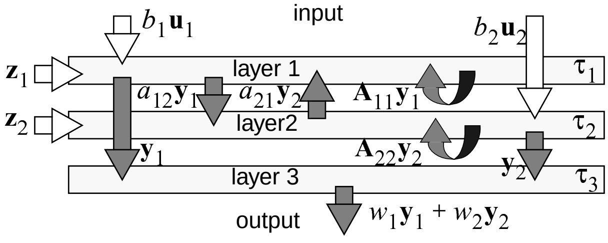


Fig. 2. Diagram of the 2nd-order CNN.

[i] This work has been supported by ESPRIT V Project IST-1999-19007 and by ONR/NICOP Grant N-00014-00-1-0429, and the Spanish CICYT Proyect TIC-1999-0826.

($\tau_3 \ll \tau_1, \tau_2$). Complex dynamics can be programmed via the adjustment of the intra- and inter-layer coupling strengths. The evolution law of each cell, $C(i, j)$, is given by two coupled differential equations:

$$\begin{aligned}\tau_1 \frac{dx_{1,ij}(t)}{dt} &= -g[x_{1,ij}(t)] + b_{11,00}\cdot u_{1,ij} + z_{1,ij} + \\ &+ \sum_{k=-r_1}^{r_1}\sum_{l=-r_1}^{r_1} a_{11,kl}\cdot y_{1,(i+k)(j+l)} + a_{12}\cdot y_{2,ij} \\ \tau_2 \frac{dx_{2,ij}(t)}{dt} &= -g[x_{2,ij}(t)] + b_{22,00}\cdot u_{2,ij} + z_{2,ij} + \\ &+ \sum_{k=-r_2}^{r_2}\sum_{l=-r_2}^{r_2} a_{22,kl}\cdot y_{2,(i+k)(j+l)} + a_{21}\cdot y_{1ij}\end{aligned} \qquad (1)$$

where the nonlinear losses term and the output function in each layer are those of the FSR CNN model [4]:

$$g(x_{n,ij}) = \lim_{m\to\infty}\begin{cases} mx_{n,ij} & \text{if} \quad x_{n,ij} > 1 \\ x_{n,ij} & \text{if} \quad |x_{n,ij}| \le 1 \\ -mx_{n,ij} & \text{if} \quad x_{n,ij} < -1 \end{cases} \qquad (2)$$

and $y_{n,ij} = f(x_{n,ij}) = \frac{1}{2}(|x_{n,ij}+1| - |x_{n,ij}-1|)$ (3)

2. APAP architecture

2. 1. Prototype chip floorplan

The proposed chip consists in a APAP of 32×32 identical cells (Fig. 7). It is surrounded by the circuits implementing the boundary conditions for the CNN dynamics. There is also an I/O interface, a timing and control unit and a program memory. The I/O interface consists in a serializing-deserializing analog multiplexor. The program memory is composed of 24 blocks of DRAM of 64 bytes of capacity, 1kB dedicated to the analog program, and 0.5kB to the logic program. In addition, 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 derived from electromagnetic interference, voltage drops in long metal lines carrying currents can be quite noticeable. Thus, signal buffering and low-resistance paths must be provided to avoid this. Finally, the timing unit is composed by an internal clock/counter and a set of FSMs that generate the internal signals that enable the processes of images up/downloading and program memory accesses.

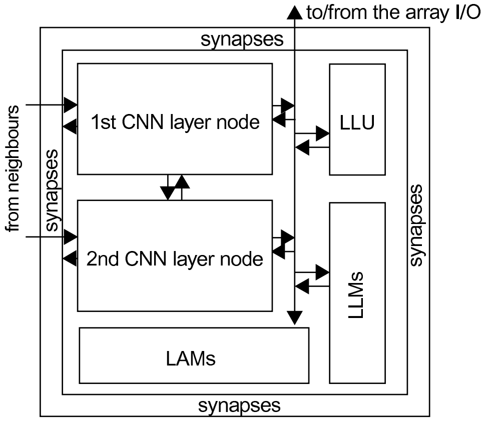


Fig. 3. Conceptual diagram of the basic cell

2. 2. Basic cell scheme

The elementary processor of the CNN-based APAP includes two coupled continuous-time CNN cores (Fig. 3). 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 contains also 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 intracell 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.

