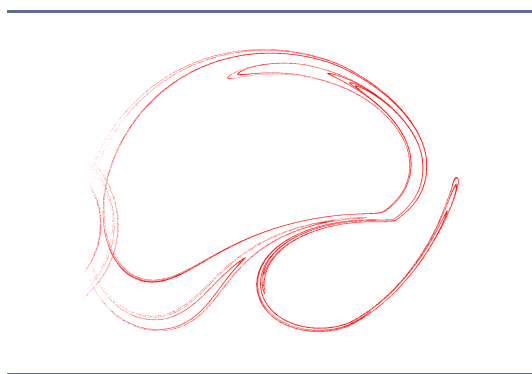


SPATIO-TEMPORAL PATTERNS
AND
ACTIVE WAVE COMPUTING



Ph.D. Dissertation

ISTVÁN PETRÁS

Analogical and Neural Computing Systems Laboratory
Computer and Automation Institute
Hungarian Academy of Sciences

2004

“... How dear are your thoughts to me, O God! How great is the number of them! If I made up their number, it would be more than the grains of sand ...”

(Psalm of David)

Acknowledgement

I would like to say thank you

To my professor, *prof. Tamás Roska*, who helped and supported me in many ways.

To my Parents.

To *prof. Marco Gilli*, who provided me with fruitful discussion, ideas and help.

To *prof. Leon O. Chua*, who accepted me in his Nonlinear Research Laboratory in University of California at Berkeley

To *prof. Joos Vandewalle*, at Electrical Engineering Department of Katholieke Universiteit Leuven where I spent a month.

To my colleagues, with whom I discussed ideas and question and worked together in different projects:

Dávid Bálya, György Cserey, Péter Földesy, Viktor Gál, Péter Jónás, Kristóf Karacs, László Kék, László Orzó, Csaba Rekeczky, István Szatmári, Zoltán Szlávik, Gergely Tímár; Seniors: Ákos Zarándy, Péter Szolgay, Tamás Szirányi.

To *Katalin Keserű* (assistant) and *Gabriella Kék* (secretary) for their kind help.

To the MTA-SZTAKI where I work and spent my Ph.D. student years
(<http://lab.analogic.sztaki.hu/>)

To my professors and teachers in

Veszprémi Egyetem (University of Veszprém, <http://www.vein.hu/>)

Bánki Donáth Műszaki Főiskola (Bánki Donáth Politechnic, <http://www.banki.hu/>)

Erkel Ferenc Gimnázium (Secondary school, <http://www.erkel.hu/>)

Gyulai 1. számú Általános Iskola (Elementary school)

To my town, Gyula (<http://www.gyula.hu/>) where I was born and grew up.

I would like to acknowledge the financial support of the

Hungarian National Research and Development Program: *TeleSense NKFP 2001/02/035*

European Community, “Information Society Technologies” Program: *Locust IST-2001-38097*

Table Of Contents

ACKNOWLEDGEMENT.....	5
1. INTRODUCTION.....	1
2. CNN PARADIGM.....	9
2.1 CNN PARADIGM.....	10
2.1.1 Basic notations.....	10
2.1.2 Mathematical formulation.....	11
2.1.3 Space invariant linear CNN.....	13
2.1.4 Autonomous CNN and PDEs.....	14
2.2 CNN UNIVERSAL MACHINE – ANALOGIC COMPUTER.....	15
2.2.1 Architecture.....	15
2.2.2 Algorithm design.....	16
2.2.3 Hardware implementations.....	17
2.2.4 Application development environment.....	18
3. DYNAMIC PATTERNS.....	21
3.1 CNN MODEL.....	23
3.1.1 Simulation.....	23
3.1.2 Programmable chip measurements – the experimental test bed.....	23
3.2 THE EFFECT OF VERTICAL COUPLING.....	23
3.2.1 Positive vertical coupling ($r > 0$).....	25
3.2.2 Negative vertical coupling ($r < 0$).....	26
3.3 THE EFFECT OF THE CENTRAL TEMPLATE ELEMENT.....	27
3.3.1 Simulation.....	28
3.3.2 Programmed chip measurement.....	32
3.3.3 Spatio-temporal signatures.....	35
3.4 THE EFFECT OF THE CONSTANT INPUT AND INITIAL STATE.....	37
3.4.1 Periodic-chaotic transition.....	37
3.4.2 Stable-periodic transition.....	38
3.5 1D CHAOS.....	39
3.6 ADDITIONAL TRAVELING PATTERN EXAMPLES.....	40
3.6.1 Wave shadow.....	40
3.6.2 “Four pixels” examples.....	42
3.7 SIMULATION TIME VS. REAL-TIME MEASUREMENTS.....	43
4. COMPLEX DYNAMICS IN 1D CNN.....	45
4.1 EFFECT OF BOUNDARY CONDITION.....	46
4.2 NONLINEAR DYNAMICS.....	52

4.2.1	<i>Study of 1D CNN with a selected template</i>	52
4.2.2	<i>Coexistence of attractors in a 4×1 CNN</i>	60
4.2.3	<i>Effect of the perturbation of template parameters on the exponents</i>	62
4.2.4	<i>Higher dimension - Hyperchaos</i>	63
5.	SPECIAL WAVE OPERATOR.....	69
5.1	SHAPE DEFORMATION.....	71
5.2	NUMBER OF LINEAR CELLS.....	73
5.2.1	<i>Horizontally or vertically aligned boundary region</i>	73
5.2.2	<i>Boundary region aligned with $\pi/4$</i>	75
5.3	THE CNN AS A CURVE EVOLUTION COMPUTER.....	78
5.3.1	<i>Examples</i>	85
5.3.2	<i>Connection to bipolar waves</i>	88
6.	APPLICATION AND EXPERIMENTS.....	91
6.1	WAVE COMPUTING.....	91
6.1.1	<i>Direction constrained wave</i>	91
6.1.2	<i>Bipolar waves</i>	93
6.1.3	<i>Curvature and concavity based object decomposition</i>	94
6.1.4	<i>Fork detection</i>	95
6.1.5	<i>Hand orientation detection</i>	96
6.1.6	<i>Texton segmentation</i>	98
6.2	MOBILE NAVIGATION UNIT.....	98
6.2.1	<i>Hardware infrastructure</i>	98
6.2.2	<i>Software infrastructure</i>	101
6.2.3	<i>Optic flow estimation</i>	102
6.2.4	<i>Visual collision avoidance</i>	104
6.2.5	<i>Feature detection</i>	106
6.3	COMPLEX SPATIO-TEMPORAL WAVE EXPERIMENTS WITH CNN.....	106
6.3.1	<i>Mathematical model</i>	106
6.3.2	<i>Stored programmability</i>	107
6.3.3	<i>Wave phenomenon</i>	108
	REFERENCES.....	115
	APPENDIX A.....	127
	APPENDIX B.....	129

1. Introduction

Patterns and waves are so “natural” that we do not pay particular attention to them in our life normally. We do not even notice most of them. Yet, they are inherent phenomena of our world. In general sense pattern can be an arrangement of matter, energy or other substance. Patterns are around and in us everywhere and they play an important role even in our understanding and perceiving. A wave is created when the state or position of a substance locally changes spatially and temporally synchronized in such a way that this local change propagates. In a broad sense the pattern is a distribution of a property of the medium. A kind of mixture of waves and patterns is also possible: a traveling pattern can form a wave, or from the other point of view a wave can have texture (see Fig. 1 and Fig. 2). In this case special local interactions form and maintain an arrangement or texture in the spreading wave. An interesting example is shown in the third thesis of this dissertation.

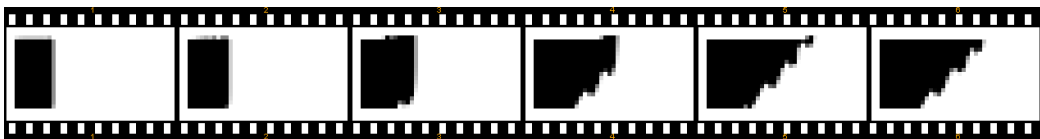


Figure 1. Novel type of traveling nonlinear wave. It propagates to a finite distance. After that distance is reached only ripples along the edge travel. Consecutive snapshots of the active medium.

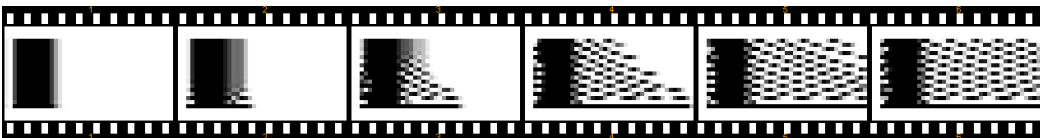


Figure 2. Novel type of traveling pattern. Consecutive snapshots of the active medium.

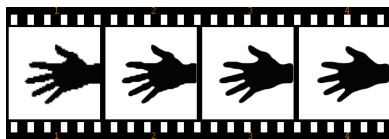


Figure 3. Special trigger wave. Observe the smoothing of the sharp edges along the contour.

When we are talking about waves one usually thinks of the classical waves that spread in conservative systems. An example of this well-known by everybody is the surface waves of water. In this case, the energy pumped into the system is conserved.

Nonlinear wave can also be formed in water. Russel [2] discovered the so-called “solitary waves” or soliton. This is a wave consisting of a single elevation, of height not necessarily small compared to the depth of the fluid. If properly started it can travel mostly without change in a uniform canal. Similarly, in the ocean an isolated giant wave triggered by earthquake travels thousands of kilometres without losing its shape and energy. At present solitons are typical objects in hydrodynamics, acoustics, optics, plasma physics, theory of superconductivity, etc.

