J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

CELLULAR ARRAYS

History

This section gives a brief overview on cellular array architectures and related computing principles. Cellular arrays are massively parallel computing structures composed of cells placed on a regular grid. These cells interact locally, and the array can have both local and global dynamics. Indeed, it can be shown that in approximating any partial differential equation (*PDE*), continuous in space and time, one can use a cellular array as a computing environment, either in software simulation or in analog or digital hardware emulation. Two cellular array structures will be discussed, in detail: (i) cellular automata (*CA*)—discrete in space, time, and value, and (ii) cellular nonlinear or neural networks (*CNN*)—discrete in space and continuous in time and value. It will be shown that there is a biological relevance and motivation behind promoting cellular architectures as good prototypes for parallel computing. It is discussed how a stored-program analog array computer can be built, the CNN Universal Machine (*CNN-UM*), which makes it possible to synthesize novel spatiotemporal algorithms. Various physical implementations will be discussed, with emphasis on integrated array sensing and reconfigurable computing, that is, visual microprocessor architectures embedding stored programmability. Some application potentials will also be highlighted in engineering, biology, chemistry, and physics.

Historically, research on cellular arrays was initiated by von Neumann (1) in the 1960s, who studied the homogeneous structure of *cellular automata* and its evolution as a general framework for modeling complex structures. Such a structure consists of a set of cells, each one capable of performing only a set of primitive operations. Depending on the interconnection pattern among the cells and the initial content, the structure evolves through a sequence of states. With the introduction of CA, von Neumann's primary interest was to derive a computationally universal cellular space with *self-reproducing* and *self-repair* configurations. After a thorough study, he framed a "universal" CA with a set of transition rules involving five-neighborhood cells (a cross-shaped neighborhood of a cell), each having 29 distinct states. The transition function with this relatively small set of states turned out to be sufficient to achieve a powerful set of elementary operations: local logical operations, wire-branching, transmission, relative delays, construction, and destruction. Based on these operations he successfully synthesized various organs (pulser, decoder, coded channel, etc.) and higher-level functions (e.g., sensing); furthermore, he proposed a universal computer. Following this pioneering work, several researchers attempted to improve the construction of von Neumann's cellular space (23–4), and others started analyzing CA with the tools of abstract algebra, topology, and measure theory (56–7) with the ambitious goal of developing a unified theory of cellular models (89–10). In this first phase of CA research already a number of applications were proposed, including parallel language recognition, image processing, and biological modeling.

The new phase of activities in CA research is due to Wolfram (11, 12) in the early 1980s, who pioneered the investigation of CA as mathematical models for self-organizing statistical systems. He focused on a simplified structure: a discrete lattice of cells in which each cell has only two possible values (binary cells). The next state of a cell depends on itself and its two neighbors (three-neighborhood dependency). The cells evolve in discrete time steps according to some deterministic rule depending only on local neighbors. Thus, he defined a network that can be built based on a simple storage elements and combinatorial logic. Using polynomial and matrix algebraic tools, he moved toward in characterizing the state-transition behavior and the global properties of CA. Based on the statistical properties Wolfram classified the three-neighborhood CA broadly into four major categories (12). Packard and Wolfram (13) also defined the framework of the two-dimensional CA and characterized its basic properties. At the same time, others started to characterize CA models in various unrelated fields. Among other CA-based models for physical systems [e.g., simple crystal growth (14)], probabilistic analysis of CA behavior [e.g., the *Game of Life* algorithm (15)] and the architectural research of CA machines [e.g., *CAMs* (16)] should be mentioned. With the advent of very large scale integrated (*VLSI*) technology, a number of specific-purpose chips were built mainly for test-pattern generation and various cryptographic applications.

Parallel to this second phase of CA investigations another research field (17) also drew the attention of the scientific community. After almost two decades of relative silence, Hopfield's results (18, 19) on global optimization problems and biological modeling renewed the activity in the *neural network* (*NN*) and *artificial intelligence* (*AI*) research. In the next few years various models and architectures were proposed for a broad class of classification, optimization, and approximation problems (e.g., Refs 2021–22). However, because all these structures were densely or fully connected, their implementation posed a major challenge for VLSI designers and the wiring problems could not be adequately solved.

In the late 1980s a new cellular array architecture and computing principle was proposed by Chua and Yang (23), called *cellular neural/nonlinear networks* (*CNN*), standing at the crossroads of CA and NN research. CNN inherited the analog (autonomous-nonautonomous) network dynamics from NN and the cellular (locally connected) structure from CA, and soon it proved to be a powerful paradigm fertilizing research activity in a number of disciplines. It turned out to be a well-constructed framework for biological modeling especially modeling vision (242526272829–30), simulating or emulating partial differential equations (*PDEs*) (313233–34), and in various engineering designs (3536373839404142434445464748495051525354–55).

CNNs are regular, single-layer or multilayer, parallel processing structures with analog nonlinear computing units (base cells). The state value of the individual processors is continuous in time and their connectivity is local in space. The program of these networks is completely determined by the pattern of the local interactions, the so-called template. The time evolution of the analog transient, "driven" by the template operator and the processor dynamics, represents the computation in CNN (results can be defined both in equilibrium or nonequilibrium states of the network). The standard CNN equation (23) contains only first-order cells placed on a regular grid and the interconnection pattern is linear. This was soon generalized to higher-order cells, nonuniform grids, and nonlinear and/or delay-type templates (56, 57).

Completing the base cells of the CNN with local sensors, local data memories, arithmetical and logical units, furthermore, with global program memories and control units results in the CNN Universal Machine (*CNN-UM*) architecture [Roska and Chua (58)]. The CNN-UM is an analogic (analog and logic) supercomputer; it is universal both in the Turing sense (59) and also as a nonlinear operator (25). Therefore, when designing CNN processors, it can be used as a general architectural framework. Currently, there exist different physical implementations of this architecture: mixed-signal VLSI (6061626364656667–68), emulated digital VLSI (69), and optical implementation (70).

From hardware design point of view there is another important group of cellular arrays: *smart sensors*, among them probably the most important the *specific-purpose vision chips*, which also appeared in the 1980s after the pioneering work of Mead (71) and his research group. This line of investigation follows a more bottom-up approach: cells are constructed in analog VLSI and are connected to perform a certain sensing-computing task. Though the structures are CNN-like, there is no well-defined system level theory describing the qualitative properties of these architectures. The emphasis is on sensing combined with a specific type of computing for data prefiltering, reduction, and estimation of motion, of color-, and of form etc., rather than on solving a complex task on a reprogrammable cellular architecture.

This article gives a short overview of various cellular array architectures. First the mathematical description of different coupled cellular structures is presented by putting the emphasis on CA or CNN frameworks and their relation to PDE-based descriptions. Then various architectures and their biological relevance are discussed, followed by physical implementations and some state-of-the-art engineering designs. General application areas are also briefly described. Finally, the key concepts are summarized and future trends in cellular array research and implementations are predicted.

Foundations and Mathematical Descriptions

In this section the mathematical description of CA- and CNN-type cellular arrays is given along with the basic types of PDEs. It has been proven that all PDEs can be approximated to any desired accuracy by introducing finite differences (72) (and possibly discrete variables), that is, they can always be mapped to a cellular structure. Whereas all resulting models are locally connected, they may exhibit a very complicated global dynamics. Cellular arrays are computationally universal (7, 15, 59) and from an engineering point of view, they can be considered as dedicated architectures for building a universal parallel computer. The presentation of this section aims to underline the engineering view that any "PDE machine" that can be built is likely to be a cellular array.

PDE Machines as Cellular Arrays: an Engineering Approach. In a continuous space-time approach diffusive and wavelike transport mechanisms in physics can be well characterized by PDEs. A detailed qualitative and quantitative analysis of these processes usually focuses on space-time evolution of these systems; however, a closed-form solution of the describing equations is available only in fairly simple cases. Then, one may consider on constructing a universal computer programmed by PDEs and capable of calculating all the solutions. The major obstacle in building the ideal PDE machine—continuous in both space and time—is well explained by quantum mechanics stating that "the spectrum of the volume of any physical region is discrete" (73). For instance, even when choosing some gas or fluid as a basic material, there is always a certain microscopic scale at which these substances should be regarded as spatially discrete. On the other hand, treating them on a macroscopic scale (thus continuous in space), another engineering problem arises: adjusting some of the (global) physical parameters of this system would mean a very restricted way of programming. As a consequence, it is very likely that this "computer" would be self-describing, capable of calculating and predicting only those processes that it actually represents. Are there any better mathematical abstractions for parallel computing machines in engineering design?

Discretization along all spatial coordinates maps the PDE-based system into a dynamical system described by ordinary differential equations (*ODE*). Stated with other words, the PDE is approximated on a fixed grid looking at its properties at a chosen scale. This leads to a cellular array structure in which each node represents an *analog* (continuous state and time) computing device. In this case, it is easier to imagine the corresponding universal computer composed of cells having a prescribed sphere of influence in a local neighborhood, similar to the sensory organs and the nervous system of different species in nature. Here, reprogramming would mean changing the cell types and/or their local interaction (in order to change the local configuration) to realize a universal computation.

Further simplification is possible by discretizing the ODE system in time and limiting the cell states to a finite number of states. This "fully discretized" system, described by ordinary difference-differential equations (*ODDE*), is theoretically still capable of reproducing the qualitative properties of the underlying PDE while possessing structural simplicity that makes it tailor-made for physical implementation. For instance, imagine a structure of molecules in a chemical substance in which the spin of the electrons is described by discrete states and the chemical reaction evolves in phases. In this cellular array structure each node represents a *digital* (discrete state and time) computing device. Similarly to the analog case, reprogramming would ensure the construction of possible local configurations and the realization of a universal computation.

The above-described analog and digital cellular arrays are not only good approximations of various PDE-based dynamical systems, but they can also be considered as universal computers (7, 15, 59). Furthermore, it has been proven that there are cases in which for a certain cellular array structure (described by ODEs or local rules) the limiting PDE representation with the same qualitative properties does not even exist (74). This promotes a provocative view that PDEs are merely idealizations of (or only alternatives to) cellular structures described by coupled ODEs or local update rules (75, 76).

