CELLULAR NEURAL NETWORKS WITH NONLINEAR AND DELAY-TYPE TEMPLATE ELEMENTS

Tamás Roska[+] and Leon O. Chua

Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory, University of California, Berkeley, CA 94720

[+] *on leave from the Computer and Automation Institute of the Hungarian Academy of Sciences, Úri-u.49, Budapest, H-1014*

ABSTRACT

The cellular neural network (CNN) paradigm is a powerful framework for analog nonlinear processing arrays placed on a regular grid. In this paper we extend the current repertoire of CNN cloning template elements (atoms) by introducing additional nonlinear and delay-type characteristics. With this generalization, several well-known and powerful analog array-computing structures can be interpreted as special cases of the CNN. Moreover, we show that the CNN with these generalized cloning templates has a general programmable circuit structure with analog macros and algorithms. The relations with the cellular automaton (CA) and the systolic arrays (SA) are analysed. Finally, some robust stability results and the state space structure of the dynamics are presented.

I. INTRODUCTION

The cellular neural network (CNN) paradigm [1] provides a powerful analog nonlinear computing structure for a variety of array computations. Array computations can be defined as the parallel execution of complex operations on a large number of processors placed on a geometrically regular grid. If the operations are logical (involving only a few bits) the classical cellular automaton (CA) of John von Neumann and its recent variants are the ideal tools. If the operations are numerical (typically 8, 16 or 32 bits) the systolic arrays are the best current approach. In both cases the computing devices are digital processors. If, however, the signal values are continuous and/or analog real-time operations are performed, then the CNN is the optimal solution with respect to both area and speed. Its structural simplicity, robust stability and function variability are all ideal for VLSI implementation [2,3,4].

In this paper we show that by retaining all the important features of the original CNN structure and by introducing very simple nonlinear and delay-type templates, the CNN becomes a powerful framework for general analog array dynamics.

From a theoretical point of view it has well defined qualitative properties and it contains, as special cases, a variety of powerful practical image processing solutions (both artificial and biologically motivated applications, e.g. the "silicon retina" and other resistive grids [5,6,7,8,9]. From a practical point of view it provides a natural paradigm for designing programmable analog VLSI or opto-electronic real-time computing arrays.

In Section II we present the general framework and show the simple programmability properties inherent in the CNN structure. In Section III we present some special cases which are practically important and promising. In section IV we show that our CNN is a general programmable analog array computer. With their simplest template elements (atoms), like the gates, we can define the CNN templates as macros and introduce the notion of CNN algorithms based on these macros.

In Section V we present some robust stability results for a class of nonsymmetric nonlinear CNN templates, aswell as some other qualitative properties. In Section VI we summarize our conclusions and pose some open questions and new directions.

II. THE GENERAL CNN FRAMEWORK WITH NONLINEAR AND DELAY-TYPE TEMPLATES

Consider a two-dimensional grid, such as those shown in Figure 1. The small squares represent the processors and their connections show the possible communication lines (feedback and feed forward connections) with their neighbors. Here, the solid lines correspond to the nearest neighbor interconnections. This two-dimensional grid of processors is called a layer. Multiple layers of processors are also allowed and can be defined similarly as the multilayer cellular neural networks in [1].

The analog processor element, the cell, is shown in Figure 2. It differs from that of the original cell (Figure 3 on p. 1258 in [1]) basically in the controlled sources Ixy and Ixu only.

Namely, instead of the two linear controlled sources defined by

$$A(ij;kl)v_{ykl} \quad \text{and} \quad B(ij;kl)v_{ukl}$$

associated with a typical cell C_{ij} and a typical interacting cell C_{kl}, we allow nonlinear and delayed controlled sources defined by

$$\hat{A}_{ij;kl}(v_{ykl},v_{yij}) + A^{\tau}{}_{ij;kl}v_{ykl}(t-\tau)$$

and

$$\hat{B}_{ij;kl}(v_{ukl},v_{uij}) + B^{\tau}{}_{ij;kl}v_{ukl}(t-\tau)$$

That is, instead of having a linear VCCS (voltage controlled current source) in the A and B cloning templates, we now have a nonlinear and/or delay-type VCCS. The structure of the nonlinearity in the templates is also important: it is a function of at most two variables, namely, the output voltage of cell C_{ij} and that of a neighbor C_{kl}.

We also allow the output function to have a wider range of output voltages so that the saturation voltage is $\pm K$ instead of ± 1. However, these slight changes does extend the class of CNN dynamics significantly.