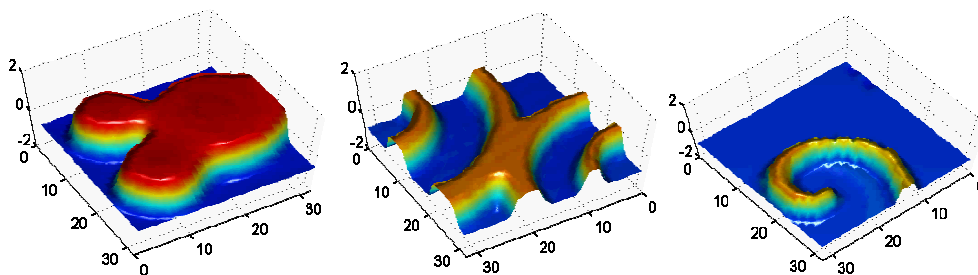


Figure 4. Snapshots of different autowaves. Trigger, target and spiral waves. These results were measured on a programmable analog VLSI array computer.

Nonlinear waves spreading in excitable (non-conservative) medium differs from the “classical” waves. Waves in the classical sense (e.g. electromagnetic waves spreading in vacuum) mostly constitute physically closed system in themselves, i.e. they do not interact with other systems e.g. with a media. It is enough to integrate the state variables of the closed system into the energy balance of such waves and the energy computed taking them into account remains constant during the process. In the case of the excitable waves the conservation of energy is hold only when one considers the state of the wave and the media and the interaction between them. However it seems not to be hold if we consider the wave as a subsystem.

Nonlinear waves spread at the expense of energy stored in the media such a way that an activated point activates the neighboring points of the medium. From this comes the name *autowaves* [3] coined by R.V. Khorlov, as an abbreviation for autonomous waves. The name expresses well the property that the propagation is self-sustaining. The dispersion and decay properties of a spreading autowave are characteristically different from that of the classical ones: the wave does not decay nor the waveform is distorted during the propagation.

Autowaves cannot be reflected from the boundaries, nor can they interfere. When colliding, autowaves annihilate each other. Diffraction is the only property classical and autowaves have in common. Table 1. shows the comparison of classical and autowaves.

Property	Classical waves	Autowaves
1. <i>conservation of energy</i>	+	–
2. <i>conservation of amplitude and wave form</i>	–	+
3. <i>reversibility</i>	–	+
4. <i>reflection</i>	+	–
5. <i>interference</i>	+	–
6. <i>diffraction</i>	+	+
7. <i>annihilation</i>	–	+

Table 1. Comparison of classical and autowave properties [1].

Up to now, several types of active waves were described and studied. Depending on the nature of the observed phenomena different names have been used in literature, such as autowaves, traveling waves or trigger/triggered waves, target waves or concentric waves, spiral waves, scroll waves (in 3D) [31]. See Fig. 4 and 3 for a few examples. There is no well-established convention to find appropriate names for different types of waves. Some authors use the term “trigger wave” for autowaves with annihilation properties [80], other authors use it for a traveling wave that propagates in the medium in such a way that the bistable media flips from one stable state to another and stays in that state for the rest of the time. This kind of behaviour is typical e.g. for phase transition waves. In the following I use the “trigger wave” in the latter sense. A difference between the trigger waves and other autowaves is that when two trigger waves meet they simply merge.

We know several well-described phenomena of active wave propagation. Flame propagation in combustion systems are typical examples. Chemical reaction–diffusion systems can also produce autowaves [4]. Many autowave processes were found in biology. Perhaps the most obvious one is the nerve impulse propagation [5].

An important autowave process can be found in the cardiac muscle [29, 143]. The contraction of the muscle follows travelling wave patterns due to the propagation of electrical impulses in the heart tissue. Sometimes the propagation fails due to some inhomogeneity (dead or damaged cells) or too early reexcitation of freshly excited cells. In these cases the wave can break or anchor at an inhomogeneity and evolves into spiral wave that rotates. Thus heart stops pumping at normal frequency. Instead, high frequency disordered contraction patterns occur that eventually lead to sudden death.

Another rather surprising example of autowaves is the communication method of the cellular slime mold amoeba species [6,7]. Interestingly, the members of a colony of amoeba communicate by spiral and target waves. When no more food (bacteria) is left and they are starving some of them send out spiral and target waves and attract the neighbouring amoebas. Then they transform to spores until food becomes available. When this happens the spores are regenerated into amoebas again [147].

Alan Turing proposed a model to explain mechanism of pattern formation in a reaction–diffusion system in his paper in 1952 [8]. He suggested that chemicals can react in such a way that the steady state of the reaction is a heterogeneous spatial pattern of chemical concentration and that this can be the chemical basis of morphogenesis.

Since the fundamental work of Turing hundreds of papers and several books dealt with the possible explanations of the complexity of forms found in Nature and their corresponding mathematical models [7,9–13,28,151,141]. As a good starting point see [142].

These phenomena detailed above attracted remarkable attention and several researchers have become involved in different fields of sciences in recent decades. This resulted in the emergence of new multidisciplinary research areas. The phenomena have been studied under different guises in literature, such as *order from disorder* (Schrödinger, [16]), *synergetics* (Haken [17–19,141]) *self-organization* (Nicolis & Prigogine, [20]), dissipative structures operating *far from thermodynamic equilibrium* (Prigogine, [9,15]), *edge of chaos* (Langton, 1990 [21]). Another recently developed theory, the *local activity* (Chua [11, 22–26]) offers a unified paradigm for studying these phenomena. It provides precise necessary mathematical conditions for the possible emergence of complexity in energetically active media.

As a result of efforts of many researchers in the field several methodologies and mathematical tools have been developed. A classical approach for modeling is to use PDEs. However, recently, it became apparent that not all phenomena can be reproduced with the continuous PDE models (Keener [27]).

Systems of discretely coupled cells with some kind of transfer process between the cells are often used to model the above mentioned phenomena that occur in living cells, tissues, nervous system, ecosystems, reaction–diffusion systems describing chemical processes. Such discrete modeling frameworks are the cellular automaton model (CA), coupled map lattice (CML [148, 149]), nonlinear lattice (NLL) and cellular nonlinear/neural networks (CNN). See Table 2. for comparison. The latter one is a powerful hardware feasible model, i.e. it can be effectively implemented in analog VLSI chip. Moreover, it turned out that it is a suitable unifying framework for PDEs, CA and NLL since almost all PDEs can be transformed into CNN form with appropriate spatial discretization and the other two can be considered as a special case of CNN [26, 30, 145, 146]. Surprisingly Gilli et. al. showed that CNN dynamics represents a broader class than PDEs [135].

Model	Space	Time	State
<i>PDE</i>	C	C	C
<i>CNN</i>	D	C	C
<i>NLL</i>	D	C	C
<i>CML</i>	D	D	C
<i>CA</i>	D	D	D

Table 2. Comparison of different modeling tools. C = continuous, D = discrete

CNN is a locally connected ensemble of nonlinear dynamical systems called cells. It is discrete in space but continuous in time. Its connections or couplings determine the dy-

namics of the system. CNN provides a well-defined mathematical and physical framework to study the emergence of patterns encoded by the local interactions of identical cells. Chapters 3 and 6.3 of this dissertation present examples of this.

In spatially discrete systems, such as the CNN, a traveling pattern is the assembly of synchronized oscillations of individual cells. Thus it is straightforward to deduce that the trajectory of the system in the state space is characteristic to the pattern. If the attractor that the trajectory moves around is a chaotic one, a rather complicated pattern can be formed. This suggests that coupled oscillations can serve as the generator of forms and patterns. Chapter 3 and 4 of this dissertation deal with this.

Besides the modeling and studying of complex phenomena, CNN makes it possible to conceive novel, inherently fast computation principles. Such parallel computation method is presented in the third thesis and chapter 5 of this dissertation.

Although in the state-of-the-art microprocessors more or less parallelization can be found, algorithms are executed sequentially since the computing principle is based on an abstract mathematical construct called Turing machine that has a fundamentally sequential operation. On the contrary, in active media things happen parallel. CNN is organized typically into a regular grid-like array. Electrical circuit implementation can be the optimum choice since circuits can be well controlled compared to e.g. chemical solutions and current VLSI technology enables the easy mass production of large arrays. The elementary cells of the arrays are identical circuits that operate parallel. If we want to make use of this for computation we must think different from the usual way one got used to at digital computers. One should take into account that the computation goes parallel and the task should be expressed in the form of the available parameters. They are the initial state of the system, couplings and boundary conditions. The result of computation is established as the steady state or any intermediate state of the system developed by the collective behaviour of the array.

The mammalian retina consist of interconnected layers of mostly locally connected cells. CNN shows strong resemblance to this structure. It naturally offers itself as a modeling tool [71]. Hardware implementation makes it a promising basis for further possible retinal prosthesis prototypes [131].

The receptive field is a basic structure found everywhere in the neural pathway. It can be found in the tactile system, retina and in the cortex. The receptive field is the set of neurons from where a neuron receives input [150]. This usually means the local neighborhood of the neuron. With the aid of the CNN several types of artificial receptive field can be built [130].

In the first thesis I present novel 2D traveling patterns exhibiting rich nonlinear dynamics including spatio-temporal chaos. I study the corresponding dynamical system and determine the parameters that control the character and the dynamics of the pattern. I show that the 2D pattern is composed of 1D rows of coupled oscillators. In the second thesis I focus on the analysis of one row. I show that even the boundary conditions can change the dynamic behaviour e.g. from periodic to chaotic.

In the third thesis I present the mathematical analysis of a local curvature controlled trigger wave. I show how it can be used for practical image processing purposes.

In Chapter 6 I present algorithms based on CNN computing and I show how to reproduce basic active wave phenomena in the two-layer CNN Universal Machine (CNN-UM [49]) VLSI chip [59]. Actually, using the CNN-UM as a stored programmable spatiotemporal computer with the CNN dynamics as an elementary instruction, a new world of wave based algorithms is emerging. With these algorithms I have solved some practical problems.

2. *CNN paradigm*

THE Cellular Neural/Nonlinear Network is a two or higher dimensional set of nonlinear dynamical systems [99] organized into a grid. Its elements called cells or “neurons” have local interconnections. The grid may have various topologies such as rectangular, toroidal or even hexagonal. Up to date mainly rectangular and toroidal ones were the subjects of research. Such system can be expressed as a two or higher dimensional array of cells.