Systems Described by Partial Differential Equations. In the following, some basic types of PDEs, which describe continuous space-time systems, will be introduced. Without loss of generality the mathematical descriptions will be specified in a two-dimensional (*2-D*) setting. Let us consider a planar signal flow (a spatiotemporal intensity function) $I(x,y,t)$, where (x,y) represent the spatial coordinates and t the time dependence. Then the nonlinear reaction-diffusion equation is represented by

$$\frac{\partial I(x,y,t)}{\partial t} = D \operatorname{div} \operatorname{grad}\{g(I(x,y,t))\} + F(I(x,y,t),L) \qquad (1)$$

where g and F stand for nonlinear functions, D and L are scalars. As a special case, assuming no changes in time (and setting $F = 0$), one obtains the following Laplace equation

$$0 = D \operatorname{div} \operatorname{grad}\{g(I(x,y,t))\} \qquad (2)$$

Using the same notation, a simple wave-type equation can be described as

$$\frac{\partial^2 I(x,y,t)}{\partial t^2} = D \operatorname{div} \operatorname{grad}\{g(I(x,y,t))\} \qquad (3)$$

Other types of useful PDEs, developed particularly for 2-D signal (image) processing, can be found in 777879–80.

Equations (1, 2, 3) and their variants are often used to describe diffusion and wave-type transport mechanisms in physics. There is an important common thread in these descriptions: there are no distant interactions in space-time; from the continuity assumption it follows that only “infinitely close” particles interact. This makes it impossible to build a device that exactly corresponds to the above-noted PDEs; however, it provides the chance to approximate these equations by simple locally connected cellular systems.

Systems Described by Ordinary Differential Equations: Cellular Neural or Nonlinear Network. Cellular neural or nonlinear networks (CNN) are two- or higher-dimensional arrays defined by simple dynamical systems arranged at the node points of a regular grid (Fig. 1). The state of these systems (cells) is represented by a real number, the cells interact locally, and their operation is continuous in time. The cell dynamics is described by the following nonlinear ordinary differential equation with delayed-nonlinear intercell

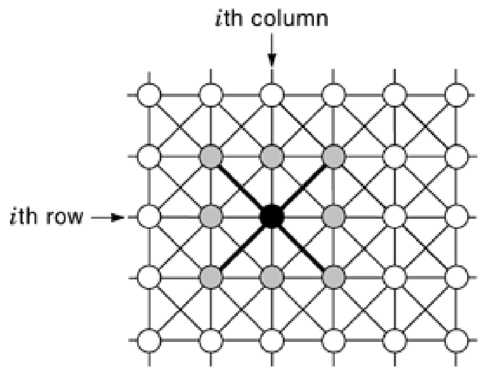

Fig. 1. A two-dimensional cellular array defined on a square grid. The *ij*th cell of the array is shaded black; cells that fall within the sphere of influence of neighborhood radius $r = 1$ (the nearest neighbors) are gray.

coupling (the extension to higher dimensions is straightforward, allowing similar interlayer interactions):

$$\begin{aligned} C\frac{d}{dt}x_{ij}(t) &= -R^{-1}x_{ij}(t) + \sum_{kl\in N_r} \hat{A}_{ij,kl}(y_{kl}(t-\tau_A), y_{kl}(t-\tau_A)) \\ &\quad + \sum_{kl\in N_r} \hat{B}_{ij,kl}(u_{kl}(t-\tau_B), u_{ij}(t-\tau_B)) + z_{ij} \\ y_{ij}(t) &= -C_y\frac{d}{dt}x_{ij} + f(x_{ij}(t)) \end{aligned} \tag{4}$$

where $x_{ij,}$ $u_{ij,}$ and y_{ij} are the state, input, and output voltages of the specified CNN cell, respectively. The notation ij refers to a grid point associated with a cell on the 2-D $M \times N$ grid, and $kl{\in}N_r$ is a grid point in the neighborhood within the radius r of the cell ij. $\hat{A}_{ij,kl}$ represents the delayed nonlinear feedback, $\hat{B}_{ij,kl}$ the delayed nonlinear control, and τ_A and τ_B represent finite time delays, respectively. The constant z_{ij} is the cell current, which could also be interpreted as a space-varying threshold. In general, the CNN template, which is the "program" of the CNN array, consists of the $[A\ B\ z]$ terms (omitting the indices). The output characteristic f is a sigmoid-type (e.g., piecewise linear) function. Note that the output equation also describes a first-order dynamical system. The time constant of a (first-order) CNN cell is determined by the linear capacitor (C) and the linear resistor (R), and it can be expressed as $\tau = RC$. Without loss of generality $R = 1$ and $C = 1$ will be considered.

Simplifying the interaction types to nondelayed linear operators ($\tau_A = 0$ and $\tau_B = 0$), omitting the dynamics in the output equation ($C_y = 0$), and ceasing the space-variant nature of the cell current ($z_{ij} \to z$) result in the "standard-CNN" system [(23), the circuit theoretic model can be seen in Fig. 2]:

$$\begin{aligned} \frac{d}{dt}x_{ij}(t) &= -x_{ij}(t) + \sum_{kl\in N_r} A_{ij,kl}y_{kl}(t) + \sum_{kl\in N_r} B_{ij,kl}u_{kl}(t) + z \\ y_{ij}(t) &= f(x_{ij}(t)) \end{aligned} \tag{5}$$

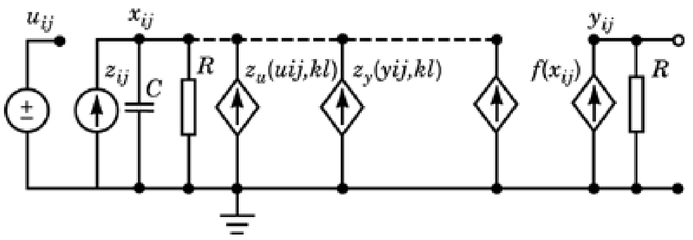

Fig. 2. The circuit theoretic model of a standard CNN cell with voltage-controlled current sources.

Qualitative properties. Major efforts in CNN research have been concentrated on studying 2-D rectangular standard cellular systems described by Eq. (5). Depending on the template values, the associated CNN might exhibit a very complicated dynamics: stable, oscillatory, and chaotic patterns can also be generated. In 2-D signal (image) processing applications, usually the globally or completely stable CNN behavior is exploited by creating the analog kernels of a complex algorithm. A comprehensive overview on stable CNNs can be found in Ref. 81.

PDE versus CNN. The cellular array described by Eq. (4) can be used to approximate Eqs. (1)(2) (3) to any required precision by introducing finite differences in space. As an example, let us derive the CNN equation corresponding to a variant of Eq. (1), the linear heat equation [setting $F = 0$, $L = 0$ and $g(I) = I$], by using a four-neighbor discretization of the Laplacian:

$$\begin{aligned} \frac{d}{dt} I(x,y,t) &= D\left(\frac{\partial^2 I(x,y,t)}{\partial x^2} + \frac{\partial^2 I(x,y,t)}{\partial y^2}\right) \\ &\approx D\frac{1}{4}\left[I_{ij-1}(t) + I_{ij+1}(t) + I_{i-1j}(t) + I_{i+1j}(t)\right] - DI_{ij}(t) \quad (6) \end{aligned}$$

Thus the linear heat (diffusion) equation can be directly mapped onto the CNN array resulting in the following simple template (choosing $D = 1$):

$$A = \begin{bmatrix} 0 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix}, \qquad B = 0, \qquad z = 0 \qquad (7)$$

Programmed by this template and operating in the central linear region of the output characteristic, the CNN solves the associated heat equation. Indeed, this "PDE machine" based on Eq. (5) has already been built as an analog VLSI circuitry (e.g., Ref. 67). In general, Eq. (4) can be viewed as the mathematical framework for *analog parallel processing cellular arrays*, discrete in space but continuous in time.

Systems Described by Local Rules: Cellular Automata. A CA consists of a regular lattice of cells (see Fig. 1). Each cell takes q possible values, and is updated in discrete time steps according to a local rule Φ that depends on the value of the neighboring cells. The value x_{ij} of a cell at position i,j in a 2-D CA with a rule that depends on the neighbors within the radius $k,l \in N_r$ evolves according to

$$x_{ij}(t_{m+1}) = \Phi(\forall\, x_{kl}(t_m) | kl \in N_r) \qquad (8)$$

There are several possible lattices and neighborhood structures for 2-D CAs. For instance, a five-neighbor square CA (a "standard CA") evolves as follows:

$$x_{ij}(t_{m+1}) = \Phi(x_{i,j}(t_m), x_{i-1,j}(t_m), x_{i+1,j}(t_m), x_{i,j-1}(t_m), x_{i,j+1}(t_m)) \qquad (9)$$

In most cases, x_{ij} takes binary values and Φ is a Boolean logic function of the neighboring cell values.

Qualitative properties. A major line of investigations in CA research deals with 1-D and 2-D standard cellular systems described by Eq. (9). A qualitative characterization orders these CAs into four classes (12): (i) evolution leads to a homogeneous state, (ii) evolution leads to a set of separated simple stable or periodic structures, (iii) evolution leads to a chaotic pattern, and (iv) evolution leads to complex localized structures (capable of "universal" computation).

PDE versus CA. The cellular array described by Eq. (8) can be used to approximate Eqs. 1, 2, 3 to any required precision by introducing finite differences in space-time and discretizing the state variable. Taking again the linear heat equation [derived from Eq. (1)] as an example, one obtains ($\Delta\, t = h$)

$$\begin{aligned} I_{ij}(t_{m+1}) &= (1 - hD)I_{ij}(t_m) \\ &+ h\frac{D}{4}\left[I_{i-1,j}(t_m) + I_{i+1,j}(t_m) + I_{i,j-1}(t_m) + I_{i,j+1}(t_m)\right] \qquad (10) \end{aligned}$$

Observe that in Eq. (10) the local rule Φ represents the linear combination of the neighboring cell values. The state value I_{ij} is quantized and in order to approximate properly the original PDE it should take multiple discrete values.

CNN versus CA. Equation (10) is a fairly general CA model; however, it can also be viewed as a simple discrete-time CNN model with no input (82). This also clarifies some important differences when comparing the CA and CNN frameworks. The CA represents autonomous discrete-valued lattice dynamical systems, while the CNN can be regarded as a general description of analog-valued nonautonomous cellular dynamical systems.

The "standard" framework of both the CA and CNN research is the mathematical description that closely describes the core of the first hardware implementations. Note that a simple algorithm running on a CNN hardware (67) can exactly emulate a binary CA with Boolean logic update rules (13).