Again, with MxN cells in an array, we define the r-neighborhood of the cell C_{ij} as

$$N_r(ij) = \{ C_{kl}: \max(|k-i|,|l-j|) = r \text{ (integer)}\} \qquad (1)$$

Let us now formulate the canonical equations describing the CNN analog nonlinear processor array with nonlinear and delay-type templates. Henceforth, we will use the same term CNN in this generalized concept and refer to a linear template CNN when only linear controlled sources are being used in the cloning templates. Without loss of generality we assume that the processor cells are drawn on a rectangular grid with MxN cell units, $1<i<M$, $1<j<N$.

Referring to Figures 1 and 2, we can define a CNN mathematically as follows:

1. State equation

$$C\,\dot{v}_{xij} = -(1/R_x)\,v_{xij}(t) + I + \sum_{C_{kl}\in N_r(ij)} \hat{A}_{ij;kl}\,(v_{ykl}(t), v_{yij}(t)) + \sum_{C_{kl}\in N_r(ij)} \hat{B}_{ij;kl}\,(v_{ukl}(t), v_{uij}(t)) + \sum_{C_{kl}\in N_r(ij)} A^{\tau}_{ij;kl}\,v_{ykl}(t-\tau) + \sum_{C_{kl}\in N_r(ij)} B^{\tau}_{ij;kl}\,v_{ukl}(t-\tau) \qquad (2a)$$

where A, B and A^{τ}, B^{τ} are associated with the nonlinear and the delay-type cloning templates, respectively. In particular, $A_{ij;kl}$, $B_{ij;kl}$ are continuous functions of at most two variables and the $A^{\tau}_{ij;kl}$, $B^{\tau}_{ij;kl}$ are real constants.

For example, with $d_1 = c_1(\exp(v_{ykl}) - 1)$, $d_2 = c_2(v_{ykl} - v_{yij})$ the cloning templates can be defined symbolically as the two nonlinear cloning templates

$$\hat{A} = \begin{bmatrix} 0 & d_1 & 0 \\ d_1 & 2 & d_1 \\ 0 & d_1 & 0 \end{bmatrix} \qquad \hat{B} = \begin{bmatrix} 0 & 0 & 0 \\ d_2 & 1 & d_2 \\ 0 & 0 & 0 \end{bmatrix}$$

and $I=0$.

Here, we have tacitly assumed that the templates are space invariant, though, they could be space variant too. Henceforth, we use space invariant templates.

2. Output equation

The memoryless output function

$$v_{yij}(t) = f(v_{xij}(t)) \qquad (2b)$$

is depicted on Figure 3. It is a piecewise-linear characteristic with unity slope in the range (-K, K), K>0. It can be approximated to within any precision by a smooth (C^1) strictly monotone-increasing sigmoid function. This approximation is sometimes desirable in analytical proofs where the C^1 condition is more convenient to work with. It is also a more realistic assumption since the physically realized characteristic is actually C^1.

As a further generalization, we can allow the output function to have its own dynamics, too. For example, Eq.(2b) can be replaced by a first-order state equation

$$\dot{v}_{yij} = -v_{yij} + f(v_{xij}(t)) \qquad (2b')$$

or by a higher-order state equation.

Moreover, remaining monotone increasing, f might be more complicated.

3. Input equation:

$$v_{uij} = E_{ij} \tag{2c}$$

4. Constraint equations:

$$|v_{xij}(0)| \leq 1 \tag{2d}$$

$$|v_{uij}| \leq 1 \tag{2e}$$

5. Parameter assumptions:

$$K \geq 0, \qquad C \geq 0, \qquad R_x > 0, \qquad \tau \geq 0 \tag{2g}$$

In the case when $\hat{A}_{ij;kl}$, $\hat{B}_{ij;kl}$ are linear functions (of a single variable) and $\tau = 0$ or, equivalently,

$$\hat{A}_{ij;kl} = A_{ij;kl}\, v_{ykl}$$

$$\hat{B}_{ij;kl} = B_{ij;kl}\, v_{ukl}$$

and $A^{\tau}ij;kl$, $B^{\tau}ij;kl$ are equal to zero, we have the original CNN [1] with linear cloning templates.

Observe that the structure of the original CNN is retained and that the nonlinear as well as the delay-type cloning template funtions are both voltoge controlled current sources (VCCS). These properties are very useful to the circuit layout and the software programmability of the various representations of this quite general CNN architecture. These important practical considerations will be discussed in detail in Section IV.

As to the information processing or computing function of the CNN array, it has two input ports and one output port per cell. Namely, besides the generic input $E_{ij}(t) = v_{uij}(t)$ the initial value of the state, i. e. $v_{xij}(0)$, can also be used as an input. The output $v_{yij}(t)$ is defined as either the dc steady state $v_{yij}(oo)$, or as a snapshot at a given time instant $t = T$, i.e. $v_{uij}(T)$. In the former case, the successive inputs can be applied in a sampled- data mode allowing a sufficient settling time to elapse between the samples. It is clear that the most direct application area of the CNN is in image processing (where the light intensity of the pixel I_{ij} is the input). However, other applications such as solving special types of partial differential equations are also potentially important. Even the two input possibilities can be combined, e.g.for processing successive images of a motion picture. Using the A^{τ}, B^{τ} delay-type templates some forms of successive (moving) images can be generated within the CNN.