Since the original paper [46] several articles have dealt with different theoretical issues including stability problems and novel phenomena, with applications and hardware implementations. CNN has been appeared to be suitable theoretical and hardware framework [50-59] for computing partial differential equations [75,76], reaction–diffusion systems and high speed near-sensory image processing techniques [56,64-74].

In the CNN the couplings, the initial state of the system and the boundary conditions determine the operation of the array. The couplings represent the weighting of the effects of the neighbouring cells on each other.

Cellular Neural Networks [46,47], the CNN paradigm [48] and the analogic computer, the CNN Universal Machine [49], provide a new computational approach to spatiotemporal computing in particular image processing.

The concept was originally invented by circuit theorists thus it promotes hardware realizations. Due to its local connectivity, the principle fits in with the current VLSI technology. Moreover, its local processing manner naturally makes it a near–sensory technology that enables ultra–high speed parallel computation without I/O bottleneck [128,129].

2.1 CNN Paradigm

Since the inception of the CNN several enhanced and modified models were proposed. In the following I briefly review the most important variants.

2.1.1 Basic notations

The CNN is defined by two mathematical constructs according to [26]:

- I. A spatially discrete collection of continuous nonlinear dynamical systems called *cells*, where information can be coded into each cell via three independent variables called *input*, *threshold*, and *initial state*.
- II. A *coupling law* relating one or more relevant variables of each cell to all neighboring cells located within a prescribed sphere of influence $N_{r(ij)}$ of radius r centered at ij (see Fig. 5).

$$N_{r(ij)} = \left\{ C(k,l) \mid \max_{1 \leq k \leq M, 1 \leq l \leq N} \{|k-i|, |l-j|\} \leq r \right\}, \text{ where } C(k,l) \text{ denotes the } j \text{ cell in the } i \text{ row.}$$

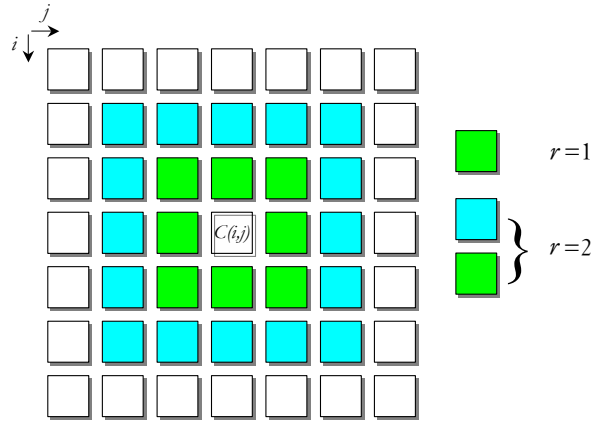


Figure 5. The meaning of sphere of influence $N_{r(ij)}$ of cell $C(i,j)$ in rectangular grid. $1 < i \leq M$, $1 < j \leq N$.

It is important to note that the boundary condition is also possible information storage and therefore it is necessary to prescribe it. Indeed, as it will be shown in Chapter 4, it is an important system variable and bifurcation parameter that controls the dynamics of the whole system. Boundary condition is represented in the form of virtual cells: $V = \{C(i,j) \mid i=0, M+1, j=0, N+1\}$.

Boundary condition types:

- I. Fixed (Dirichlet)
 $C(i,0)=I_1, C(i,N+1)=I_2, C(0,j)=I_3, C(M+1,j)=I_4$

- II. Zero flux (Neumann)
 $C(i,0) = C(i,1), C(i,N+1) = C(i,N), C(0,j) = C(1,j), C(M+1,j) = C(M,j)$
- III. Periodic (Toroidal)
 $C(i,0) = C(i,N), C(i,N+1) = C(i,1), C(0,j) = C(M,j), C(M+1,j) = C(1,j)$

2.1.2 Mathematical formulation

Throughout my theses, I use the one-layer CNN model with first order elementary cells organized into regular grid.

First order CNN or *first order (core) cells* means that the elementary cells of the CNN are described by first order ordinary differential equations ignoring the couplings between the cells. *Second* or *third order cells* mean second or third order differential equations describing the elementary cell.

CNN with second order cells is introduced in Chapter 6. Indeed, higher order cells can be formulated with multiple layers and vice versa. Therefore, the two-layer CNN with first order cells and single layer CNN with second order cells are practically equivalent. In the following I give an overview of one-layer, first order, hardware feasible version of CNN.

General first order CNN state equation:

$$C_x \frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x} x_{ij}(t) + \sum_{k=-r}^r \sum_{l=-r}^r A(i,j;k,l; y_{i+k,j+l}(t)) + \sum_{k=-r}^r \sum_{l=-r}^r B(i,j;k,l; u_{i+k,j+l}) + \sum_{k=-r}^r \sum_{l=-r}^r C(i,j;k,l; x_{i+k,j+l}) + \sum_{k=-r}^r \sum_{l=-r}^r D(i,j;k,l; y_{i+k,j+l}; x_{i+k,j+l}; u_{i+k,j+l}) + z_{ij} \quad (2.1)$$

where

C_x, R_x are (linear) capacitance and (linear) resistance. They determine the time constant of the CNN: $\tau = R_x C_x$. With no loss of generality they are considered to be 1.

A is the operator for the connections from the outputs of the neighbors including its own.

B is the operator for the connections from the inputs of the neighbors including its own.

C is the operator for the connections from the state.

D is the operator for the mixed connections.

x_{ij} is the *state* of the cell being in the i^{th} row and j^{th} column.

y_{ij} is the *output* of the cell.

u_{ij} is the *input* of the cell.

z_{ij} is the *bias* (also referred to as current or threshold) of the cell that adds a constant value to the state. It can be space variant or invariant. In the latter case it is a single number.

r is the neighbourhood. It typically ranges from one to three.

i, j are the cell indices

M, N are the vertical and horizontal dimensions respectively.
 $1 < i \leq M, 1 < j \leq N$

Output equation:

$$y = f(x) \equiv \frac{1}{2}(|x+1| - |x-1|) \quad (2.2)$$

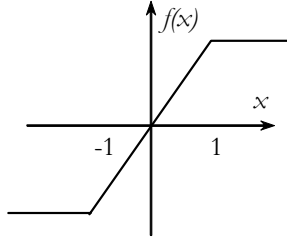


Figure 6. Piece-wise linear nonlinearity used in the original Chua-Yang model.

The operators A, B, C, D are called (*cloning*) *templates*. They are written usually as matrices, their elements can be functions or constants. They represent the strength of the coupling between the neighbouring cells. Considering the case when the cells are organized into a 2D array, the *output* (or *state*) of the system is a two-dimensional image in which each cell corresponds to one pixel of the image. The *input* u_{ij} of the CNN is also an image. The *bias* z_{ij} that can be a single constant or an image that adds constant bias to each cell.

The CNN paradigm allows general synaptic coupling. We may introduce couplings depending on

- the output and input $A(i;j;k;l, y_{i+k, j+l}(t)), B(i;j;k;l, u_{i+k, j+l}(t))$ respectively.
- the state (template C). Intensive research was done related to this coupling type by Arena [105, 139].
- mixed variables coupling (template D). In this case the coupling is a function of mixed variables x, y, u . Such connection can be for example when the coupling strength is: $D(\cdot) = D(x_{ij} - x_{kl}), D(y_{ij} - y_{kl})$ or $D(x_{ij} - u_{kl})$. Shi presented a nonlinear statistical filtering based on this type of coupling [138]. Rekeczky also did in-depth research related to this type of coupling [67]. He developed operators and algorithms for local statistics based filtering [68].

For examples and additional coupling types such as delay type ones see [62] where the interested reader finds several operators designed for specific tasks.

In the most general case templates may vary with position (i, j) and can be nonlinear function of variables $x_{ij}, x_{kl}, y_{ij}, y_{kl}, x_{ij}, u_{kl}$. However, due to VLSI realization issues CNNs with linear and space invariant coupling are the most widely studied ones. Namely, when $A(i;j;k;l, y_{i+k, j+l}(t)) = a_{kl} y_{kl}, B(i;j;k;l, u_{i+k, j+l}(t)) = b_{kl} u_{kl}$.

The first order CNN with nonlinear function (2.2), with $r = 1$, with linear templates A , B and with bias z is called *standard CNN*. The following linear template matrices show the couplings for the standard CNN. The elements of the matrices are constant.

$$\mathbf{A} = \begin{bmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{00} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{00} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} \quad z_{ij} \quad (2.3)$$

Together with the initial state, input, boundary conditions and possible with a bias map these 19 numbers characterize completely the whole array of standard CNN.

2.1.3 Space invariant linear CNN

This type of CNN model is the most widely used both for theory and physical implementations. In the following I present two variants. The first one is more often is the subject of theoretical works while the second that is the modification of the original so called Chua–Yang model is frequently used in chips.

Chua–Yang model

This model was described in the original paper [46]. A_{kl} and B_{kl} are matrices with constant coefficient like in eq. (2.3). The general state equation (2.1) simplifies to the following:

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} y_{i+k, j+l}(t) + \sum_{k=-r}^r \sum_{l=-r}^r B_{kl} u_{i+k, j+l} + z_{ij} \quad (2.4)$$

The nonlinear function remains the same as eq. (2.2). The original Chua–Yang cell circuit is shown in Fig. 7.