Remarks. Any numerical integration formula solving (approximating) a PDE (and an associated CNN) in the "binary universe" can be regarded as a general CA model although with multiple discrete states and complicated local update rules. In general, Eq. (8) can be viewed as the mathematical framework for autonomous *digital parallel processing cellular arrays*, discrete in space-time. A general nonautonomous nonlinear model could be described by the discrete time version of Eq. (4), the discrete-time CNN equation [(82), *DTCNN*].

Architectures and Biological Relevance

Different CA and CNN architectures. Based on the preceding generic definition of the CNN [Eq. (4)] and CA [Eq. (8)], several types of cellular arrays can be generated. They can be classified according to the types of the *grid*, the *processor* (cell), the *interaction* (template or update rule), and the *mode of operation*. Thus, building the core of a massively parallel computer based on cellular array architecture, the following key points should be addressed in detail:

(1) *Grid types* Cellular arrays are usually defined on a spatially discrete *square* (*rectangular*) grid, however, *hexagonal* and *triangular* arrangements can also be considered (Fig. 3(a–c)). These grids are the only

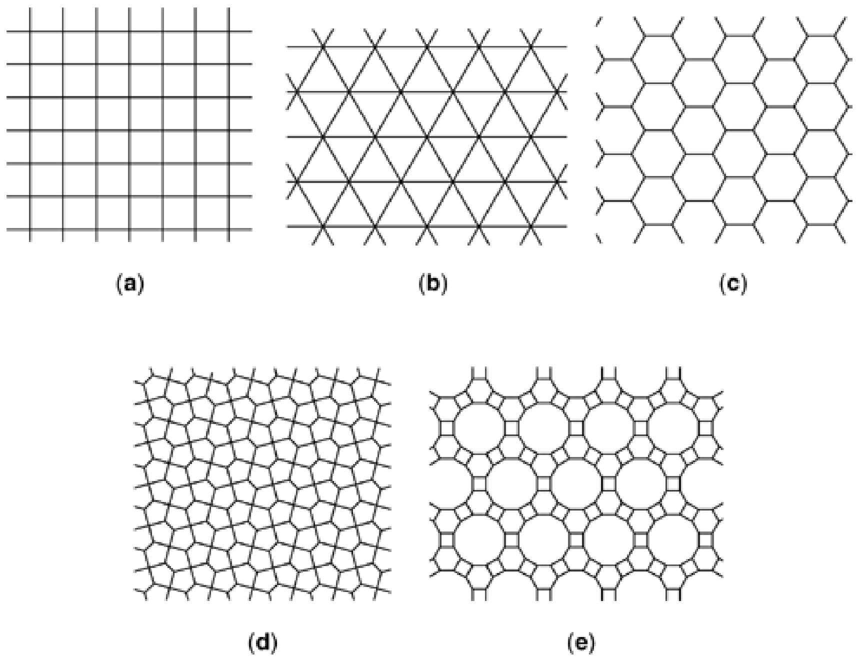

Fig. 3. The most common and two special grid types of cellular architectures. Regular contiguous tessellations of the plane based on congruent polygons are shown in the top row: (a) rectangular, (b), triangular, and (c) hexagonal lattices. In the bottom row two special grid types are given that belong to the ornamental group. All grid types (b)–(e) could be mapped to a rectangular grid (a) with periodic space-variant interconnections.

regular contiguous tessellations of the plane based on congruent polygons alone. Other grid types can also be created based on nonregular congruent polygons or from a regular vertex grid through discrete geometrical transformations: rotations and translations [ornamental groups, Fig. 3(d–e)]. A great number of these grids can be mapped on a typical eight-neighbor rectangular structure with periodic space-variant connections (83). Multiple and varying grid sizes (e.g., coarse and fine grids, and logarithmically changing size) may be useful in simulating adaptive biological systems (e.g., retina, lateral geniculate nucleus (LGN), or magno-parvo pathways in the cortex).

(2) *Cell (processor) types and boundary cell types* Cellular arrays can be built from linear or nonlinear, first- or higher-order cells. A linear or "small-signal" operation (as a special case) is achievable through piecewise linear output characteristics. A highly nonlinear dynamic behavior (typical CNN models) is attainable through a sigmoid, Gaussian, or inverse-Gaussian type output characteristics. Cells can include additional local analog and/or logical memories for storing intermediate processing results. A cell might perform local logic operations (typical CA models) or a combined analogic (analog and logic) computation (extended CNN models). The cells of an array can be uniform or nonuniform (regularly or slightly varying). In some processing tasks two or three cell types in a regular grid might be very useful, for example, in color image processing. Since in any physical implementation only a finite array can be built, a boundary condition should be exactly defined. Creating boundary cells that are not involved in the cooperative computation ("virtual cells") can satisfy this requirement. The most important boundary cell specifications are as follows: (i) In the *fixed* (*Dirichlet*) type, constant values are assigned to all boundary cells. (ii) In the *zero-flux* (*Neumann*) type, boundary cells are made to follow cells that are on the same side of the array. (iii) In the

periodic (*toroidal*) type, boundary cells are made to track the values of cells from the opposite side of the array.

(3) *Neighborhood size and interaction types* The interaction type (the biological "synapse") between the grid cells represents the program of a cellular array. Depending on whether these interaction types are fixed or programmable, the array can be regarded as a specific-purpose (see previous section on CA and smart-sensor implementations) or a reconfigurable (see previous section on the CNN-UM and its implementations) parallel array architecture. The nearest neighborhood is the most common sphere of influence in the intercell communication for both CA and CNN models (either the cross-shaped fourfold-connected or the star-shaped eightfold-connected neighborhoods); however, larger neighborhood sizes can also be considered (typically in biological modeling or when creating adaptive artificial systems). The interaction types can be linear or nonlinear memoryless functions [e.g., a typical CA update rule corresponding to Eq. (8)] of one, two, or more variables. Delayed nonlinear [e.g., a typical CNN template corresponding to Eq. (4)] and dynamic (lumped) interactions are more general connection types. By breaking the symmetry or isotropy and moreover varying the nature of the interconnections in space and/or in time, an extremely rich collection of further templates (or local update rules) can be created.

(4) *Modes of the operation* The mode of operation can be continuous (general CNN models) or discrete time (general DTCNN and CA models). In the case of a discrete-time procedure the update mechanism can be synchronous or asynchronous. The computation can be analog, logic, or analogic (see also item 2) and can be executed either in local mode or in propagating mode (on a decoupled or coupled array, respectively). Besides the usual fixed-point operational mode (equilibrium computing), transient (nonequilibrium computing), oscillating, chaotic, and general stochastic modes can also be used.

The CNN Universal Machine: An Analogic Stored Program Cellular Array Computer. A cellular architecture that includes the main properties listed and discussed in the previous subsection is the *CNN Universal Machine* [*CNN-UM* (58)] The CNN-UM makes it possible to combine efficiently analog array operations with local logic. Because the reprogramming time is approximately equal to the settling time of a nonpropagating analog operation, it is capable of executing complex analogic (analog and logic) algorithms. To ensure programmability, a global programming unit was added to the array and for an efficient reuse of intermediate results, each computing cell was extended by local memories. In addition to local storage, every cell might be equipped with local sensors and additional circuitry to perform cellwise analog and logical operations. The architecture of the CNN-UM is shown in Fig. 4.

The CNN-UM is built around the dynamic computing core of a simple CNN. An image can be acquired through the sensory input [e.g., optical sensor (*OPT*)]. Local memories store analog [local analog memory (*LAM*)] and logic [local logical memory (*LLM*)] values in each cell. A local analog output unit (*LAOU*) and a local logic unit (*LLU*) perform cellwise analog and logic operations on the stored values. The output is always transferred to one of the local memories. The local communication and control unit (*LCCU*) ensures the communication between the extended cell and the central programming unit of the machine, the global analogic programming unit (*GAPU*). The GAPU has four functional blocks. The analog program register (*APR*) stores the analog program instructions, the CNN templates. In the case of linear templates, for connectivity $r = 1$, a set of 19 real numbers has to be stored (this number is even less for both linear and nonlinear templates assuming spatial symmetry and isotropy). All other units within the GAPU are logic registers containing the control codes for operating the cell array. The local program register (*LPR*) contains control sequences for the individual cell's LLU, and the switch configuration register (*SCR*) stores the codes to initiate the different switch configurations when accessing different functional units (e.g., whether to run a linear or nonlinear template). The global analogic control unit (*GACU*) stores the instruction sequence of the main (analogic) program. The GACU also controls the timing, sequence of instructions and data transfers on the chip and synchronizes the communication with any external controlling device.

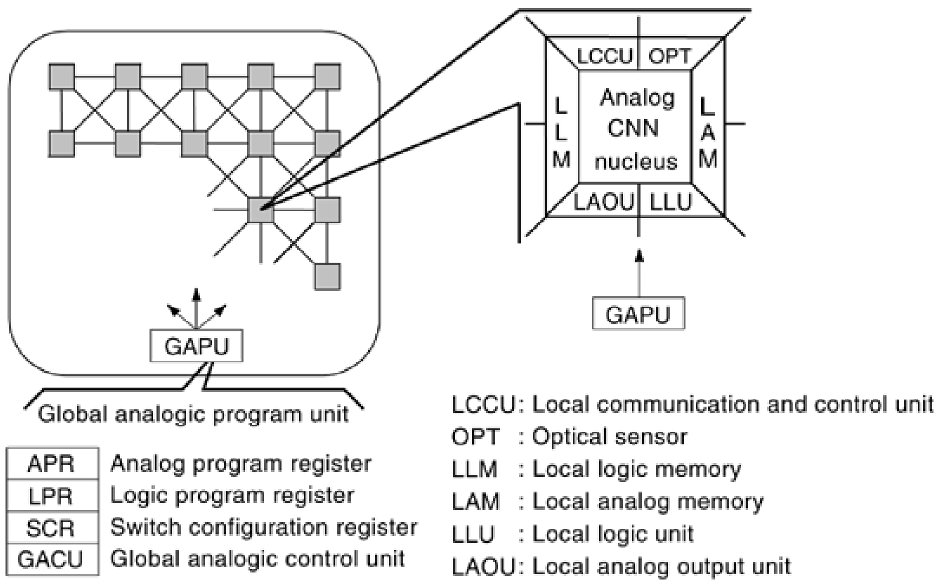
LCCU
OPT
L L M
Analog CNN nucleus
L A M
LAOU
LLU
GAPU
GAPU
Global analogic program unit
APR Analog program register
LPR Logic program register
SCR Switch configuration register
GACU Global analogic control unit
LCCU: Local communication and control unit
OPT : Optical sensor
LLM : Local logic memory
LAM : Local analog memory
LLU : Local logic unit
LAOU: Local analog output unit

Fig. 4. The architecture of the CNN Universal Machine, the stored program analogic array supercomputer.