The internal structure of each CNN core is depicted in the diagram of Fig. 4. They receive contributions from the rest of the processing nodes in the neighbourhood 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 feedforward action of the stored input and bias patterns. There is a voltage limiter for implementing the FSR CNN model. The state variable is transmitted in voltage form to the synaptic blocks, in the periphery of the cell, where weighted contributions to the neighbours' 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. Analog building blocks for the basic cell

3. 1. Single-transistor synapse

The synapse is a four-quadrant analog multiplier. Their inputs will be the cell state or input and the weight volt-

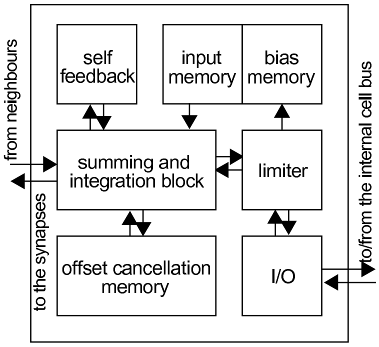

self feedback
input memory
bias memory
from neighbours
summing and integration block
limiter
to the synapses
offset cancellation memory
I/O
to/from the internal cell bus

Fig. 4. Internal structure of each CNN layer node.

ages, while the output will be the cell's current contribution to a neighbouring cell. It can be achieved by a single transistor biased in the ohmic region [5]. For a PMOS with gate voltage $V_X = V_{x_0} + V_x$, and the p-diffusion terminals at $V_W = V_{w_0} + V_w$ and V_w, the drain-to-source current is:

$$I_o = -\beta_p V_w V_x - \beta_p V_w \left(V_{x_0} + \left| \hat{V}_{T_p} \right| - V_{w_0} - \frac{V_w}{2} \right) \quad (4)$$

which is a four-quadrant multiplier with an offset term that is time-invariant —at least during the evolution of the network— and not depending on the cell state. This offset that can be eliminated by a calibration step, with the help of a current memory. Fig. 5 shows the simulated characteristic of the multiplier (using Hspice v. 99.2 and a MOS model of level 49).

3. 2. Current conveyor and level shifting

For the synapse to operate properly, the input node of the CNN core must be kept at constant voltage, independently of what current is entered. This is achieved by a current conveyor (Fig. 6). Any difference between the voltage at node ⓛ and the reference V_{w_0} is amplified and the negative feedback corrects the deviation. Notice that a voltage offset in the amplifier results in an error of the same order. Using the offset cancellation mechanism in Fig. 6 the total current injected into the load is offset-free:

$$I_L = I_o + I_{\text{mem}} - I_{\text{b}} = g_m v_d \quad (5)$$

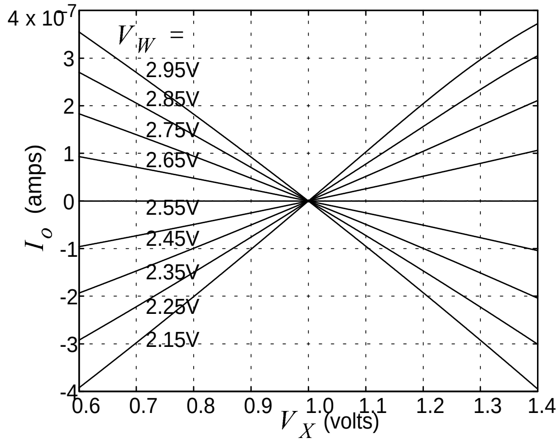


Fig. 5. Multiplier output vs. V_X w/o offset.

3. 3. S^{3}I current memory

As it has been referred, the offset term of the synapse current must be removed for its output current to represent the result of a four-quadrant multiplication. For this purpose all the synapses are reset to $V_X = V_{x_0}$. Then the resulting current, which is the sum of the offset currents of all the synapses concurrently connected to the same node, is memorized. This value will be substracted on-line from the input current when the CNN loop is closed, resulting in a one-step cancellation of the errors of all the synapses. The validity of this method relies in the accuracy of the current memory. For instance, in this chip, the sum of all the contributions will range, for the applications for which it has been designed, from $18\mu A$ to $46\mu A$. On the other side, the maximum signal to be handled is $1\mu A$. If a signal resolution of $8b$ is pretended, then $0.5LSB = 2nA$. Thus, our current memory must be able to distinguish $2nA$ out of $46\mu A$. This represents an equivalent resolution of $14.5b$. In order to achieve such accuracy level, a S^3I current memory is used. It is composed by three stages (Fig. 6), each one consisting in a switch, a capacitor and a transistor. I_B is the current to be memorized. After memorization the only error left corresponds to the last stage. The former stages do not contribute to the error in the memorized current. If the S^3I block is designed so as to store the most significant bits in the first capacitor, and the less significant bits in the last one, the error can be made quite small.