In picture processing applications the gray-scale image pixel values are coded between +1 and -1, which correspond to black and white extremes.

Let us derive next an estimate for the dynamic range.

Proposition 1

For a CNN characterized by bounded nonlinear cloning templates (but without delay) and a memoryless piecewise-linear output equation (Eq.(2b)) in the canonical equations (2), all states v_{xij} are bounded for all time $t > 0$ and the bound vmax can be computed by the following formula:

$$v_{max} = 1 + R_x|I| + R_x K \max\Big[\sum_{C_{kl}\in N_r(ij)} (\max|\hat{A}_{ij;kl}| + \max|\hat{B}_{ij;kl}|)\Big] \quad (3)$$

$$1\le i\le M,\ 1\le j\le N$$

The proof follows the same way as in Section III of [1] except that the bounds $|A_{ij;kl}|$ and $|B_{ij;kl}|$ are replaced by $\max|\hat{A}_{ij;kl}|$ and $\max|\hat{B}_{ij;kl}|$), and the bound for $v_{yij}(t)$ is equal to K (instead of 1).

III. SOME IMPORTANT SPECIAL CASES

As we have mentioned in the introduction, a CNN with nonlinear and delay-type cloning templates as defined by the canonical equatious (2) can be considered as a unifying paradigm for analog nonlinear processor arrays where the processors are placed on a regular 3-dimensional geometrical grid. One layer of this array is placed on a two-dimensional regular grid (see Figure 1). We will now describe some important special cases:

Special Case 1: Nonlinear resistive grid (with capacitors)

Several recent analog processor arrays which mimic the vertebrate retina [5, 6] or some functional model of the visual process [8] make use of the circuit structure depicted in Figure 4. Obsrve that this structure is a simple special form of the CNN defined in Section II.

Namely, choose

$$I=0, \qquad A^{\tau} = B^{\tau} = 0$$

Choose $I_{xu} = I_b = B_{ij;ij}\, E_{ij}$

so that $\hat{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ in Fig.4(a) or $\hat{B} = \begin{bmatrix} 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 \end{bmatrix}$ in Fig.4(b)

Choose $v_{yij} = f(v_{xij})$ with $K\gg 1$

$\hat{A}_{ij;kl} = G(v)$ or 0 ; $G = G\,(v_{ykl}-v_{yij})$,

so that $\hat{A} = \begin{bmatrix} 0 & G & 0 \\ G & 0 & G \\ 0 & G & 0 \end{bmatrix}$ in Fig.4(a) or $\hat{A} = \begin{bmatrix} G & G \\ G & 0 & G \\ G & G \end{bmatrix}$ in Fig.4(b)

The nonlinear resistor v-i characteristics can be defined according to the various applications [5-9, 16]. The type of "fading" resistor characteristics of Figure 5 is used in several vision related applications having different names (e.g. resistive "fuse", etc.).

We wish to emphasize that the CNN defined here can operate in two modes; namely, either in transient mode or in dc steady state mode. In transient mode the output is the snapshot at a given time instant t=T (e.g. in the scale-spacing method associated with the anisotropic diffusion equation [8]). In the dc steady-state mode the output is the dc steady-state of the circuit. Equivalently, we can choose the speed of the input to be much smaller than the settling time of the circuit (e.g., speed is choosen equal to the reciprocal time constant in [6]).

Special Case 2: linear resistive grid without dynamics

Choose $K > 1, \quad C = 0, \quad A^{\tau} = B^{\tau} = 0$

Choose $\hat{A}_{ij;kl} = A_{ij;kl}\, v_{ykl}$ and $\hat{B} = 0$

or

$\hat{B}_{ij;kl} = B_{ij;kl}\, v_{ukl}$ and $\hat{A} = 0$

A Gabor transform can be calculated this way.

Similarly, many other interesting special cases can be generated (e.g. in case of motion estimation, etc.).

Remark 1

The above cases are special cases, indeed. The choice of many different resistive grids with other nonlinear characteristics like square, square-root, absolute value, etc.; can lead to many interesting results.

As an example consider the input picture of Figure 6. This picture is applied as the initial state ($v_{xij}(0)$) to a connected component detection CNN with the following linear templates on a square grid

$$\hat{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \hat{B} = 0 \qquad I = 0$$

The output $v_{yij}(\infty)$ shown in Figure 7(a) gives the number of connected components when Figure 6 is projected along the horizontal axis. Now, suppose we choose next

$$\hat{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \hat{B} = 0 \quad I = 0 \quad A^{\tau} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B^{\tau} = 0$$

$\tau = dh$ when $h=1$ is the time step unit and $d=3$. Hence $\tau=3$. The output shown in Figure 7(b) gives a combination of a connected component detector and a vertical line detector with a delay. This new CNN therefore gives a qualitatively new image processing effect.