Synthesizing an analogic algorithm running on the CNN-UM, the designer should decompose the solution to a sequence of analog and logical operations. A limited number of intermediate results can be locally stored and combined. Some of these outputs can be used as a bias map (space-variant bias) or fixed-state map (space-variant mask) in the next operation by adding spatial adaptivity to the algorithms without introducing complicated intercell couplings. Either a linear or a nonlinear template defines analog operations. The output can be defined both in fixed and nonfixed states of the network (equilibrium and nonequilibrium computing), depending on the control of the transient length. It can be assumed that elementary logical (NOT, AND, OR, etc.) and arithmetical (ADD, SUB) operations are implemented and can be used at the cell level between LLM and LAM locations, respectively. In addition, data transfer and conversion can be performed between LAMs and LLMs.

Biological Relevance. Earlier it was explained why cellular arrays can be viewed as efficient PDE machines in simulating physics. Here, some key observations made in biology—based on morphological, pharmacological, and physiological measurements—will be recalled that underline the biological relevance of cellular architectures and also the *analogic* operational mode of array computing.

At the *neuronal level*, it is instructive that many complex neural functions are performed without spikes, and in many visual functions (e.g., within the retina of the vertebrates) 2-D layers of locally connected neurons perform some tasks that combine both spikeless (analog) and spike-type (logic) processing as well (84). Cellular arrays are also natural models for *topographic maps* of biological sensory systems (84). At the *neural systems level*, recent physiological discoveries have shown that complex visual tasks are solved by using several different projections of the same image stored "retinotopically" in different parts of the brain (cortex) as a "multiscreen theater" (85). Finally, at the *cognitive systems level*, the evidence of functional cerebral hemispheric asymmetry of the brain (86) provides a strong motivation for formulating the cellular computing model as a nonlinear analogic (dual) computing structure (87) and justifies a novel approach to computational complexity (88).

Physical Implementation

In this section a brief overview on the main physical implementation trends of cellular arrays will be given. The emphasis will be put on those *reconfigurable cellular architectures that interact with the physical world*, that is, where on-chip sensing and flexible computing are integrated. For these engineering prototypes CA models cannot be considered as a general framework. The unifying paradigm is rather a CNN-type general model with locally connected analog "computing sensors" arranged on a regular rectangular grid. Specific purpose vision chips ("smart sensors") make up a very important class of these architectures, in which sensing is combined with a specific functionality. These chips are special-purpose nonprogrammable (only tunable) information sensing devices. Programmable single-chip visual microprocessors can be considered as a more general class developed within the CNN framework. The general architecture of these reconfigurable devices (CNN-UM) has been described in the previous section and lead to efficient optoelectronic prototypes. Throughout the section the phrase *visual sensing* should be understood in a very broad sense: it rather describes a general 2-D sampling of any physical property of the environment (e.g., color, temperature, pressure, smell) than only the 2-D representation of the light intensity within the visible spectrum.

CA implementations. The CA theory evolved in parallel with the advance of digital technology that implemented noncellular architectures and used sequential computing principles for decades. Though the theory of autonomous cellular automata machines (CAMs) was well developed in the 1980s, single-chip large-scale 2-D prototypes had not been fabricated. The CA framework was used in parallel supercomputer [e.g., the Connection Machine (89)] and massively parallel processor design [*MPP* (90)], furthermore in specific cryptographic and test hardware synthesis. However, this framework was not widely used in engineering designs integrating array sensing and computing.

Specific-Purpose vision Chips: Smart Sensors. Carver Mead, who first introduced the idea of *neuromorphic engineering* using VLSI technologies in the 1980s, pioneered the research field of smart sensors (71, 91, 92). The main idea was to integrate the photodetecting elements with the processing circuits on the same chip and perform sensor information processing without redundant and unnecessary data acquisition ("smart sensing"). The smart-sensor concept implies a *low-level interaction* between the sensors and processors; therefore, modular camera-processor combinations do not belong to this class. During the last decade the sensors that have been embedded into these devices were almost exclusively photodetecting elements; however, recently other types have also been designed in cellular array architectures (e.g., high-resolution tactile sensors).

Specific-purpose vision chips are optimized for certain functionality, for the quality of processing, instead of the acquired image quality aimed by high-resolution and precision cameras. The advantage of vision chips lies in their *very high processing speed*, *large* (usually adaptively controlled) *dynamic range*, *small size*, and *low power dissipation* compared to camera-processor combinations. At the current stage, it is a drawback that most of these devices are fully custom designed (time consuming, costly, and error prone) and that the implemented algorithms should account for the hardly controllable inaccuracies of analog VLSI systems. Furthermore, *none* of the vision chips are of general purpose, i.e., they could be parameter tunable within certain limits but are not programmable to perform different vision tasks. This property is particularly undesirable during the development and test phase of a vision system.

The main technologies used in vision chip fabrication are complementary metal-oxide semiconductor (*CMOS*), charge-coupled device (*CCD*), bimetal CMOS (*BiCMOS*), and GaAs (MES-FET and HEMT), though CMOS was exclusively used in the majority of the designs. Vision chips can be classified (93) into spatial (e.g. 94,95), spatiotemporal (e.g., 9697–98), and optical neurochip (e.g., Ref 99) groups.

Programmable Visual Microprocessors (CNNM-UM chips). Specific-purpose vision chips are non-programmable cellular architectures. Since they implement only a specific functionality they require a top level system for complex information processing. Similarly, all fully connected neural network chip realizations have a common feature: they implement a single instruction only; thus the weight matrix is fixed when processing

Table 1: Comparison of the Performance Characteristics of Different CNN-UM Chips

Place of design	Berkeley and Munich	Serville	Leuven	Seville	Berkley	Helsinki	Seville
Date of design	1993	1994	1995	1995	1996	1997	1998
Array size	12×12	32×32	20×20	20×22	16×16	48×48	64×64
Cell type	DTCNN	Full range	Chua-Yang	Full range	Chua-Yang	Chua-Yang	Full range
Technology	2 μm	1 μm	0.7 μm	0.8 μm	1 μm	0.5 μm	0.5 μm
Time constant (τ)	300 ns		4.8 μs	400 ns	27 ns	50 ns	250 ns
Input	Analog	Binary and optical	Analog	Binary and optical	Analog	Binary	Analog
Output	Binary	Binary	Analog	Binary	Analog and binary	Binary	Analog and binary
APR	External	8	External	8	External	1	32
LLU	AND	Programmable		Programmable	Programmable	Programmable	Programmable
LLM	2	4		4	2	2	4
LAM							4

some input. Reprogramming (i.e., changing the weight matrix) is possible for some devices but takes longer time (by orders of magnitudes) than the computation itself.

These observations motivated the design of the CNN-UM (58), *a stored program nonlinear array computer*. The CNN-UM can also be viewed as a *visual microprocessor* especially when it is built with a focal plane sensory array. Different analog VLSI implementations of CNN-UM chips and a comparison of their performance characteristics can be found in Table 1. The first fully working implementation that can run analogic algorithms is the 1995 mixed-signal version (it has an optical input) from Seville (62). This chip, embedded into the CNN prototyping system (100), was used in various experiments validating CNN templates and algorithms. The most promising is certainly the latest version of these implementations fabricated in 1998 (67) (see the last column). It has a 64×64 CNN array with optical input, and in addition to the features shown in Table 1 it allows the use of fixed-state map techniques, global logical lines, and ARAMs (analog random access memory) (66) during the algorithm synthesis. This prototype already paves the road toward industrial applications.

There is always a gap between the system level design and chip implementations. Trying to synthesize powerful algorithms that can keep up with the state-of-the art methods of signal-processing disciplines, the engineer at the system level always faces the problem of hardware limitations and has to simplify the methodologies used. On the other hand, a VLSI designer would like to know the priority of the requirements motivated by different applications. At the current stage, the need for higher-order (complex) cells, nonlinear interactions, larger neighborhood size, and space-variant programming contradicts the requirement of higher resolution. Designing the future generation of CNN-UM chips the cell area versus functionality trade-off will always be one of the main issues with which to cope.

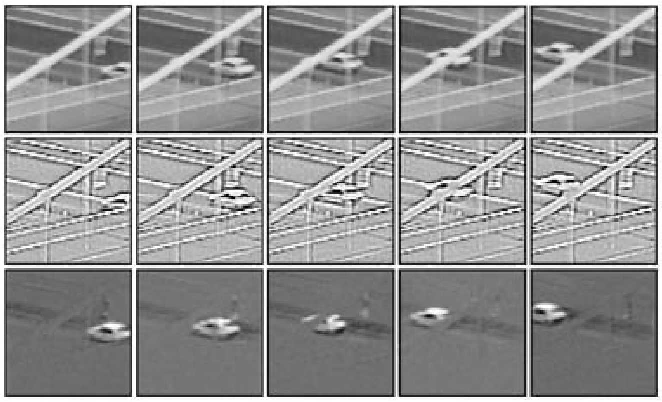

Fig. 5. Spatial edge (middle row) and spatiotemporal motion (bottom row) detection output from a "retinotopic" computation executed on a CNN-UM chip (input is shown in the top row). Simply reprogramming the array can test different biological retina models.

Applications

In this section some applications based on cellular array architectures will be highlighted. Since these areas are very diverse the objective of this summary is only to show how various disciplines can profit from having a programmable (reconfigurable) "visual" microprocessor equipped with both analog and logical computing capabilities. It is stressed that the algorithmic flexibility connecting biology, physics, and chemistry on these cellular architectures also fertilizes the engineering designs targeting industrial and other real-life applications. Experimental results based on existing prototypes (63, 67) will also be demonstrated.

Biological Modeling: Revealing Adaptation and Plasticity. Cellular architectures have been extensively used in various biological modeling experiments (e.g., 2425262728–29) but probably *retina modeling* is the only interdisciplinary research field that has significantly influenced the engineering designs. For instance, most specific-purpose vision chips aim to reproduce some well-defined functionality (e.g., spatial edge detection or spatiotemporal motion detection) of this "front end" of biological visual systems.