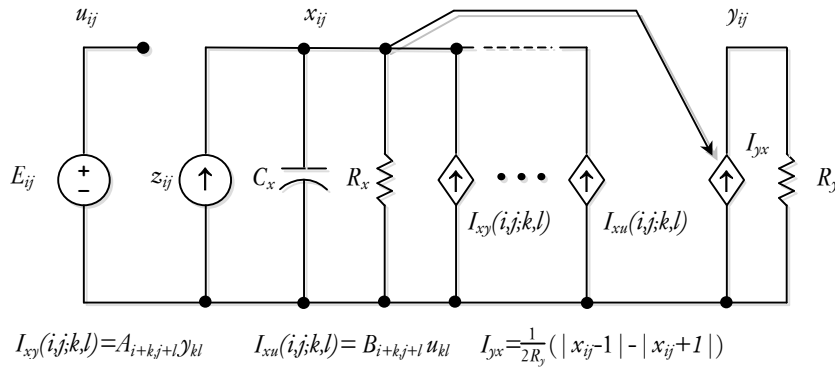


Figure 7. Standard CNN cell core circuit [46]. Notations are according to eq. 2.1. R_x and R_y are linear resistors; z_{ij} is an independent voltage source; $I_{xu}(i,k;k,l)$ and $I_{xy}(i,k;k,l)$ are linear voltage-controlled current sources with the characteristics $I_{xy}(i,j;k,l) = A_{i+k, j+l} y_{kl}$ and $I_{xu}(i,j;k,l) = B_{i+k, j+l} u_{kl}$; $I_{yx} = f(x)$ is a piecewise-linear voltage-controlled current source with characteristic $y = \frac{1}{2R_y} (|x_{ij}-1| - |x_{ij}+1|)$; u_{ij} is an independent voltage source.

Full range (FSR) model

This is a modification of the original model due to the VLSI implementation issues [97]. The difference is that the state is always bounded and the same as the output. This modification does not change the qualitative behaviour in most of the parameter range. However Gilli and Corinto [136,137] has shown that even if the two models exhibit similar properties, they are not topologically equivalent, i.e. there exist sets of identical parameters for which they present a qualitatively different dynamic behaviour. As a consequence, all the results based on the original Chua-Yang model, should be carefully checked, before extending them to the Full-Range model. The state equation is the following:

$$\dot{x}_{ij}(t) = -g(x_{ij}(t)) + \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} x_{i+k, j+l}(t) + \sum_{k=-r}^r \sum_{l=-r}^r B_{kl} u_{i+k, j+l} + z_{ij} \quad (2.5)$$

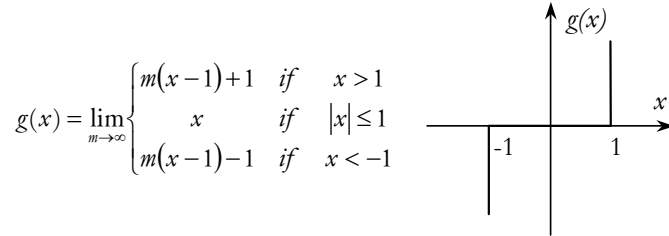


Figure 8. Full-range model's ideal nonlinearity.

Fig. 8 shows the ideal nonlinearity. However, it can not be realized physically, instead a “less hard” real nonlinearity was implemented. See Fig. 15 on page 23.

An illustrative example for the different qualitative behaviour can be when the dynamics of the array is e.g. chaotic with a given parameter set in the Chua–Yang model and periodic with the FSR model and vice versa.

2.1.4 Autonomous CNN and PDEs

CNNs having no input represent an important subclass. Several papers related to them showed their importance. They can be the suitable medium for modeling and generating many pattern formation and active wave phenomena mentioned in the Introduction. Several types of nonlinear waves have been reproduced and analyzed on this structure depending on the complexity of the elementary cells [31, 59, 77-82]. In literature results were presented with first [64, 83, 85], second [59, 84] and third order CNN cells [31, 152–154]. The so-called Chua circuit was used as a third order core cell. It is a well-described simple yet dynamically rich three-dimensional dynamical system [155].

The classical physics postulates in most of the cases the local character of interaction. Waves as global effects arise from local effects of the dynamic evolution of the media. Appropriate mathematical description of local dynamics of the media is a partial differential equation. Most of the PDEs cannot be solved in closed form. In these cases, one uses numerical integration. To do this spatial discretization needs to be performed. This trans-

forms the PDE into a set of ODEs. This means that the original spatially continuous system is transformed into an array of small, discrete, interacting systems.

Interestingly it turned out that spatially discrete system represents a broader class than PDEs [135]. Not all phenomenon exhibited by discrete systems can be reproduced in PDEs (Keener [27]).

The CNN paradigm is a natural and flexible framework to describe locally interconnected, simple, dynamical systems that have lattice like structure. CNN offers a massively parallel and analog solution for the computation of discretized PDEs. Programmability of interactions and boundary conditions provides flexibility for treatment of various problems.

It should be noted that the discretized analog solution provides a different precision compared to that of the digital processors. While in case of digital computing we can define e.g. 32 bit precision for each (virtual) cell in our analog processor equivalent precision can not be defined since each cell has analog dynamics. The elementary cell has virtually “infinite” precision though due to the fabrication process the cells have a distribution of parameters. This should be considered and requires a different thinking when one designs algorithms and applications.

2.2 CNN Universal Machine – Analogic computer

To build algorithms some kind of programmability is obviously required. The earlier neural network chips (e.g. Intel’s ETANN i80170NX) were not truly successful because the reprogramming took order of magnitude longer time than the computing itself due to the high number of synaptic couplings, which heavily limited the range of applications. The space invariant linear CNN requires only the same 19 numbers to be stored in each cell. Thus reprogramming takes approximately the same time as the transient of a CNN with simple, non-propagating template setting. This template constitutes the elementary analog instruction of the whole CNN array. This means that if $r=1$, 19 numbers are enough to describe the operation of the whole array. To compose programs based on these instructions, additional devices and circuitry should be added to the cells and the array in addition to local and global logic.

2.2.1 Architecture

The CNN Universal Machine array computer [49] was designed according to these principles. Local Analog Memories (LAM) and Local Logic Memories (LLM) were added to the cells to store intermediate results, which means a great advantage in computational speed. Local Logic Unit (LLU) was added to perform preprogrammed logic (AND, NOT, OR, XOR, etc.) on the intermediate results. Thus logic operations together with the analog ones form the analogic instruction set of the array processor.

On the other hand, at the array level a Global Analogic Programming Unit (GAPU) was added to the machine. It includes Analog Program Register (APR) to store the analog program and Logic Program Register (LPR) to hold the control sequences for logic program of the individual cells. The Switch Configuration Register (SCR) stores the settings of

switches that control the behaviour and functionality of the cells. The Global Analogic Control Unit (GACU) stores the sequence of analogic instructions that forms an algorithm. It controls the data transfers, timing, program execution, and communication with other devices.

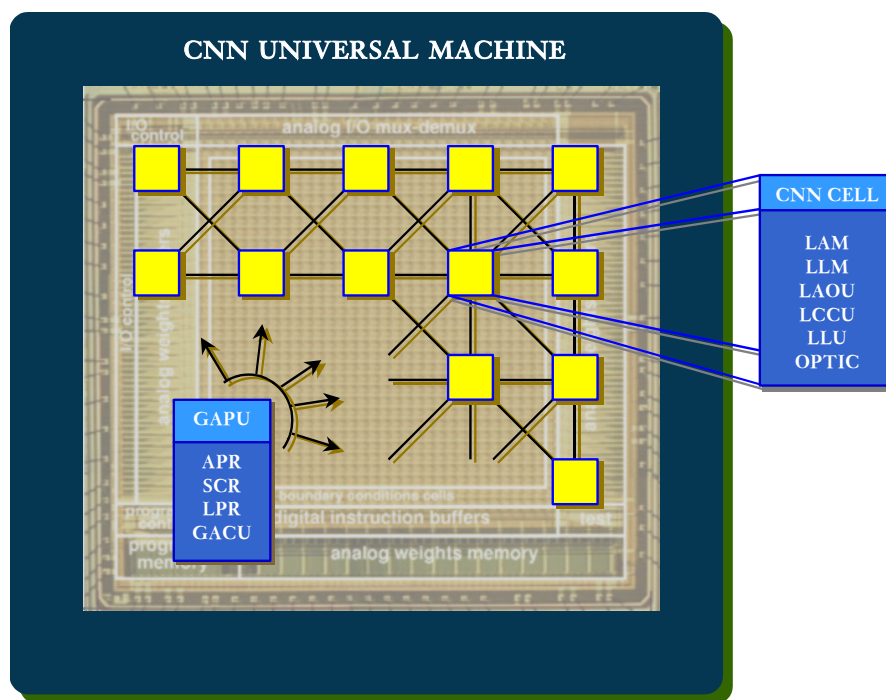


Figure 9. CNN Universal machine.

2.2.2 Algorithm design

Contrary to usual digital computing, the application of the CNN paradigm and analogic algorithms require a completely different way of thinking. Instead of sequentially executed arithmetic and logic instructions executed for individual pixels, the CNN analogic programs consist of the combination of parallel logic and spatiotemporal analog operations. This analog operation defined by a template can perform complex computational tasks in a single dynamic wave or process as illustrated in Chapter 5. Such spatiotemporal processing principle is inherent in biology, for example such computing method can be found in the retina [70,131].

In the algorithmic design it is important to keep in mind the key principles of CNN processing:

- I. Distributed parallel processing based on mainly local interactions.
- II. Local storage of necessary information and intermediate results.

- III. Decisions are based on global properties e.g. all pixels are white. This implies the reduced information transfer since this kind of detection can be implemented easily.

2.2.3 Hardware implementations

Since the inception of the CNN paradigm several successful hardware implementation has been realized due to the intensive interoperation of circuit theorists, mathematicians, computer scientists, circuit designers and neurobiologists [50-59]. The array dimension being 12×12 at the beginning [50] has been increased more than tenfold to 128×128 within ten years [55]. After the first steps a hardware accelerator board was designed to put into PC. Later binary input/output chips were fabricated. Latest versions have grayscale input/output and also have optical input that enables near-sensory processing. Table 3 shows the comparison of the main characteristics of CNN-UM chips; Fig. 10 m shows the photos of some CNN-UM chips; Table 4 shows some characteristics of the new CNN-UM based smart camera system produced by Analogic Computers Ltd [108].