All these calculations were performed by usig the CNND simulator program [15] running on an IBM PS/2 Model 60 computer.

IV. CNN AS ANALOG PROGRAMMABLE COMPUTING ARRAYS

We can compare the input and the specification of a cellular automaton, a systolic array, and a cellular neural network as follows:

	CA Cellular automaton	SA Systolic array	CNN Cellular neural network
Input	logic values (1-4 bits)	numerical values (8-32 bits)	analog values
Specification	truth table	numerical algorithm	cloning templates

The relations between a CA and a CNN with linear cloning templates has been studied in [13].

The computing efficiency in terms of finite area/time/dissipation varies from problem to problem. All three clases of arrays have their own advantages and disadvantages.

A crucial problem, however, is the programmability question. Both the CA and the SA can be programmed easily. The programmability of the CNN may not be obvious. We will show that due to its simple current-additive structure (even with nonlinear and delay-type templates), the CNN is in fact a programmable analog VLSI array computer.

The basic building blocks (atoms) of the CNN are the coefficients in the cloning templates, namely,

$$\hat{A}_{ij;kl}, \qquad \hat{B}_{ij;kl}, \qquad A^{\tau}{}_{ij;kl}, \qquad B^{\tau}{}_{ij;kl} \qquad I$$

These coefficients are realized by voltage-controlled current sources (VCCS). Note that I can be considered as a VCCS with a constant controlling voltage.

For example, in the cloning templates

$$\hat{A} = \begin{bmatrix} 0 & d_1 & 0 \\ d_1 & 2 & d_1 \\ 0 & d_1 & 0 \end{bmatrix} ; \qquad \hat{B} = 0 ; \quad I = 1.5 ; \quad A^{\tau} = B^{\tau} = 0$$

$$d_1 = 0.01(\exp(v_{ykl} - v_{yij}) - 1)$$

The basic building blocks are the nonlinear controlled sources d_1 and the linear controlled sources $A_{ii} = 2$ and $I = 1.5$.

These analog template atoms are really programmable since each additive constant term can be easily selected by switching on or off the relevant component from a bank of parallel connected linear and nonlinear controlled sources.

The cloning template is built up from the above template atoms, thereby forming an elementary processing function or an analog macro instruction.

The sequential or parallel applications of these analog macro instructions or cloning templates constitute the analog algorithm or analog software of the CNN. In fact, this notion of the software is in good agreement with both the basic physical meaning [16] and the recursive function analogy [17]. Hence, the following correspondence gives not only a clear structure of the hierarchy of both types of computation, but, also its engineering realization.

analog CNN	digital CA or SA
algorithm	
sequence of templates (series or parallel) -analog software	recursive function/algorithm /software
network	
cloning template	machine instruction
analog "macro"	macro
physical circuit	
cloning template atom	"gate"

Programmability: software - electrical - physical (VLSI mask)

Finally, we remark that in some recent practical applications various cloning template sequences are used succesfully for different tasks and even the analog and digital funtions can be used jointly in a dual computing structure [15, 17].

V. SOME QUALITATIVE PROPERTIES

The canonical equations (2) describe the dynamics of the CNN. In order to apply specific qualitative results from dynamical systems theory, it is convenient to reformulate these equations in a closed vector-matrix form. Without loss of generality, suppose, $R_x = 1$, $C = 1$ and $M = N$.

In the case of linear cloning templates we have the simple form

$$\dot{\underset{\sim}{v}}_x = -U\,\underset{\sim}{v}_x + A^L\underset{\sim}{v}_y + B^L\underset{\sim}{v}_u + I \qquad (4)$$

where $\underset{\sim}{v}_x(t)$, $\underset{\sim}{v}_y(t)$, $\underset{\sim}{v}_u(t) \in R^{N^2} \times R^1$ (i.e. N^2 vectors)

$\underset{\sim}{I} \in R^{N^2}$ with all elements equal to I

U is an $N^2 \times N^2$ unit matrix

$\underset{\sim}{v}_y = \underset{\sim}{f}(\underset{\sim}{v}_x)$ is a diagonal mapping with $v_{yi} = f(v_{xi})$

A^L, $B^L \in R^{N^2 \times N^2}$ (i.e. $N^2 \times N^2$ matrices)

We have N^2 cells, i.e. N^2 elements in the vectors. The question is how to order them. Figure 8 shows three possibilities ((a),(b) and (c)) for the case of a square grid with $N = 5$. Observe that many nearest neighbor cells will not be placed next to each other in this reordering. For example, for the three reorderings in Figures 8(a),(b) and (c), the nearest neighbors can be found in the vectors as far as the following index distances from each other:

$$N + 1, \quad 2N - 2 \quad \text{and} \quad 2N - 1 \qquad (5)$$

For example, we have 13-7=6 in Fig.8(a), 19-11=8 in Fig.8(b), and 15-6=9 in Fig.8(c).