The *retina* is said to be the "*approachable part of the brain*" (101) and indeed, currently a large number of morphological observations, pharmacological, and physiological measurements are at hand helping us to understand the cells, synapses, and circuitry of this tiny neural processing layer. It is certainly due to its enormous complexity and emerging computational capability that even nowadays the neural code sent to the cortex is not fully understood. Vision chip design exploits the information learned about the structure and some functionality of the retina, but usually there are radical differences comparing biological and silicon retinas at the cell and synaptic levels. Also, the observed measured functional richness, the adaptation and plasticity properties in the retina assume, in engineering terms, a very flexible re-programmability that is not incorporated into these smart sensor architectures.

Theoretical research and implementation in the CNN field have evolved into a closer interaction with retinal modeling experiments (25). Due to this fact the latest CNNU-UM chip (67) is capable of computing both the spatial edge and spatiotemporal motion features of an image sequence (Fig. 5). Other functionality can also be easily tested, i.e., simply by reprogramming this architecture, one can generate hypotheses on various retinal models.

Retina modeling also motivated a new direction in CNN chip designs trading resolution for so-called complex cells (higher-order cells with more complicated synapses) that can also be interpreted as having multilayer CNN structures with simple cells (102). Just taking a glance at a simplified outer-retina model

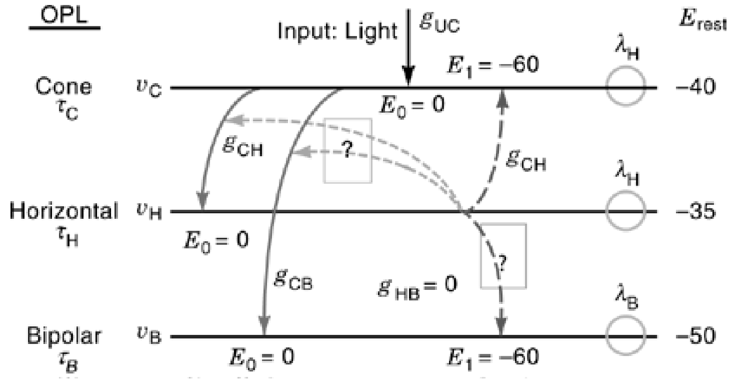

Fig. 6. A three-layer base model of the outer plexiform layer (*OPL*) in the retina. Each biological cell layer is mapped onto a CNN layer that consists of first- or second-order (two mutually coupled first-order) cells. Some pathways and connections are also indicated that still represent a controversial issue in neurobiology.

[(28), Fig. 6] that requires at least a three-layer cellular architecture, it is easy to understand why the physical implementation of cellular arrays has made only the first steps toward a real neuromorphic and programmable visual microprocessor.

Programmable Physics: Diffusion and Wave Models. The PDE-related diffusion-type CNN template derived earlier could be used to emulate a diffusion mechanism on a CNN-UM chip (67). Starting from an arbitrary initial condition (a gray-scale image) the linear heat diffusion results in a progressively low-pass-filtered version of the original image [Fig. 7(a)]. Changing the value of the central term in the feedback matrix ($a_0 = 3$) and the bias value ($z = 2.75$) a nonlinear trigger-wave propagation can also be generated from an arbitrary initial patch [Fig. 7(b)]. As it can be seen the direction of the propagation is reversible (it depends on the sign of the bias value). These two simple cases, based on first-order cells, demonstrate how physics can be easily programmed on a cellular architecture. More complicated wave phenomena can also be generated based on second-order cells as shown in Fig. 7(c)–(f) (see further examples in 38 and 45).

Programmable Chemistry: Self-Assembly in Chemical Substances. There are several classes of CNN templates (4041–42) capable of simulating self-assembly in chemical substances. In the following experiments it will be shown how checkers, patches, stripes, and various combination of these spatial patterns can be formed from random initial conditions on a CNN-UM chip (67) programmed by these templates (Fig. 8).

Cellular Computing in Engineering Designs. In this subsection the image-flow processing capabilities of cellular architectures will be discussed. It is demonstrated how a cellular computer vision system builds on the instructions derived during biological, physical and chemical modeling experiments. This is a radically different approach to synthesizing a "computer vision" algorithm compared to traditional methods since it is based on programmable *receptive fields*, *transport mechanisms* (diffusion and waves), and *self-organization phenomena*.

A cellular computing system is capable of multidimensional information processing. The data can be acquired either in the focal plane (through the on-chip sensors integrated with the cells of the system) or transferred from other array sensor devices (e.g., a CCD camera). In this subsection two such engineering designs based on cellular architectures and analogic cellular computing will also be shortly presented. In these prototypes the flexibility of a recently designed CNN-UM chip (67) and hardware-software environment (100) is exploited building a *focal-plane* and an *online video-flow* processing application.

Cellular Computer Vision. The architectures and physical implementation of cellular systems were discussed in previous sections. Here, some key operators and the algorithmic aspects of cellular computing will

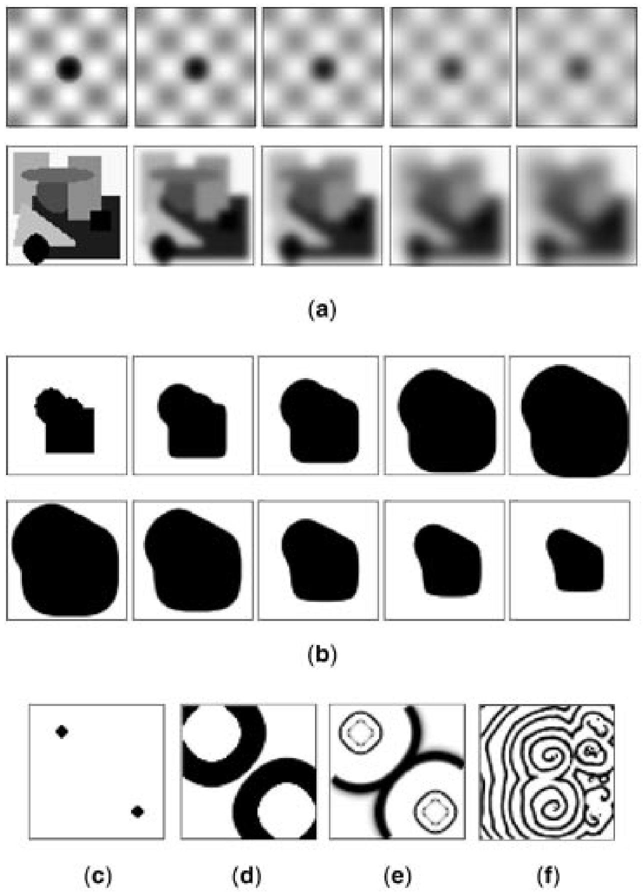

Fig. 7. Programmable physics on a cellular architecture based on first-order and second-order cells: (a) snapshots of a controlled diffusion process, (b) snapshots of a reversed trigger-wave propagation, (c) initial patches for wave generation, (d) traveling waves, (e) autowaves, (f) spiral waves.

be addressed. Figures 9, 10, 11 illustrate how biological, physical, and chemical models can be interpreted as meaningful operators in image processing.

In Fig. 9 programmable receptive fields (edge and corner detection), wave-type computation (reconstruction and connected component detection) and spatial logic (XOR) is used to build up the analogic algorithm solving an object classification problem. In this simple example [Fig. 9(a)] isolated objects that have no corners should be identified marking the horizontal and vertical object extension (coordinates) along the image edges [Fig. 9(g–h)]. The task is solved combining analog operations (either with local or global dynamics) with spatial logic relying on a programmable cellular architecture with local memories.

Various filters can be constructed based on diffusion-type computation as illustrated in Fig. 10. Observe in Fig. 10(b) the low-pass filtered and in Fig. 10(c) the band-pass filtered output of the test image given in Fig. 10(a). Self-organizing phenomena can also be exploited in image processing. By creating a halftone version of a gray-scale image, an adaptive pattern-creating feature helps us to obtain a visually pleasing binary representation of the original image as shown in Fig. 10(d).

The global dynamics of locally connected cellular arrays is a key property in building powerful image flow processing algorithms. Operators derived from transport mechanisms allow us to combine controlled diffusion

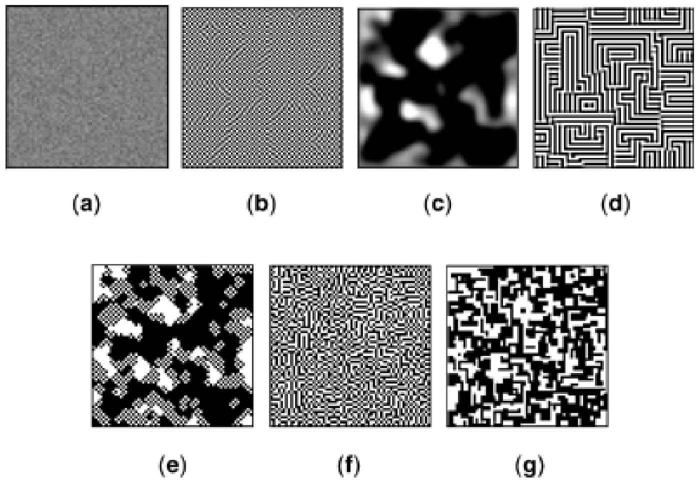

Fig. 8. Programmable chemistry on a cellular architecture pattern-creating self-organization phenomena: (b) checkers, (c) patches, and (d) stripes formed from a random initial condition (a). A colorful combination of the basic motifs can also be generated (e)–(g).