3. 4. Time-constant scaling

The differential equation that governs the evolution of the network (1) can be written as a sum of current contributions injected to the state capacitor. Scaling up/down this sum of currents is equivalent to scaling the capacitor and, thus, speeding up/down the network dynamics. Therefore, scaling the input current with the help of a current mirror, for instance, will have the effect of scaling the time-constant. A circuit for continuously adjusting the current gain of a mirror can be designed based on a regulated-Cascode current mirror in the ohmic region. But the strong dependence of the ohmic-region biased transistors on the power rail voltage causes mismatches in τ between cells in the same layer. An alternative to this is a binary programmable current mirror. It trades resolution in τ for robustness, hence, the mismatch between the time constants of the different cells is now fairly attenuated.

A new problem arises, though, because of current scaling. If the input current can be reshaped to a 16-times smaller waveform, then the current memory has operate over larger and the smaller signals. But, if designed to operate on large currents, the current memory will not work for the tiny currents of the scaled version of the input. If it is designed to run on small input currents, long transistors will be needed, and the operation will be unreliable for the larger currents. One way of avoiding this situation is to make the S^{3}I memory to work on the original unscaled version of the input current. Therefore, the adjustable-time-constant CNN core will be a current conveyor, followed by the S^{3}I current memory and then the binary weighted current mirror. The problem now is that the offsets introduced by the scaling block add up to the signal and the required accuracy levels can be lost. Our proposal is depicted in Fig. 6. It consists in placing the scaling block (programmable mirror) between the current conveyor and the current memory. In this way, any offset error will be cancelled at the auto-zeroing phase. In the picture, the voltage reference generated with the current conveyor, the regulated-Cascode current mirrors and the S^{3}I memory can be easily identified. The inverter, A_i, driving the gates of the transistors of the current memory is required for stability. Without it, the output node, Ⓐ, will diverge from the equilibrium. The critical aspects of this circuit are related with the feedback loop formed by M_{p1}, M_{p2}, M_{n2}, the inverting amplifier A_i and the transistors M_m, when sensing the offset current. During this process the output current I_o is zero because the current path to the state capacitor is open.

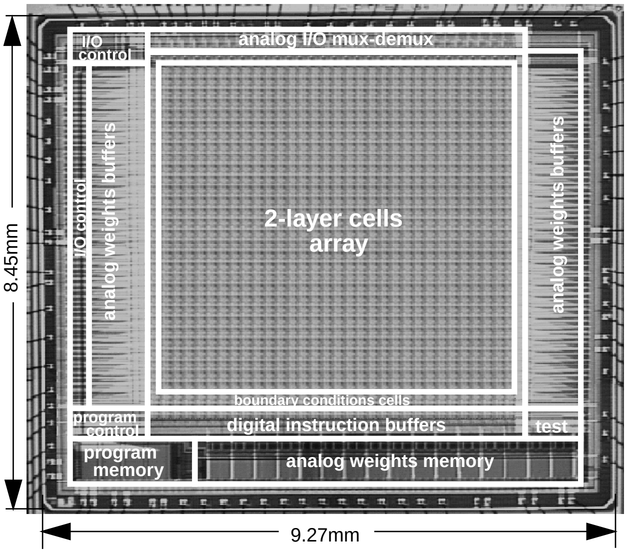


Fig. 7. Prototype chip photograph

4. Chip data and simulations

A prototype chip has been designed and fabricated in a 0.5µm single-poly triple-metal CMOS technology. Its dimensions are 9.27×8.45 sq. mm (photograph in Fig. 7). The cell density achieved is $29.24 \text{cells} / \text{mm}^2$. The programmable dynamics of the chip permit the observation of different phenomena of the type of propagation of waves, pattern generation, etc. Fig. 8 displays the evolution of the state variable in a reduced network, 4×4 cells, in which 1-D anisotropic diffusion (shadowing) has been programmed to run at different speeds on each layer. By controlling the network dynamics and combining the results with the help of the built-in local logic and arithmetic operators, rather involved image processing tasks can be programmed. For instance, tuning to the appropriate template elements [3] (the analog program) allows gray-scale contour detection (Fig. 9).