For example, consider the linear cloning template

$$A = \begin{bmatrix} 0 & p_N & 0 \\ p_W & 2 & p_E \\ 0 & p_S & 0 \end{bmatrix} \quad ; \qquad B = 0; \qquad I = 0; \qquad (6)$$

and choose the diagonal ordering of Figure 8(b). The A^L matrix associated with this ordering is shown below where the entries in some selected rows and columns are given:

$$A^L = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ . \\ . \\ 9 \\ 10 \\ 11 \\ . \\ . \\ 19 \\ . \\ . \\ 25 \end{array} \begin{array}{c} \begin{array}{ccccccccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & \ldots\ldots & 9 & 10 & 11 & \ldots\ldots & 19 & \ldots\ldots & 25 \end{array} \\ \begin{bmatrix} 2 & p_E & p_S & & & & & & & & & & & \\ p_W & 2 & & p_S & p_E & & & & & & & & & \\ p_N & & 2 & p_S & p_E & & & & & & & & & \\ & p_N & 2 & & & & & p_E & p_S & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & p_W & p_N & & & 2 & & & & & & \\ & & & p_N & & & & & 2 & p_S & & & & \\ & & & & & & & & p_N & 2 & & p_E & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & p_W & & 2 & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & 2 \end{bmatrix} \end{array} \qquad (7)$$

The selected rows are complete. Observe that if A is centrally symmetric then A^L is symmetric, too.

For the general case of a CNN with nonlinear and delay-type templates, and using the same variables and vectors as above, the canonical matrix-vector description of the dynamics is as follows (aqain assuming a square grid with $r=1$, $R_x=1$, $C=1$).

$$\dot{\underset{\sim}{v}}_x = -U\, \underset{\sim}{v}_x + \hat{A}(\underset{\sim}{v}_y)\underset{\sim}{e} + \hat{B}(\underset{\sim}{v}_u)\underset{\sim}{e} + A^{\tau}\underset{\sim}{v}_y(t-\tau) + B^{\tau}\underset{\sim}{v}_u(t-\tau) + \underset{\sim}{I} \qquad (8)$$

where $\underset{\sim}{e} = [1, 1, 1, \ldots\ldots\ldots, 1]^T \in R^N$

$\hat{A}$, $\hat{B}$ are N^2xN^2 matrices with elements $\hat{A}_{ij;kl}(v_{ykl}, v_{yij})$, $\hat{B}_{ij;kl}(v_{ukl}, v_{uij})$

A^τ, B^τ are N^2xN^2 matrices with elements $A^\tau_{ij;kl}$, $B^\tau_{ij;kl}$

The ordering of the cell variables into the vectors v_x, v_y and v_u, is choosen in the same way as above for the linear templates.

For example, if we have a nonlinear cloning template

$$\hat{A} = \begin{bmatrix} 0 & p_N & 0 \\ p_W & 2v_{yii} & p_E \\ 0 & p_S & 0 \end{bmatrix} \quad ; \quad B = 0 \; ; \quad I=0; \qquad (9)$$

then the first 2 rows of the matrix A in Eq.(8) is as follows:

$$A = \begin{array}{c} \begin{array}{cccccccc} 1 & & 2 & & 3 & 4 & 5 & 6 & \ldots \end{array} \\ \begin{bmatrix} 2v_{yii} & p_E(v_{y2}, v_{y1}) & p_S(v_{y3}, v_{y1}) & 0 & 0 & 0 & \ldots \\ p_W(v_{y1}, v_{y2}) & 2v_{y2} & 0 & 0 & p_S(v_{y5}, v_{y2}) & p_E(v_{y6}, v_{y2}) & \ldots \end{bmatrix} \end{array}$$

Writing equation (8) in scalar form with $\tau = 0$, $A^\tau = B^\tau = 0$ we have

$$\dot{v}_{xi} = -v_{xi} + I + \hat{A}_{i1}(v_{y1}, v_{yi}) + \ldots + \hat{A}_{iN^2}(v_{yN}, v_{yi}) + \\ + \hat{B}_{i1}(v_{u1}, v_{ui}) + \ldots + \hat{B}_{iN^2}(v_{uN}, v_{ui}); \quad i=1, \ldots N^2 \qquad (12)$$

Based on the structure of the above canonical equations, several important stability results can be proved using similar techniques as in [1, 2, 12]. Here, as a representative result, we will prove only our next Proposition.