Place of design	Berkeley & Munich	Seville	Leuven	Seville	Berkeley	Helsinki	Seville	Seville	Seville	Budapest
Date of design	1993	1994	1995	1995	1996	1997	1998	2000	2001, 2004	2004
Array size	12×12	32×32	20×20	20×22	16×16	48×48	64×64	32×32	128×128	$n \times 40$
Cell type	DTCNN	Full-range	Chua-Yang	Full-range	Chua-Yang	Chua-Yang	Full-range	Full-range, complex cell	Full-range	Full-range
Technology	2μ	1μ	0.7μ	0.8μ	1μ	0.5μ	0.5μ	0.5μ	0.35μ	0.35μ
Time constant (τ)	300ns	-	4.8μ s	400ns	27ns	50ns	250ns	two < 100 ns, one is variable	250ns	-
Input	analog	binary & optical	analog	binary & optical	analog	binary	analog	analog	analog & optical	digital
Output	binary	binary	analog	binary	analog & binary	binary	analog & binary	analog	analog	digital
APR	external	8	external	8	external	1	32	32	32	20
LLU	AND	programmable	-	programmable	programmable	programmable	programmable	programmable	programmable	programmable
LLM	2	4	-	4	2	2	4	4	-	$3 \times 40 \times 12$
LAM	-	-	-	-	-	-	4	4	8	3×40

Table 3. Comparison of the performance characteristics of different CNN Universal Chips.

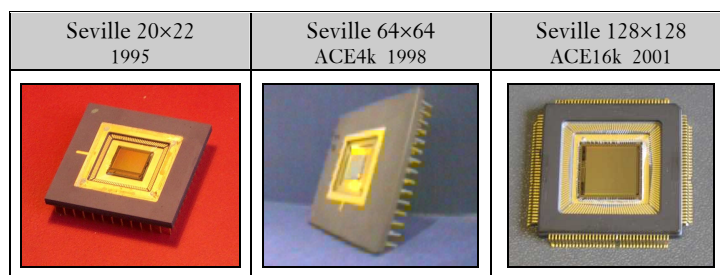


Figure 10. Selected CNN Universal Chips.

Product Name	Computing power	Digital Processor	Digital Memory	Neural Processor	Frame rate
Bi-i	1600 MIPS + ~ 1 TerraOPS using ACE16K	TMS320C6202, 250 MHz	2MB Flash, 16 MB SDRAM	ACE16K - 128x128 sensor processor	28 - 2000 fps, up to 10000

Table 4. Technical data of the latest CNN-UM equipped Bi-i smart camera system.

Other types of implementation are also possible. Such feasible example can be the optical one [156,157] where the computation is done by the light using Fourier optics. Its kernel processor is a novel type of high performance optical correlator based on the use of bacteriorhodopsin (BR) as a dynamic holographic material. This optical CNN implementation combines the optical computer's high speed, high parallelism and large applicable template size with the flexible programmability of the CNN devices (Fig. 11).

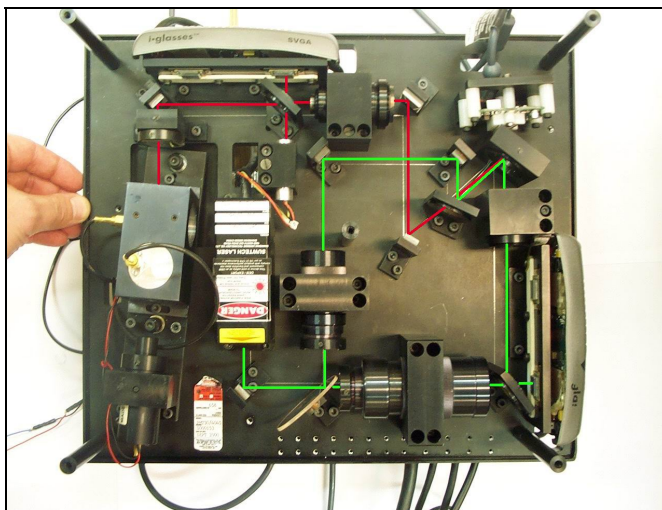


Figure 11. Laptop-size POAC (Programmable Opto-electronic Analogic CNN Computer).

2.2.4 Application development environment

In our laboratory we tested successfully almost all CNN-UM chips (see Fig. 10 for images of some chips). For this we made prototyping hardware environments [60]. The first component of the system is a host PC that displays the results and allows the user to interact. The next component is a DSP module that communicates both with the PC and with the CNNUM. The CNN chip is hosted by a transparent hardware-software interface that makes it possible to interchange various chips [133,134]. Earlier versions of the system required a PC, but the latest version is a standalone system that can operate even on battery connected to the internet through a wireless adapter (see Figure 12).

The system has different levels of software interface. At the lowest level man can use C++ library calls [63] which execute specific tasks on the chip and DSP. The next abstrac-

tion level is the so called AMC that stands for *analogic machine code*. This has the same functionality as the C++ library calls but it is interpreted in runtime. It is possible to use a high-level language called *Alpha*. The CNN software library [62] contains several CNN templates and algorithm examples.

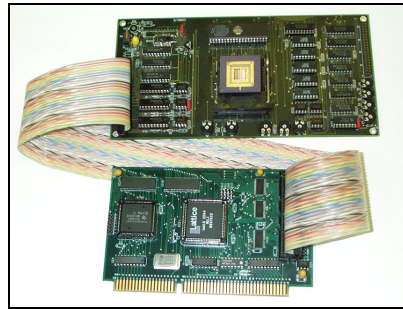
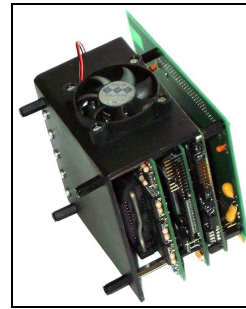
*CCPS**ACE-BOX*

Figure 12. PC-embedded hardware hosts of development environment.

*Bi-i V1**Bi-i V2*

Figure 13. CNN-UM smart cameras as standalone hardware hosts of development environment.

2. CNN PARADIGM

3. *Dynamic patterns*

First Thesis

Spatio-temporal signatures in CNN

I discovered a linear, space invariant 2D template class with few nonzero elements that can produce complex, chaotic spatio-temporal behavior depending on the template parameters and the input of the array. It generates spatially bounded or unbounded traveling patterns according to the parameters. I introduced the “Spatio-temporal Signature” – a still image that is the snapshot of the output – as a descriptor for the dynamic state of the array. This image reflects the temporal history of the dynamics in space due to the propagating effect.

I gave principles for the template design. I gave a 1D template of which corresponding CNN exhibit complex, chaotic behavior depending on the input.

NUMERICAL simulations of spatial-temporal chaotic systems require enormous digital computing power, but even so this is the usual analysis tool because it offers the advantage of easy experimentation via programming. Until now, the physically implemented chaotic circuits were “hard-coded”. The analogic cellular computing paradigm [99,49,107]

places the spatial-temporal dynamics into array computer architecture. Using the ACE4K test-bed [98-99] it is possible to make programmable real-time experiments and uncover new complex dynamic behaviours in the Cellular Nonlinear Network.

The Cellular Nonlinear Network [46-49] was introduced in Chapter 2. In this thesis I consider its standard two-dimensional one-layer, first order version except where explicitly noted.

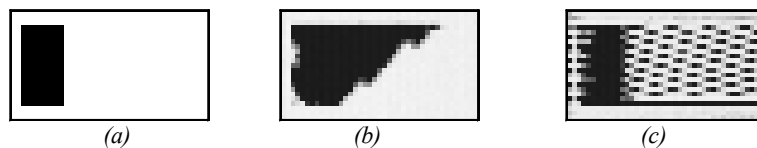


Figure 14. Typical pattern classes. (a) The input and initial state, (b) snapshot of the output when the extra coupling is greater than zero, (c) snapshot of the output when the extra coupling is less than zero (Chip measurements)

The qualitative theory of nonsymmetric feedback (A) template were first exposed in [106]. Later, several papers studied the operation of the CNN with non-symmetric or sign-antisymmetric templates [100-102]. They described some necessary conditions under which propagation effects occur or the solution is periodic. Other works investigated the pattern formation properties of the CNN [91] or studied the complex behaviour [103-105] of the CNN. However, only a few works dealt with the case when there is a constant input [64]. With constant input, we are able to change the local dynamics within the array and to localize the propagation effect into a certain region according to the extent of the input pattern. By using a constant input as a “seed”, different shapes can be generated depending on the properties of the template.

In the following I present some experimental analysis of a simple antisymmetric template class in that case when we add only one extra coupling below the central element. I introduce a basic template class and show how the behaviour of the CNN changes from stable to chaotic states at different values of the extra coupling. Fig. 14 shows two basic pattern classes that are generated with two different values of the key template element (the extra coupling is greater or less than zero). Throughout this thesis *black color* means “+1” and *white color* means “-1” in the images. The input is the same as the initial state in every measurement and example. Boundary conditions are set to *zeroflux* in the simulator and to -1 in the chip measurements, except where explicitly noted.

Section 3.1.1 describes the CNN *model* of the simulation and of the chip. Section 3.2 presents the basic *pattern classes*. Section 3.3 and Section 3.4 describes the *effect* of the *self-feedback* and of the *input and initial state* respectively. Section 3.5 presents an example of a *1D CNN* with first order cells that exhibits *chaotic* behaviour. Section 3.6 shows some *additional examples* of traveling pattern classes.

Throughout this analysis I only consider CNNs with *space invariant* templates i.e. the same template matrix describes the local couplings for each cell. In the model (3.1) we have a nonzero input u that adds a constant value to each cell.

3.1 CNN Model

The models used in this thesis were introduced in Chapter 2. Therefore I only repeat the basic equations.

3.1.1 Simulation

The mathematical model of the simulation of the CNN dynamics is the following:

$$\dot{x}_{ij}(t) = -g(x_{ij}(t)) + \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} x_{i+k, j+l}(t) + \sum_{k=-r}^r \sum_{l=-r}^r B_{kl} u_{i+k, j+l} + z \quad (3.1)$$

I use the so-called full range model [97]. In eq. (3.1) x_{ij} denotes the state, A_{kl} is the feedback template matrix, B_{kl} is the control template matrix, that describes the effect of the constant input u_{kl} , and z is the offset or bias. The integration method throughout the simulation was an *implicit Euler method*. In all cases the boundary cells are set to *zeroflux*, except if explicitly noted differently.