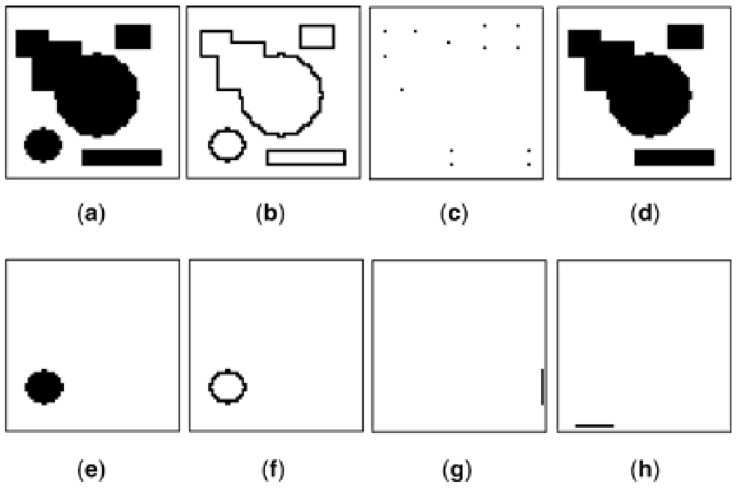

Fig. 9. Programmable receptive fields, wave-type computation, and spatial logic in analogic cellular image processing. The example demonstrates the intermediate steps of an analogic algorithm: (a) original image; (b) result of edge detection; (c) result of corner detection; (d) objects reconstructed from corners; (e) isolated object without corners obtained through spatial logic; (f) edge of the detected object; (g) vertical extension of the detected object; (h) horizontal extension of the detected object.

and waves within the analogic algorithmic framework and make it possible to solve sophisticated detection-reconstruction problems. Such an example is shown in Fig. 11, tracking the contour of the left ventricle where diffusion-type filters were used in noise suppression and trigger waves in boundary detection.

Focal-plane Object Classification. Ultrahigh frame-rate (exceeding 10,000 frame/s) image processing is an unsolved problem with current digital systems of affordable price and size. Both the limited computational

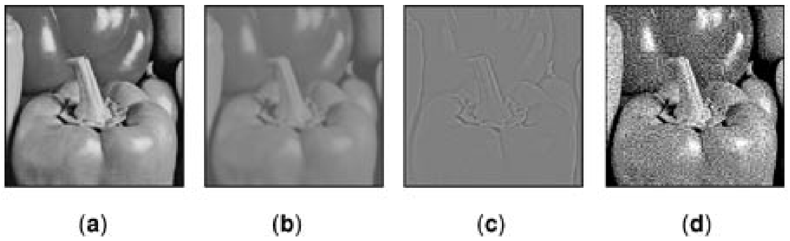

Fig. 10. Diffusion-type computation and self-organization in analogic cellular image processing. (a) Original image; (b) the output of a blurring low-pass filter realized through controlled diffusion mechanism; (c) the output of an edge enhancing band-pass filter realized through diffusion mechanism and spatial arithmetics; (d) halftone (binary representation of a gray-scale image) output obtained through adaptive pattern creating self-organization.

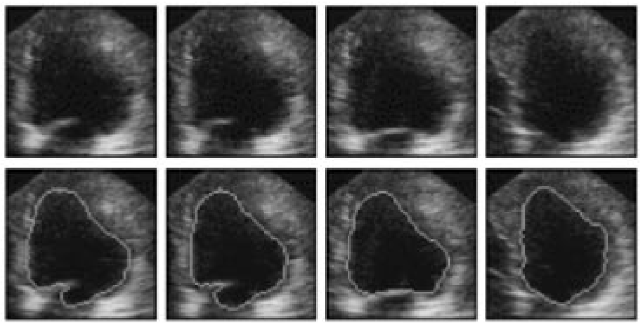

Fig. 11. Using transport (diffusion and wave) mechanisms in analogic cellular image processing. Top row: snapshots of the human heart on an echocardiography sequence. The image-flow shows the spatiotemporal deformation of the left ventricle (the largest heart chamber). Bottom row: detected contour of the left ventricle calculated by an analogic cellular algorithm. Diffusion-type filters were used in noise suppression and trigger-waves in boundary detection. The spatiotemporal consistency was ensured through inter-frame spatial logic operations.

power and the I/O bottleneck (when the image is transferred from the sensor to the processor) represent major obstacles in digital systems.

Cellular neural or nonlinear network (CNN) technology offers a parallel and analogic (combined analog and logic) approach to these problems. If a CNN chip is used as a focal-plane array, even a zero computational load requirement is achievable (the processing is limited by the image acquisition only). The chip (63) acts as a focal-plane visual microprocessor: it acquires image frames parallel through the optical input, transfers them to the processor elements, and performs the analysis also in parallel. In 20 μs, approximately five analog operations (CNN templates) and ten local logic operations can be completed. This makes it possible that even a complex morphological decision can be performed on-chip within two subsequent frames at a 50,000 frames/s operational speed.

The experimental setup of the focal-plane classification system is shown in Fig. 12(a). The CNN platform that carries the chip is mounted on the back panel of a camera (only the optics is used, no shutter is required). On a rotating disk different images are posted and during the experiment these images are projected onto the chip through the lens system of the camera [see the images acquired by the on-chip sensor in Fig. 12(b)].

In the experiment objects are classified based on their local morphological features by a complex analogical algorithm. In Fig. 13 the major subroutines of the algorithm are shown along with their measured on-chip time performance (no transfer and display time included). This demonstration proves that the current CNN system

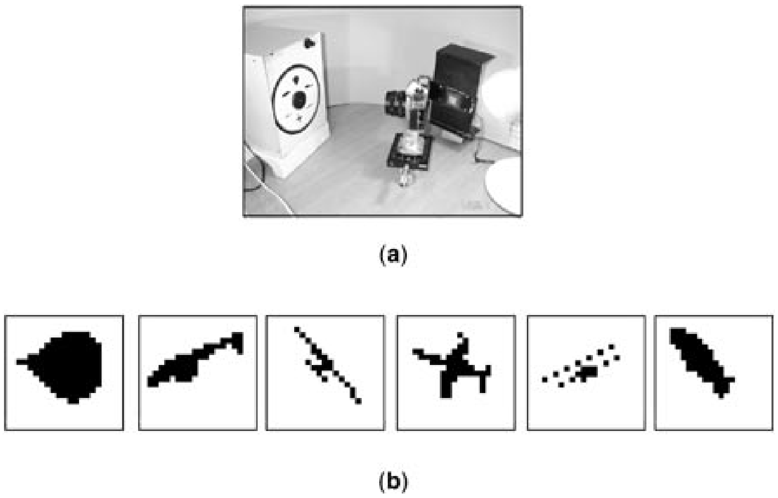

Fig. 12. The ultra high-speed focal-plane array processor system: (a) the experimental setup, (b) the objects silhouettes captured by the 20 × 22 CNN-UM chip.

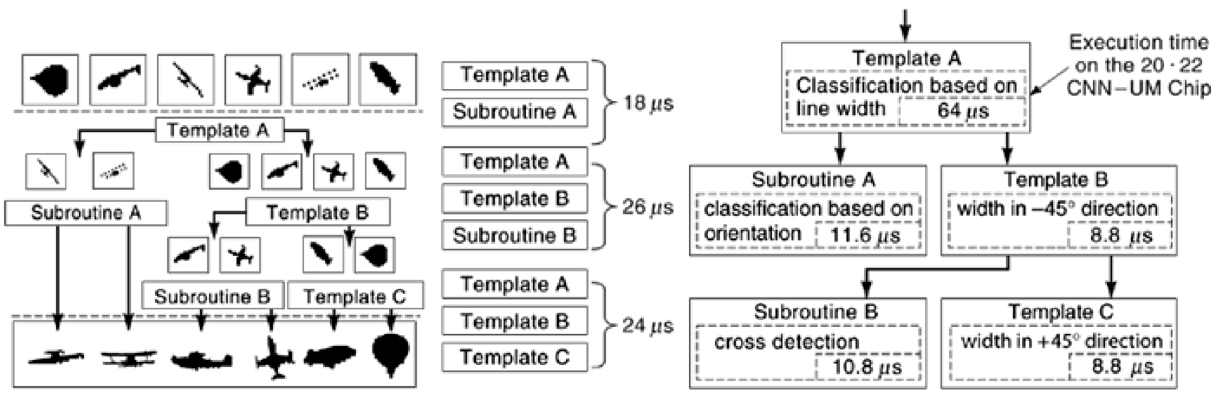

Fig. 13. The flow chart of the object classification algorithm and the measured time performance on the 20 × 22 CNN-UM chip. A binary-morphology-based analogic cellular algorithm ensures rotation and translation invariant detection.

is able to classify six different flying objects (hot-air balloons and airplanes) based on their silhouettes' low-resolution projections on the chip's optical sensors at a speed of approximately 10,000 frames/s.

Online Video-flow Processing. In the last few years particle detection and classification in fluid flows have received considerable interest among the applications requiring image processing at ultrahigh frame rates. For instance, a sensory module capable of identifying the density of debris particles in the oil flow of various engines would enable a cost-effective online monitoring of these systems (e.g., condition-based monitoring of jet engines).

In these applications the CNN chip can be used either as a focal-plane array processor or a video-flow processing visual microprocessor. In the latter case recent feasibility studies and experiments indicate that in a demonstration, the prototype system detection and classification of the particles can be performed online on a 64 × 64 CNN chip (67).

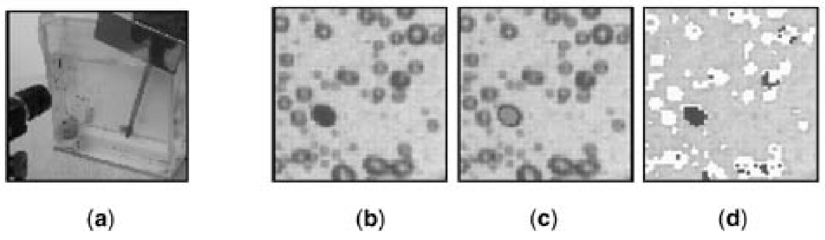

Fig. 14. The online video-flow processing system: (a) the experimental setup, (b) a typical image frame acquired by the CCD camera, (c) identified marble superimposed onto the original image, (d) identified bubbles superimposed onto the original image.

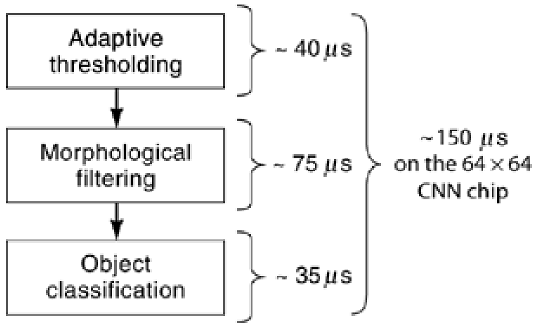

Fig. 15. The flow chart of the bubble-marble classification algorithm with the time requirement of the algorithm subroutines on a 64 × 64 CNN-UM chip. The analogic algorithm is based on binary morphology and diffusion-type operators.