Proposition 2

If a CNN has a nonlinear cell-linking template [2] which has template atoms with strictly monotone increasing I-V characteristics in the non-local variable (i.e. for $p(v_{ykl}, v_{yij})$ $\partial p/\partial v_{ykl} > 0$) then the CNN is completely stable except possibly from a set of initial conditions of measure zero. A circuit is said to be completely stable iff every trajectory tends to an equilibrium state. Consequently, such a circuit can not oscillate or become chaotic (it is called convergent in [12]).

Proof. In view of the cell-linking property the dynamics described by (12) is irreducible. Moreover, the hypothesis of Proposition 2 guarantees that all the off-diagonal terms of associated Jacobian matrixare positive. Therefore, using [5,18] the dynamics is completely stable (convergent) except for a starting set with measure zero.

VI. CONLUSIONS

The cellular neural network with nonlinear and delay-type cloning templates form a unifying paradigm for analog (nonlinear) processing arrays. It provides not only a rigorous theoretical framework but, at the same time, many practical solutions to various image processing problems can be solved by these arrays in real-time. In particular, the first two phases of many image processing problems (e.g. noise removal and feature extraction of gray-scale pictures) belong to these classes of problems. In addition to many new applications, the nonlinear cloning templates allows us to model some biological properties of the retina but they can also be used for modeling motion dynamics. The delay-type templates provides us with even more flexibilities and new applications, including the detection of some motion features.

ACKNOWLEDGEMENTS

This work has been supported by the Office of Naval Research grant N00014-89-5-1402, the National Science Foundation grant MIP-8614000 and by the National Research Fund in Hungary (OTKA) grant 1212/88.

REFERENCES

[1]a L.O.Chua and L.Yang, "Cellular neural networks:Theory", IEEE Transactions on Circuits and Systems, Vol.35, pp.1257-1272, 1988.

b L.O.Chua and L.Yang, "Cellular neural networks: Applications", ibid., pp.1273-1290.

[2] L.O.Chua and T.Roska, "Stability of a class of nonreciprocal cellular neural networks", ibid., Vol.CAS-37, December, 1990.

[3] K.R.Krieg, L.O.Chua and L.Yang, "Analog signal processing using Cellular Neural Networks", Proc. IEEE ISCAS-90, May 1990.

[4] L.Yang, L.O.Chua and K.R.Krieg, "VLSI Implementation of Cellular Neural Networks", Proc. IEEE ISCAS-90, pp.2425-27, May 1990.

[5] C.Mead and M.Ismail (eds.), Analog VLSI implementation of neural systems, Kluwer Academic Publ., Boston, 1989.

[6] Chapter 2 in [5].

[7] C.Mead, Analog VLSI and neural systems, Addison Wesley, Reading, MA, 1989.

[8] P.Perona and J.Malik, "Scale-space and edge detection using anisotropic diffusion", to appear in IEEE Trans. Pattern Rec.and Machine Intelligence, 1990.

[9] D.Anastassiou, "Error diffusion coding for A/D conversion", IEEE Transactions on Circuits and Systems, Vol.36, pp.1175-1186, 1989.

[10] W.Otterbach et.al.,"Cd Se-TFT-addressed LC-flat panels without additional storage capacitors", Proc. EURODISPLAY-87, pp.104-106, 1987.

[11] T.Roska, "Analog events and a dual computing structure using analog and digital circuits and operators", pp.225-238 in Discrete Event Systems: Models and Applications (eds. P.Varaiya and A.B.Kurzhanski), Springer-Verlag, Berlin, 1988.

[12] M.W.Hirsch, "Convergent activation dynamics in continuous time networks", Neural Networks, Vol.2, pp.331-349, 1989.

[13] L.O.Chua and B.Shi, "Exploiting cellular automata in the design of cellular neural networks for binary image processing", ERL memo UCB/ERL M89/130, University of California, Berkeley, Nov.15, 1989.

[14] G.Bártfay and T.Roska, "A new cellular neural network template and its application to skeletonization", Report, No.24/1989, Computer and Automation Institute, Hungarian Academy of Sciences (MTA SzTAKI), Budapest, 1989.

[15] CNND simulator, Cellular Neural Network embedded in a simple Dual computing structure, User,s guide Version 1.1, Report No.25/1989 (ed.T.Roska), Computer and Automation Institute, Hungarian Academy of Sciences (MTA SzTAKI), Budapest, 1989.

[16] Á.Csurgay, "Fundamental limits in large-scale circuit modelling, Proc. ECCTD-83, pp.454-457, VDE-Verlag, Berlin, 1983.

[17] T.Roska, "A dual way of computing - learning from cerebral asymmetry", Proc. Symposium on Computer and Cognitive Sciences, pp.101-108, Tampere, 1989.