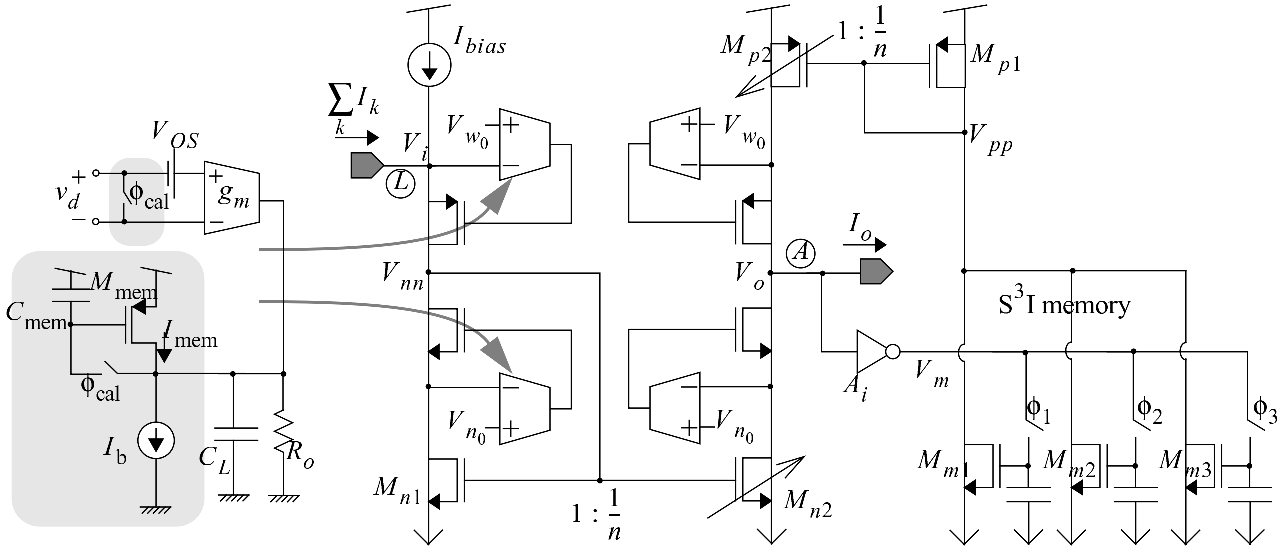
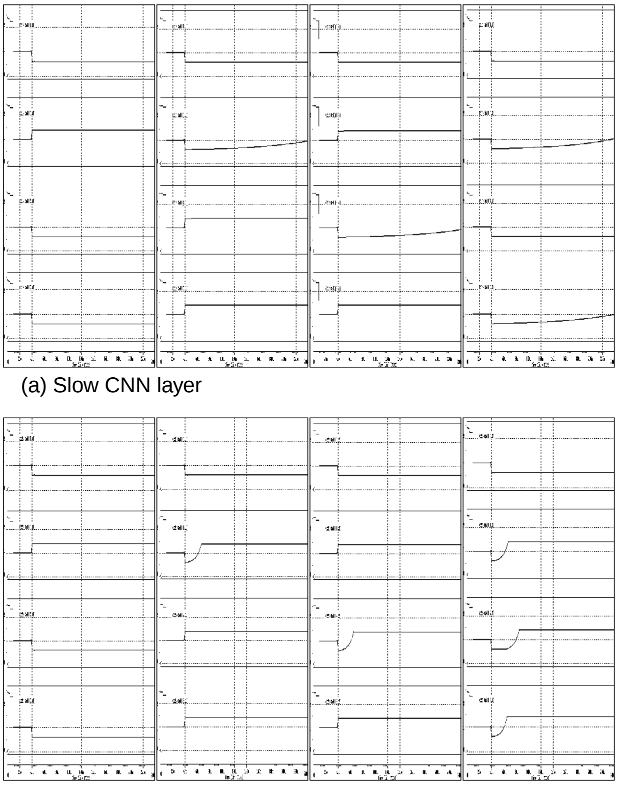


Fig. 6. Input block with current scaling, S^{3}I memory and offset-corrected OTA schematic.



(a) Slow CNN layer

(b) Fast CNN layer

Fig. 8. 1-D shadowing.