3.1.2 Programmable chip measurements – the experimental test bed

The chip experiments were made on the ACE4k test bed. The model of the CNN is the following:

$$\dot{x}_{ij}(t) = -g'(x_{ij}(t)) + \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} x_{i+k, j+l}(t) + \sum_{k=-r}^r \sum_{l=-r}^r B_{kl} u_{i+k, j+l} + z \quad (3.2)$$

On the chip, the ideal nonlinearity (see Fig. 8) is approximated by a “less hard” nonlinearity (See Fig. 15).

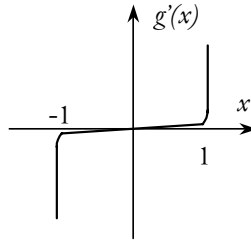


Figure 15. “Less hard” nonlinearity $g'(\cdot)$

3.2 The effect of vertical coupling

In the following, I show how an extra coupling added to a vertically uncoupled template changes the behavior of the system. At first, let us consider a one-dimensional “CCD-like” template of which solution is periodic (See Template 1 and Fig. 16).

$$\text{Template 1: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.6 & 0.3 & -0.6 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.1$$

During the transient, cells along the right hand side border of the constant input pattern act like oscillators. The oscillators are only coupled horizontally and the rows operate independently.



Figure 16. The time evolution for Template 1, snapshots of the state. Observe that the propagation decays *spatially* after a few pixels, but the oscillation remains. The initial state is the same as the input. (Simulated results, size: 41×23)

The oscillation propagates to the right along the rows starting from the triggering constant input (black pixels), and depending on the template values, it stops (dies) after a certain distance or endures until the edge of the array.

Template 2 shows the general form of the nonsymmetric template with an added vertical coupling. When $sq < 0$ the template is called *sign-antisymmetric*. This is shown in Template 3 with $s = -q$. By introducing an extra template element r below the central one (that is denoted by p) the homogeneous propagation and oscillation disappear. We get some structured pattern. The character of the structure depends on the sign of the extra template element r . (See Fig. 14 and Table 5.)

$$\begin{array}{l} \text{Template 2:} \\ \text{general form} \\ s \neq q \end{array} \quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & q \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{z} = \mathbf{z}$$

$$\begin{array}{l} \text{Template 3:} \\ \text{antisymmetric} \\ s = -q \end{array} \quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & -s \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{z} = \mathbf{z}$$

Coupling sign	Traveling pattern	Snapshots
positive: $r > 0$	solid inner part, oscillating border cells	
negative: $r < 0$	texture like traveling pattern	

Table 5. Categorization of templates containing one extra coupling. The effect of the sign on the shape of the pattern.

3.2.1 Positive vertical coupling ($r > 0$)

If the extra coupling is positive, a pattern is formed which is solid inside, however its right border is oscillating. At the beginning, the input pattern propagates oscillating to the right until a certain extent, and then the global propagation stops and cells along the right border continue oscillating (at certain parameter setting the left border can also oscillate but does not propagate). A typical snapshot of the pattern is shown in Fig. 17, its corresponding template is Template 4. The ruffles along the right border of the pattern move up and right during the evolution – it is a periodic solution in time and space (See Fig. 20). Figure 18 shows the result of the chip measurement and its corresponding template, Template 5. The exact values of the chip templates are different from the template of the simulation, however the phenomenon can be reproduced quite well. The differences between the simulation and the chip measurements are due to the effect of the AD/DA converters in the supporting circuitry. It is important to note that it is not trivial to get similar attractors in simulator and in a real, noisy physical system.

$$\text{Template 4: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.55 & 0.3 & -0.55 \\ 0 & 0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.6$$

$$\text{Template 5: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0.4 & -0.9 \\ 0 & 0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2$$

Along the right border of the shadow-like structure, local oscillators operate which are coupled horizontally to the nearest neighbor and to the cell below them (see Fig. 19). Generally, there is no oscillation inside the structure, only along its border. Together these local oscillations form the main pattern, which looks like the cross-section of waves on the surface of the water (see Fig. 20).



Figure 17. The input & initial state and snapshot of the output pattern if the coupling is positive. (Simulated result)

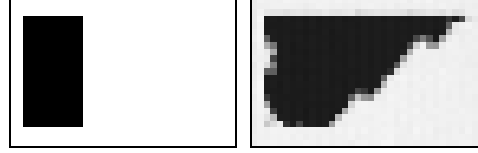


Figure 18. The input & initial state and snapshot of the output pattern if the coupling is positive. (Chip measurement)

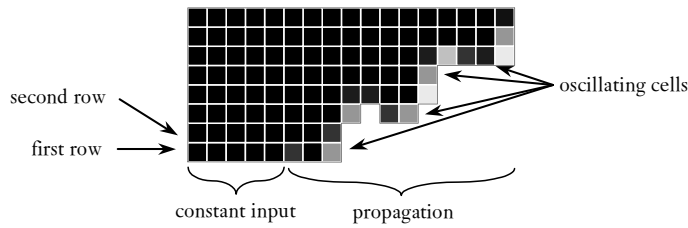


Figure 19. Structure of the pattern when p is greater than zero.

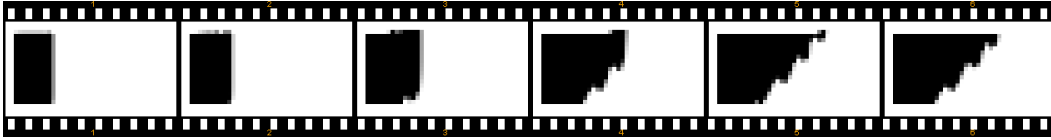


Figure 20. Time evolution of the pattern when p is positive. Snapshots of the state (simulated results, size: 41×23). The last snapshot shows the largest spatial extent of the generated pattern. There is no global propagation after that moment, only the ruffles travel along the right border of the pattern.

3.2.2 Negative vertical coupling ($r < 0$)

When the vertical coupling is negative a texture-like traveling pattern is formed. This pattern has a unique structure. It consists of propagating horizontal line segments which spread to the right (See Fig. 23). The spatial extent of the propagation to the right direction changes as we change the parameters but the main characteristics remain the same.

$$\text{Template 6: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & 0.4 & -1.1 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.4$$

$$\text{Template 7: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0.4 & -0.9 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2$$

Let us define the bottom row of the input pattern (black pixels) as the “first” row (See Fig. 24). The template produces a straight line (a shadow) for this row over a large range of parameter values. The cells in this row have really simple dynamics: they are stable (satu-

rated black pixels). The straight line serves as a constant driving for the cells in the next – “second” – row upward. The “second” row does not produce a traveling pattern. Instead, we find a few neighboring oscillating cells. The number of oscillating cells depends on the other nonzero elements of the template. The third row also gives rise to a periodic signal but with a different waveform and traveling pattern. Typical snapshots of the pattern are shown in Fig. 21 and Fig. 22 respectively for simulation and chip measurements. The corresponding generator templates are Template 6 and Template 7.

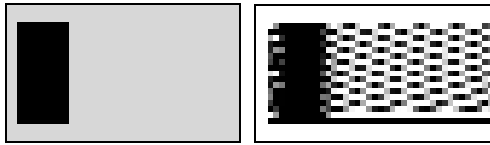


Figure 21. The input & initial state and snapshot of the output pattern if the vertical coupling is negative. (Simulated result)

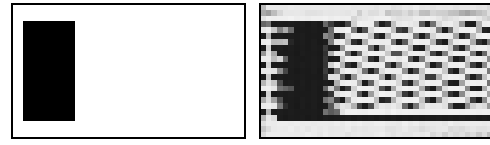


Figure 22. The input & initial state and snapshot of the output pattern if the vertical coupling is negative. (Chip measurement)

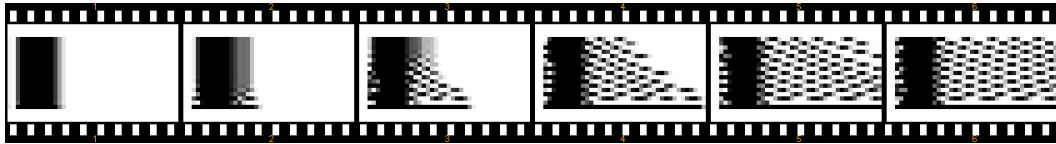


Figure 23. Time evolution of the pattern when r is negative. Snapshots of the state (simulated results, size: 41×23).

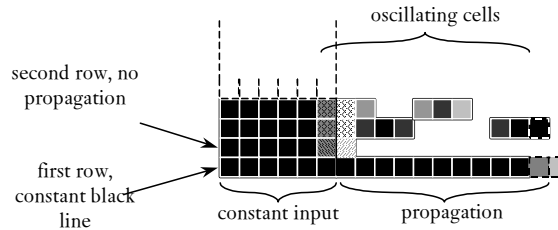


Figure 24. The structure of the traveling pattern.

3.3 The effect of the central template element

If we increase the central element (self feedback) the generated pattern becomes more and more irregular and it can become chaotic. Based on the simulations and measurements, it is possible to construct a partition of the r - p space. Fig. 25 shows the different dynamic regions of the system parameterized by r and p of Template 8.

$$\text{Template 8: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & -s \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z$$

3.3.1 Simulation

The r - p plane can be divided into *stable-periodic-chaotic* sub-regions (See Fig. 25).