[18] M.W.Hirsch, "Systems of differential equations that are competitive or cooperative II.: Convergence almost everywhere", SIAM J.Math.Anal., Vol.16, pp.423-439, 1985.

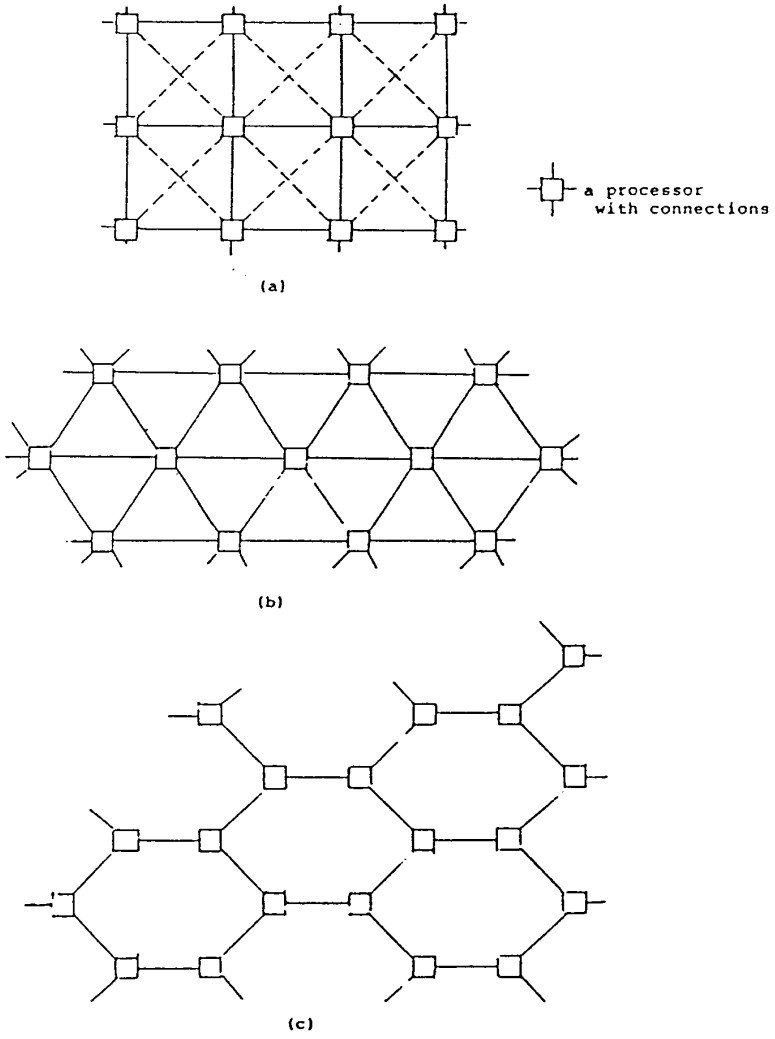
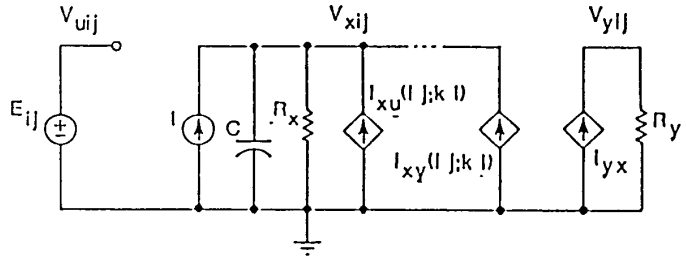


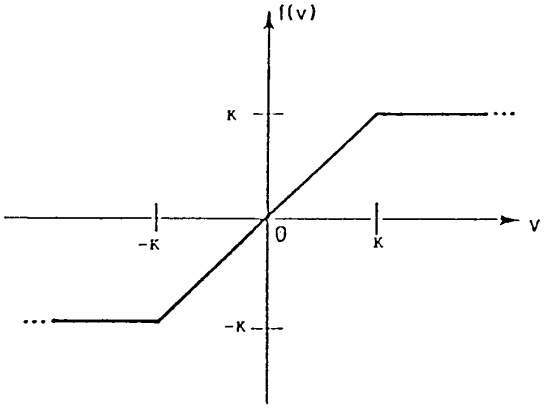
Figure 1. Typical regular grids : (a) rectangular (4 or 8 neighbors), (b) triangular (6 neighbors), (c) hexagonal (3 neighbors)



$$I_{xu}(ij;kl)=\hat{B}_{ij;kl}(v_{ukl},v_{uij}) \qquad I_{xy}(ij;kl)=\hat{A}_{ij;kl}(v_{ykl},v_{yij})$$

$$I_{yx}=\frac{1}{2R_y}\left[|1v_{xij}+K|-|v_{xij}-K|\right]$$

Figure 2. The analog processor cell circuit.



$$f(v) = \frac{1}{2}\left[\,|v+K| - |v-K|\,\right]$$

Figure 3. The nonlinear output function.