Figure 14(a) shows the experimental setup of the online video-flow demonstration. In a water tank containing bubbles and marbles, a fast turbulent flow is generated. The task is to detect and separate the marbles from air bubbles in each acquired image. A typical image acquired by the CCD camera can be seen in Fig. 14(b), a detected marble in Fig. 14(c), and the identified bubbles in Fig. 14(d).

The algorithm consists of the following processing stages: (i) *adaptive thresholding*, in which all objects are detected in the image field through a spatially adaptive thresholding, (ii) *morphological prefiltering*, in which objects are compared to prototype objects to filter out single bubbles and bubble groups, also to classify the remaining objects into different particle groups, (iii) *object classification*, such that in the last stage objects are classified based on their size and morphology (and a simple statistics is calculated for different object groups).

The on-chip time performance of the subroutines of the algorithm is summarized in Fig. 15 (no transfer and display time included). The demonstration proves that with the current CNN system a morphology-based complex algorithm can be executed during an online video-flow processing.

Summary

Various cellular array architectures that have been proposed for massively parallel computing were discussed. It has been argued that though the physical laws are often described by PDEs, the abstract engineering computing machine is rather a cellular structure that is *discrete in space*, continuous or discrete in time, and is based on analog/discrete computing devices (processors).

It has been shown that though historically rooted from cellular automata (and fully connected neural networks) current implementation trends and biological modeling results justify the intensive research efforts on the field of cellular nonlinear/neural networks. This is especially true when both array sensing and computing are to be addressed within a unified framework. It has been stressed that in order to have a reconfigurable cellular computing array, stored programmability is a key issue in architecture design targeting various application areas in engineering, biology, chemistry and physics.

BIBLIOGRAPHY

1. J. Von Neumann *Theory of Self-reproducing Automata*, Urbana: University of Illinois Press, 1966.
2. J. Thatcher Universality in von Neumann Cellular Model, Technical Report 03105-30-T, ORA, University of Michigan, 1964.
3. C. Lee Synthesis of cellular universal machine using 29-state model of von Neumann, *University of Michigan Engineering Summer Conference*, 1964.
4. E. F. Codd *Cellular Automata*, New York: Academic, 1968.
5. M. Harao S. Naguchi On some dynamical properties of finite cellular automata, *IEEE Trans. Computers*, **C-27**, 42–52, 1978.
6. D. Richardson Tessellations with local transformations, *J. Comput. Systems Sci.*, **6**: 373–388, 1972.
7. T. Toffoli Computation and construction universality of reversible cellular automata, *J. Comput Systems Sci.*, **15**: 213, 1977.
8. H. Yamada S. Amoroso Tesselation Automata, *Information Control*, **14**: 299–317, 1969.
9. H. Yamada S. Amoroso Completeness problem for pattern generation in tesselation automata, *J. Comput. System Sci.*, **4**: 137–176, 1970.
10. H. Yamada S. Amoroso Structural and behavioral equivalencies of tessellation automata, *Information Control*, **18**: 1–31, 1971.
11. S. Wolfram Statistical mechanics of cellular automata, *Rev. Mod. Phys.*, **55**: 601–644, 1983.
12. S. Wolfram Universality and complexity in cellular automata, *Physica D*, **10**: 1–35, 1984.
13. N. H. Packard S. Wolfram Two-dimensional cellular automata, *J. Statist. Phys.*, **38**: 126–171, 1985.
14. G. Crisp A Cellular Automaton Model of Crystal Growth: I) Athracene, Technical. Report, Princeton: Institute for Advanced Study, 1986.
15. E. R. Berlekamp J. H. Conway H. K. Guy *Winning Ways for your Mathematical Plays*, New York: Academic, 1982.
16. T. Toffoli N. Margolis *Cellular Automata Machines*, Cambridge, MA: MIT Press, 1987.
17. W. S. McCulloch W. Pitts A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.*, **5**: 115–137, 1943.
18. J. J. Hopfield Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci.*, **79**: 2554–2558, 1982.
19. J. J. Hopfield Neurons with graded response have collective computational properties like those of two state neurons, *Proc. Natl. Acad. Sci.*, **81**: 3088–3092, 1984.
20. M. A. Cohen S. Grossberg Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, *IEEE Trans. Syst., Man Cybern.*, **13**: 815–826, 1983.
21. T. Kohonen An introduction to neural computing, *Neural Networks*, **1**: 3–16, 1988.
22. S. Grossberg Nonlinear neural networks: Principles, mechanisms and architectures, *Neural Networks*, **1**: 17–62, 1988.
23. L. O. Chua L. Yang Cellular neural networks: Theory and applications, *IEEE Trans. Circuits Syst.*, **35**: 1257–1290, 1988.
24. T. Roska J. Hámori E. Lábos K. Lotz L. Orzó J. Takács P. L. Venetiáner Z. Vidnyánszky Á Zarándy The use of CNN models in the subcortical visual pathway, *IEEE Trans. Circuits Syst.—I: Fundam. Theory Appl.*, **40**: 182–195, 1993.
25. F. Werblin T. Roska L. O. Chua The analogic CNN universal machine as a bionic eye, *Int. J. Circuit Theory Appl.*, **23**: 541–569, 1995.
26. K. Lotz A. Jacobs J. Vandewalle F. Werblin T. Roska Z. Vidnyánszky J. Hámori Some cortical spiking neuron models using CNN, *Int. J. Circuit Theory Appl.*, **24**: 301–314, 1996.

27. A. Jacobs T. Roska F. Werblin Methods for constructing physiologically motivated neuromorphic models in CNN, *Int. J. Circuit Theory Appl.*, **24**: 315–339, 1996.
28. Cs. Rekeczky B. Roska E. Nemeth F. Werblin The network behind dynamic spatiotemporal patterns: Building low-complexity retinal models in CNN Based on morphology, pharmacology and physiology, *Int. J. Circuit Theory Appl.*, **29**: 197–240, 2001.
29. A. Bouzerdoum R. B. Pinter Shunting inhibitory cellular neural networks: Derivation and stability analysis, *IEEE Trans. Circuits Syst.*, **40**: 215–221, 1993.
30. D. Bálya B. Roska E. Nemeth T. Roska F. Werblin A Qualitative Model-Framework for Spatio-temporal Effects in Vertebrate Retinas, in *Proc. 6th IEEE Int. Workshop Cellular Neural Netw. Appl., CNNA 2000*, pp. 165–170, Catania, May 2000.
31. T. Roska D. Wolf T. Kozek R. Tetzlaff L. O. Chua Solving partial differential equations by CNN, in *Proc. Eur. Conf. Circuit Theory Design ECCTD '93*, pp. 1477–1482, Davos, 1993.
32. T. Roska L. O. Chua D. Wolf T. Kozek R. Tetzlaff F. Puffer Simulating nonlinear waves and partial differential equations via CNN—Part I: Basic techniques, *IEEE Trans. Circuits Syst.*, **42**: 807–815, 1995.
33. T. Kozek L. O. Chua T. Roska D. Wolf R. Tetzlaff F. Puffer K. Lotz Simulating nonlinear waves and partial differential equations via CNN—Part II: Typical examples, *IEEE Trans. Circuits Syst.*, **42**: 816–821, 1995.
34. T. Kozek T. Roska A double time-scale CNN for solving 2-D Navier-Stokes equations, *Int. J. Circuit Theory Appl.*, **24**: 49–56, 1996.
35. B. E. Shi Gabor-type filtering in space and time with cellular neural networks, *IEEE Trans. Circuits Syst.*, **45**: 121–132, 1998.
36. B. E. Shi T. Roska L. O. Chua Design of linear cellular neural networks for motion sensitive filtering, *IEEE Trans. Circuits Syst.*, **40**: 320–331, 1993.
37. B. E. Shi T. Roska L. O. Chua Estimating optical flow with cellular neural networks, *Int. J. Circuit Theory Appl.*, **26**: 343–364, 1998.
38. K. R. Crounse L. O. Chua Methods for image processing in cellular neural networks: A tutorial, *IEEE Trans. Circuits Syst.*, **42**: 583–601, 1995.
39. K. R. Crounse *Image Processing Techniques for Cellular Neural Network Hardware*, Ph.D. Thesis, University of California at Berkeley, 1997.
40. K. R. Crounse L. O. Chua P. Thiran G. Setti Characterization and dynamics of pattern formation in cellular neural networks, *IEEE Trans. Circuits Syst.*, **6**: 1703–1724, 1996.
41. P. Thiran K. R. Crounse L. O. Chua M. Hasler Pattern formation properties of autonomous cellular neural networks, *IEEE Trans. Circuits Syst.*, **42**: 757–774, 1995.
42. P. Arena S. Baglio L. Fortuna G. Manganaro Self-organization in a two layer CNN, *IEEE Trans. Circuits Syst.*, **45** (2): 157–162, 1998.
43. T. Szirányi J. Zerubia Markov random field image segmentation using cellular neural network, *IEEE Trans. Circuits Syst.*, **44**: 86–69, 1997.
44. Á Zarándy A. Stoffels T. Roska L. O. Chua Implementation of binary and gray-scale morphology on the CNN universal machine, *IEEE Trans. Circuits Syst.*, **45**: 163–167, 1998.
45. Cs. Rekeczky T. Roska A. Ushida CNN-based difference-controlled adaptive nonlinear image filters, *Int. J. Circuit Theory Appl.*, **26**: 375–423, 1998.
46. T. Yang L. B. Yang C. W. Wu L.O. Chua Fuzzy Cellular Neural Networks: Theory, in *Proc. 4th CNNA'96*, pp. 181–186, Seville, June 1996.
47. S. Paul K. Hüper J. A. Nossek L. O. Chua Mapping nonlinear lattice equations onto cellular neural networks, *IEEE Trans. Circuits Syst.—I: Fundam. Theory Appl.*, **40**: 196–203, 1993.
48. Á Zarándy T. Roska F. Werblin L. O. Chua "intelligent image resolution enhancement by the CNN universal machine and its relevance to TV picture enhancement, in *Proc. 3rd Int. Symp. Nonlinear Theory Appl. NOLTA'95*, pp. 701–706, Las Vegas, December 1995.
49. T. Kozek K. R. Crounse T. Roska L. O. Chua Smart image scanning algorithms for the CNN universal machine, in *Proc. 3rd Int. Symp. Nonlinear Theory Appl. NOLTA'95*, pp. 707–712, Las Vegas, December 1995.
50. T. Szirányi M. Csapodi Texture classification and segmentation by cellular neural network using genetic learning, *Comput. Vision Image Understanding (CVGIP)*, **71** (3): 255–270, 1998.
51. M. Csapodi T. Roska Dynamic analogic algorithms for complex recognition task—A first step towards the bionic eyeglass, *Int. J. Circuit Theory Appl.*, **24**: 127–145, 1996.