5. Conclusions

The proposed approach supposes a promising alternative to conventional digital image processing for applications related with early-vision and low-level focal-plane image processing. Based on a simple but precise model of part of the real biological system, a feasible efficient implementation of an artificial vision device has been designed. The peak operation speed of the chip will outdone its digital counterparts due to the fully parallel nature of the processing, which is, once more, based on the analogy not on the simulation.

References

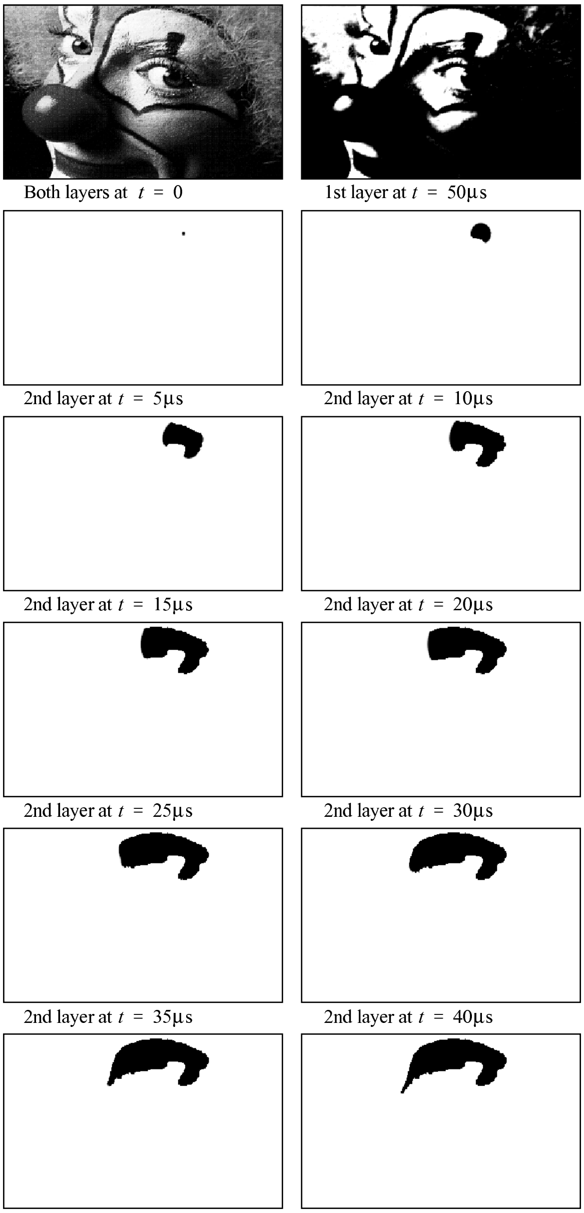
[1] F. werblin, “Synaptic Connections, Receptive Fields and Patterns of Activity in the Tiger Salamander Retina”, *Investigative Ophthalmology and Visual Science,* Vol. 32, No. 3, pp. 459-483, March 1991.

[2] F. Werblin, T. Roska and L. O. Chua, “The Analogic Cellular Neural Network as a Bionic Eye”. *International Journal of Circuit Theory and Applications,* Vol. 23, No. 6, pp. 541-69, November-December 1995.

[3] 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.

[4] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro and R. Carmona, “Convergence and Stability of the FSR CNN Model”. *Proceedings of the 3rd International Workshop on Cellular Neural Networks and their Applications,* pp. 411-417, Rome, December 1994.

[5] R. Domínguez-Castro, A. Rodríguez-Vázquez, S. Espejo and R. Carmona, “Four-Quadrant One-Transistor Synapse for High Density CNN Implementations”. *Proc. of the Fifth IEEE International Workshop on Cellular Neural Networks and their Applications,* pp. 243-248, London, UK, April 1998.



Both layers at $t = 0$

1st layer at $t = 50\mu s$

2nd layer at $t = 5\mu s$

2nd layer at $t = 10\mu s$

2nd layer at $t = 15\mu s$

2nd layer at $t = 20\mu s$

2nd layer at $t = 25\mu s$

2nd layer at $t = 30\mu s$

2nd layer at $t = 35\mu s$

2nd layer at $t = 40\mu s$

2nd layer at $t = 45\mu s$

2nd layer at $t = 50\mu s$

Fig. 9. Contour detection example.