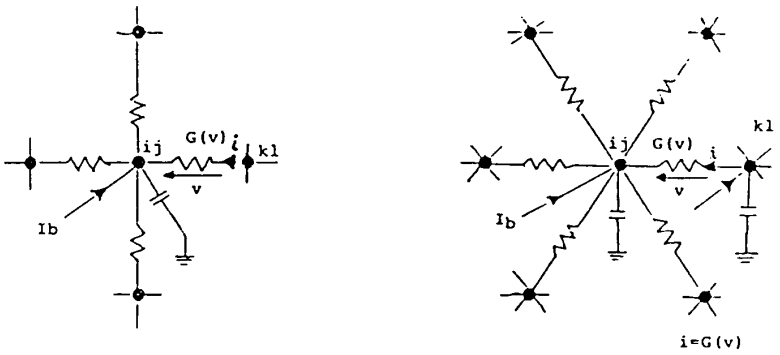


Figure 4. A resistive grid-type circuit structure. All resistors are nonlinear and characterized by i=G(v).

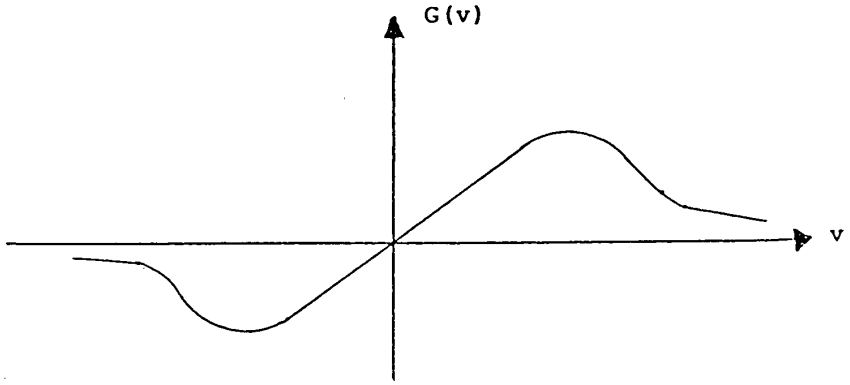


Figure 5. The v-i characteristic of a "fading" resistor.

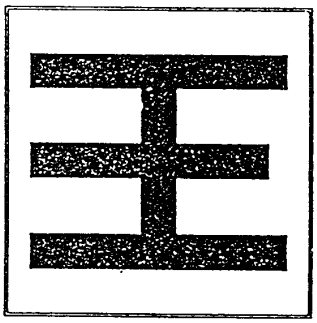
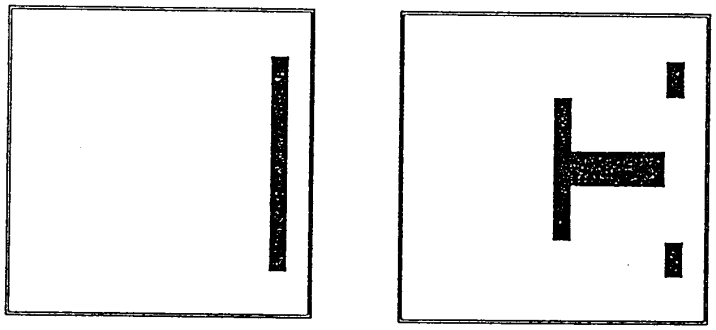
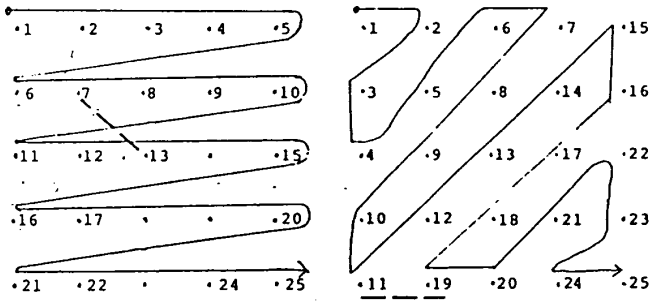


Figure 6. An input picture.

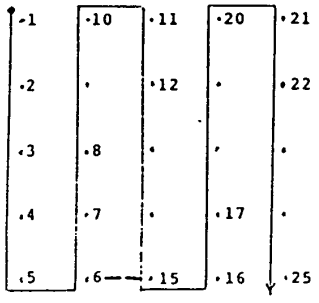


(a) (b)

Figure 7. The output pictures using (a) a linear cloning template and (b) a delay-type cloning template.



(a) (b)



(c)

Figure 8. The distinct ordering schemes.