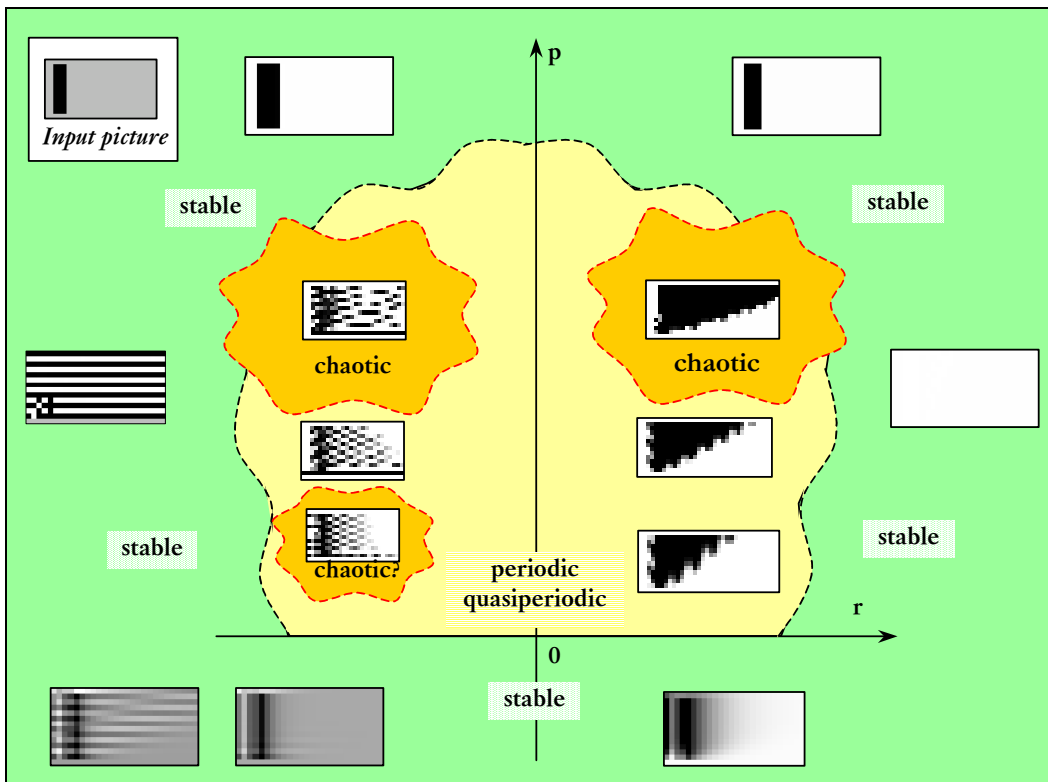


Figure 25. Partitioning of the r - p parameter space. The input & initial state are shown in the upper left corner. It is a three-pixel wide bar. The pictures in the different regions show few typical snapshots of outputs belonging to that region. The arrangement and size of the different regions gives only qualitative information.

Stable region

Around the periodic and chaotic region is a stable region with various stable patterns. When p is high the effect of the input becomes dominant and therefore the output is almost the same as the input. When p is low (negative) the character of the system is diffusive-dissipative. Between the two extreme values of p is a region where the effect of the input is less significant. Therefore the patterns are dominantly one-dimensional or there is no pattern at all.

Periodic region

The patterns propagate periodically and have the form of *solid wave-like* and *texture*, as it is described previously in Section 3.2.

Chaotic region

If p is large enough, the system can become chaotic. However, I found that chaotic behavior may occur at smaller p , when r is less than zero. (See Fig. 25).

Positive coupling ($r > 0$)

This section contains results of simulations when r is greater than zero. Observe the transition from the simple to the more complex dynamics. Fig. 26 shows the zoomed structure of the pattern. Figures 27 - 29 show the snapshot of the generated pattern, the time evolution of one sampled state variable, the power spectrum of that variable and the trajectory of the sampled cell and its neighbor cell. The captions contain the actual value of parameter p ($p = [0.5, 0.87], r = 0.3$). The generator template for the figures is Template 9.

$$\text{Template 9: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & p & -1.1 \\ 0 & 0.3 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.1$$

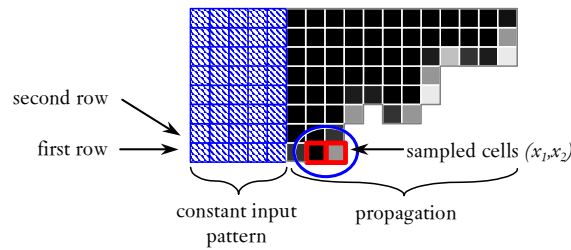


Figure 26. In the first row, to the right from the edge of the black part of the constant input (denoted by blue striped boxes) the second and third cells were sampled as shown.

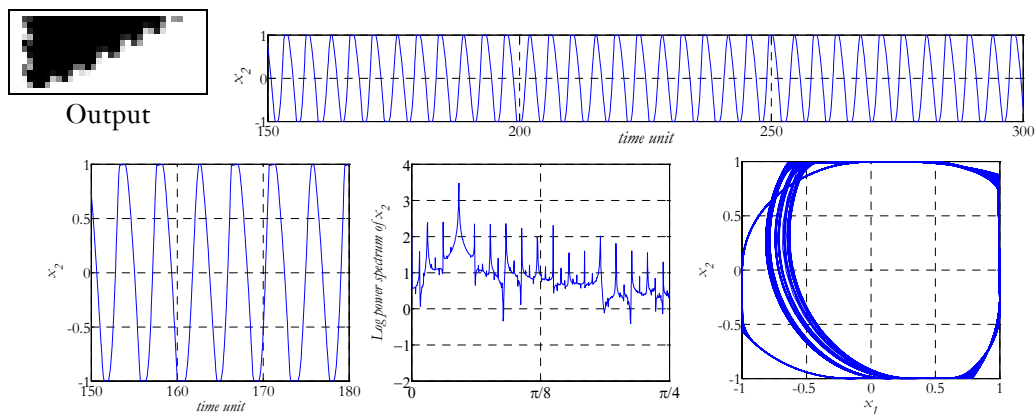


Figure 27. Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.5$).

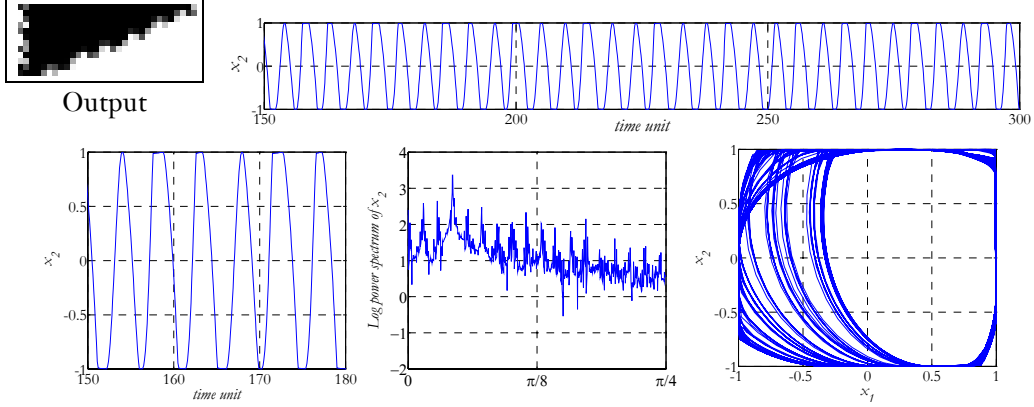


Figure 28. Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($\rho=0.6$).

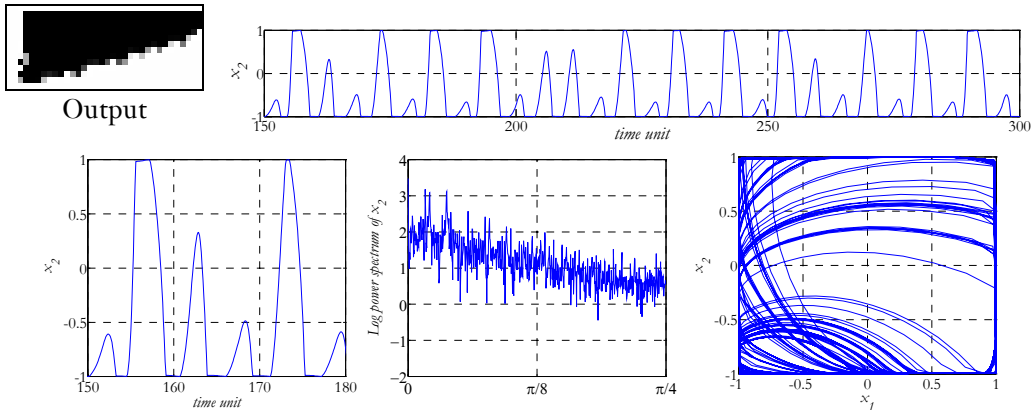


Figure 29. Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($\rho=0.87$).

Negative coupling ($r < 0$)

When r is less than zero, a texture-like traveling pattern is formed. Fig. 30 shows the zoomed structure of the pattern. Observe the transition from the simpler to the more complex dynamics in Figures 31 - 33. The figure captions contain the actual value of parameter p ($p=[0.2, 0.7]$, $r = -0.3$). See the generator template, Template 10.

$$\text{Template 10: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & p & -1.1 \\ 0 & -0.3 & 0 \end{bmatrix} \quad \mathbf{B} = [1.2] \quad z = 0.1$$

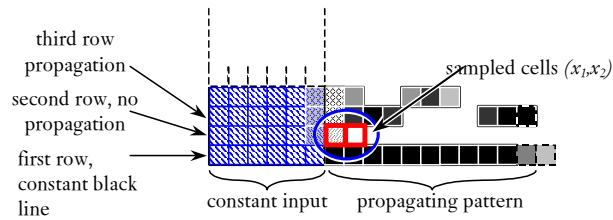


Figure 30. In the second row, right from the edge of the black part of the constant input (denoted by blue striped boxes) the first and second cells were sampled as shown.