52. Á Zarándy F. Werblin T. Roska L. O. Chua Spatial-logic algorithms using basic morphological analogic CNN operations, *Int. J. Circuit Theory Appl.*, **24**: 283–300, 1996.
53. P. L. Venetianer P. Szolgay K. R. Crounse T. Roska L. O. Chua Analog combinatorics and cellular Automata—Key algorithms and layout design, *Int. J. Circuit Theory Appl.*, **24**: 145–164, 1996.
54. Cs. Rekeczky Á. Tahy Z. Végh T. Roska Spatio-temporal nonlinear filtering and endocardial boundary detection in echocardiography, *Int. J. Circuit Theory Appl.*, **27**: 171–207, 1999.
55. Cs. Rekeczky L. O. Chua Computing with front propagation: Active contour and skeleton models in continuous-time CNN, *J. VLSI Signal Proc. Syst.*, **23** (2/3): 373–402, November–December 1999.
56. T. Roska L. O. Chua Cellular neural networks with non-linear and delay-type template elements and non-uniform grids, *Int. J. Circuit Theory Appl.*, **20**: 469–481, 1992.
57. L. O. Chua T. Roska The CNN paradigm, *IEEE Trans. Circuits Syst.*, **40**: 147–156, March 1993.
58. T. Roska L. O. Chua The CNN universal machine: An analogic array computer, *IEEE Trans. Circuits Systems*, **40**: 163–173, March 1993.
59. L. O. Chua T. Roska P. L. Venetianer The CNN is as universal as the turing machine, *IEEE Trans. Circuits Systems*, **40**: 289–291, April 1993.
60. J. M. Cruz L. O. Chua A CNN chip for connected component detection, *IEEE Trans. Circuits Systems*, **38**: 812–817, July 1991.
61. J. M. Cruz L. O. Chua T. Roska A Fast, Complex and Efficient Test Implementation of the CNN Universal Machine, in *Proceedings of the 3rd International Workshop on Cellular Neural Networks and Their Applications*, pp. 61–66, Rome, 1994.
62. S. Espejo R. Carmona R. Dominguéz-Castro A. Rodrigúez-Vázquez CNN Universal Chip in CMOS Technology, *Internat. J. Circuit Theory Applications*, **24**: 93–111, 1996.
63. R. Dominguez-Castro S. Espejo A. Rodriguez-Vazquez R. Carmona P. Földesy Á. Zarándy P. Szolgay T. Szirányi T. Roska A 0.8 μm CMOS 2-D programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage, *IEEE J. Solid State Circuits*, 103–1026, July, 1997.
64. J. M. Cruz L.O. Chua A 16 $\times$ 16 cellular neural network universal chip: The first complete single-chip dynamic computer array with distributed memory and with gray-scale input-output, *Analog Integrated Circuits Signal Processing*, **15** (3): 227–238, 1998.
65. Ari Paasio Adam Dawidziuk Kari Halonen Veikko Porra Minimum size 0.5 micron CMOS programmable 48 by 48 CNN test chip, in *Proceedings of the ECCTD '97*, pp. 154–156, Budapest, 1997.
66. R. Carmona S. Espejo R. Dominguez-Castro A. Rodríguez-Vázquez T. Roska T. Kozek L. O. Chua A 0.5 μm CMOS CNN analog random access memory chip for massive image processing, in *Proceedings of the 5th International Workshop on Cellular Neural Networks and Their Applications CNNA'98*, pp. 271–281, London, 1998.
67. S. Espejo R. Domínguez-Castro G. Liñán Á. Rodríguez-Vázquez A 64 $\times$ 64 CNN Universal Chip with Analog and Digital I/O, in *Proceedings of the 5th International Conference on Electronics, Circuits and Systems ICECS'98*, pp. 203–206, Lisbon, September 1998.
68. A. Paasio A. Kananen K. Halonen V. Porra A 48 by 48 CNN Chip Operating with B/W Images, in *Proceedings of the 5th International Conference on Electronics, Circuits and System ICECS'98*, pp.191–194, Lisbon, September 1998.
69. Á. Zarándy P. Keresztes T. Roska P. Szolgay CASTLE: An Emulated Digital CNN Architecture: Design Issues, New Results, in *Proceedings of the 5th International Conference on Electronics, Circuits and Systems (ICECS'98)*, pp.199–202, Lisbon, September 1998.
70. N. Frühauf E. Lüder G. Bader Fourier optical realization of cellular neural networks, *IEEE Trans. Circuits Systems*, **40**: 156–162, March 1993.
71. C. Mead *Analog VLSI and Neural Systems*, Reading, MA: Addison-Wesley, 1989.
72. L. V. Kantorovich V. I. Krylov *Approximate Methods of Higher Analysis*, New York: Interscience, 1964.
73. C. Rovelli L. Smolin Discreteness of Area and Volume in Quantum Gravity, *Nucl. Phys. B*, **B442**: 593–619, 1995.
74. J. P. Keener Propagation and its failure in coupled systems of discrete excitable cells, *SIAM J. Appl. Math.*, **47**: 556–572, 1987.
75. L. O. Chua CNN: A vision of complexity, *Internat. J. Bifurcation Chaos*, **7** (10): 2219–2425, 1997.
76. T. Toffoli Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics, *Physica* **10D**: 117–127, 1984.
77. P. Perona J. Malik Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis Machine Intelligence*, **12**: 629–639, July 1990.

78. F. Catté P. L. Lions J. M. Morel T. Coll Image selective smoothing and edge detection by nonlinear diffusion, *SIAM J. Num. Anal.*, **29** (1): 182–193, 1992.
79. L. Alvarez P. L. Lions J. M. Morel Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Num. Anal.*, **29** (3): 845–866, 1992.
80. L. Alvarez F. Guichard P. L. Lions J. M. Morel Axioms and fundamental equations of image processing, *Arch. Rational Mech. Anal.*, **123**: 199–257, 1993.
81. P. P. Civalleri M. Gilli On stability of cellular neural networks, *J. VLSI Signal Proc. Syst.*, **23**: 429–436, 1999.
82. H. Harrer J. A. Nossek Discrete-time cellular neural networks *Int. J. Circuit Theory Appl.*, **20**: 453–468, 1992.
83. A. Radványi On the rectangular grid representation of general CNNs, in *Proc. 6th IEEE Int. Workshop Cellular Neural Netw. Appl., CNNA 2000*, pp. 387–394, Catania, May 2000.
84. E. R. Kandel J. H. Schwarz T. M. Jesssel *Principles of Neural Science*, Amsterdam: Elsevier, 1991.
85. Special Issue on Mind and Brain, *Sci. Amer.* **274**, September 1992.
86. J. L. Bradshow N. C. Nettleton *Human Cerebral Assymetry*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
87. T. Roska Dual computing structures containing analog cellular neural networks and digital decision units, *in Proc. IFIP Workshop Silicon Architectures Neural Nets*, pp. 233–244, 1991.
88. T. Roska L. O. Chua On a framework of complexity of computations on flows—Implemented on the CNN universal machine, *DNS-15-1995*, Technical Report, Analogical and Neural Computing Systems Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest 1995.
89. W. D. Hillis The connection machine: A computer architecture based on cellular automata, *Physica D*, **10**: 213, 1984.
90. K. Preston, Jr. M. J. B. Duff *Modern Cellular Automata: Theory and Applications*, New York: Plenum, 1984.
91. C. Mead Neuromorphic electronic systems, *IEEE Proc.*, **78**: 1629–1636, 1990.
92. C. Mead M. A. Machowald A silicon model of early visual processing, *Neural Netw.*, **1**: 91–97, 1988.
93. A. Moini *Vision Chips or Seeing Silicon*, Boston: Kluwer Academic, 1999.
94. A. G. Andreou K. Strohbehn Neural information processing II, in M. Ismail and T. Fiez (eds.), *Analog VLSI of Signal and Information Processing*, New York: McGraw-Hill, Chap. 8, pp. 358–413, 1994.
95. J. Harris C. Koch J. Luo J. Wyatt Resistive fuses: Analog hardware for detecting discountinuities in early vision, in C. Mead and M. Ismail (eds.), *Analog VLSI Implementation of Neural Systems*, Boston: Kluwer Academic, pp. 27–56, 1989.
96. J. Tanner C. Mead A correlating optical motion detector, in *MIT Advanced Research in VLSI*, pp. 57–64, 1984.
97. A. Moore C. Koch A multiplication based analog motion detection chip, *Proc. SPIE, Visual Information Processing: From Neurons to Chips*, **1473**: 66–75, 1991.
98. T. Delbruck A chip that focuses on image itself, in C. Mead and M. Ismail (eds.), *Analog VLSI Implementation of Neural Systems*, Boston: Kluwer Academic, Chap. 7, pp. 171–188, 1989.
99. E. Lange Y. Nitta K. Kyuma Optical neural chips, *IEEE Micro*, **14** (6): 29–41, 1994.
100. T. Roska Á. Zarándy S. Zöld P. Földesy P. Szolgay The computational infrastructure of analogic CNN computing—part I: The CNN-UM chip prototyping system, *IEEE Trans. Circuits Syst. I: Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision*, **46** (2): 261–268, 1999.
101. J. E. Dowling *The Retina: an Approachable Part of the Brain*, Cambridge, MA: Belknap, 1987.
102. Cs. Rekeczky T. Serrano-Gatarredona T. Roska A. Rodríguez-Vázquez A Stored Program 2nd order/3-layer Complex Cell CNNM-UM, in *Proc. 6th IEEE Int. Workshop Cellular Neural Net. Appl., CNNA 2000*, pp. 213–218, Catania, May 2000.

CSABA REKECZKY
TAMÁS ROSKA
Computer and Automation Institute of the Hungarian Academy of Sciences and Péter Pázmány Catholic University