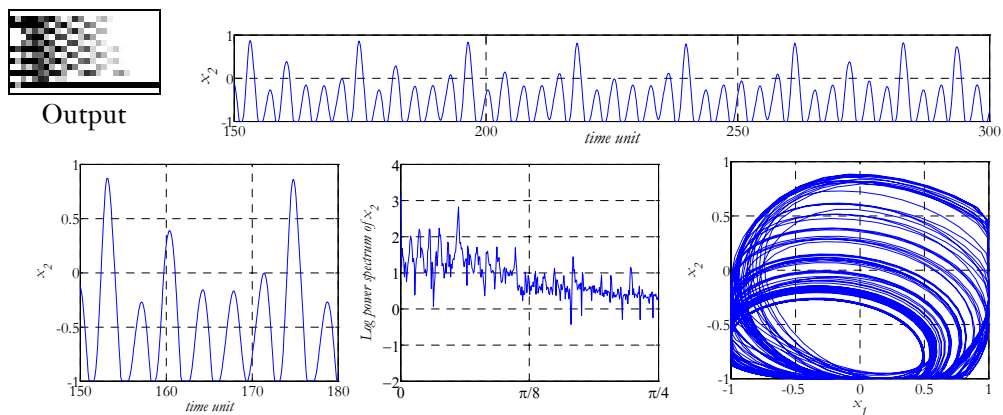


Figure 31. Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.2$).

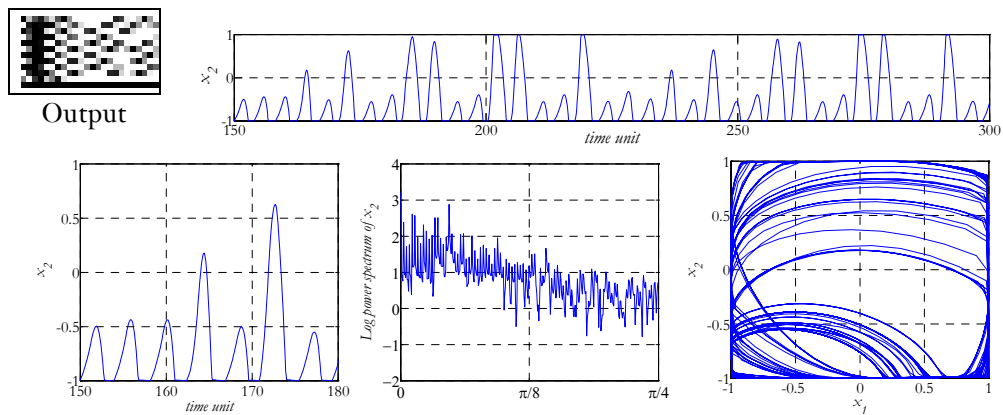


Figure 32. Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.6$).

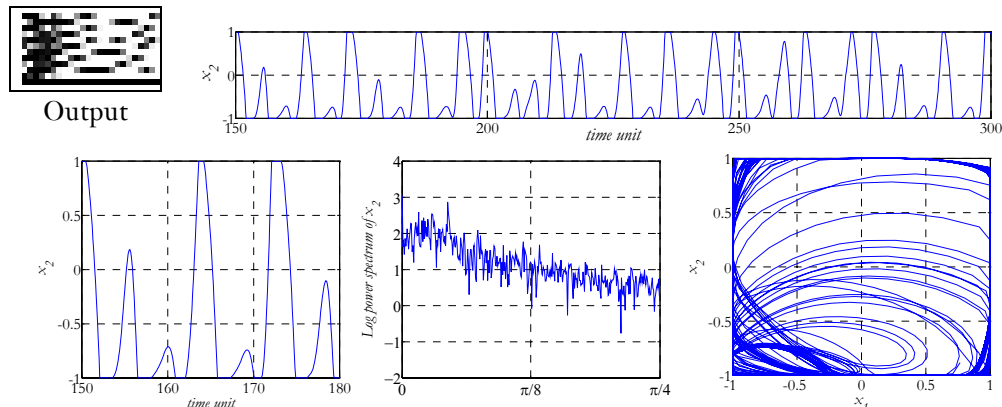


Figure 33. Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.7$).

3.3.2 Programmed chip measurement

The following subsections contain programmed chip measurements at different values of p for positive and negative values of r . While the measured waveforms do not coincide completely with that of the simulation, the qualitative details of the phenomenon are the same. In the figures the trajectories are virtually scaled and the null points are virtually shifted compared to the simulation results. This is due to the AD/DA supporting circuitry of the chip.

When there is no stable equilibrium, such as in the case of periodic or chaotic steady state behaviour, it is not trivial to find the equivalent settings for the chip to get the same result as that of the simulation. Since the chip is a real, physical, noisy system with high complexity and interconnection, the model that we simulate is inevitably different from the model of the chip. In spite of this it is still possible to get appropriate results, which show the robustness of the phenomenon.

Positive coupling ($r > 0$)

Cells from the first row were sampled (See Fig. 26). When p is small, the power spectrum contains dominant peaks according to the periodic signal. Later, when p is higher the peaks disappear or significantly decrease. Figures 34 - 36 show the measured time series, power spectrum and the trajectory of the two sampled cells. The sampling position is shown in Fig. 26. The generator template for the figures is Template 11. The captions contain the actual value of parameter p ($p = [0.51, 0.85]$, $r = 0.24$).

$$\text{Template 11: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & p & -1 \\ 0 & 0.24 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.6 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2.1$$

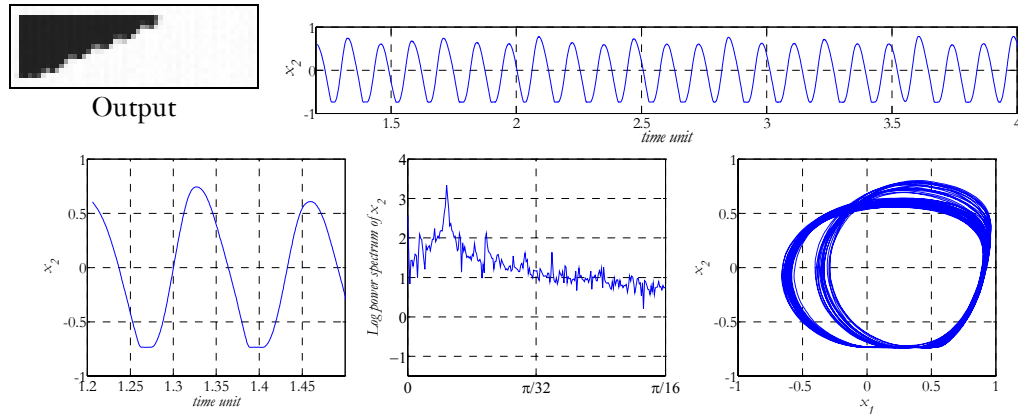


Figure 34. Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the first row ($p=0.51$).

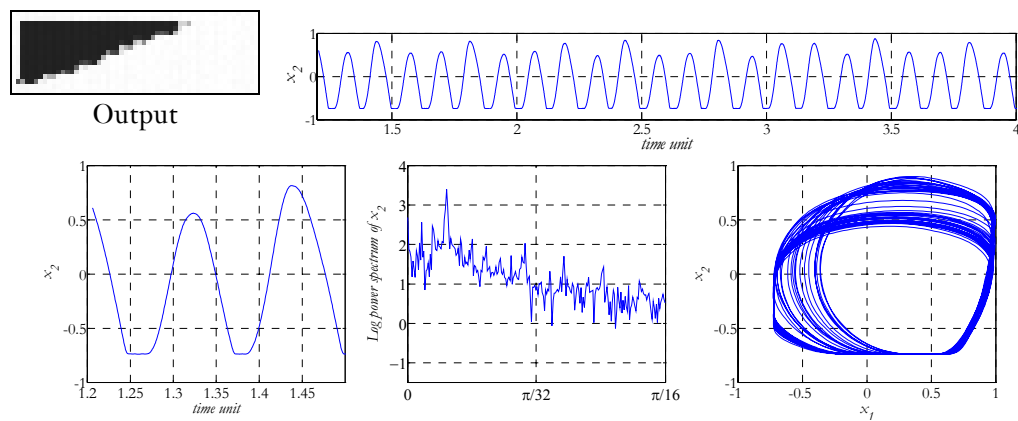
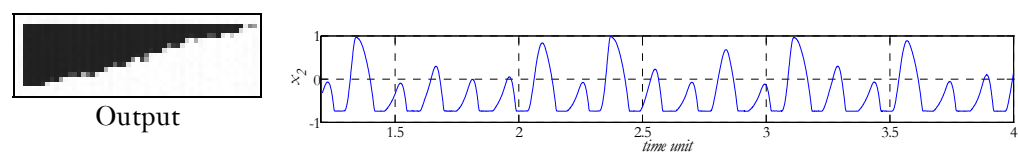


Figure 35. Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the first row ($p=0.61$).



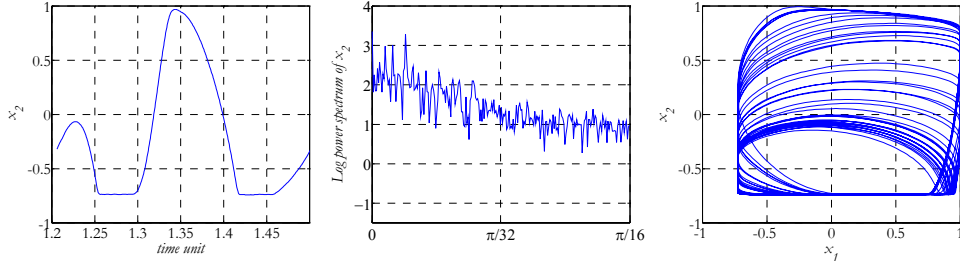


Figure 36. Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the first row ($p=0.85$).

Negative coupling ($r < 0$)

In case of negative coupling we experiment with a similar phenomenon to that of the simulation. Figures 37 - 39 show the measured time series, power spectrum and the trajectory of the two sampled cells. Fig. 30 shows the sampling position. The generator template for the figures is Template 12. The captions contain the actual value of parameter p ($p = [0.42, 0.68]$, $r = -0.4$).

$$\text{Template 12: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.2 & p & -1.2 \\ 0 & -0.4 & 0 \end{bmatrix} \quad \mathbf{B} = [1.4] \quad z = 2$$

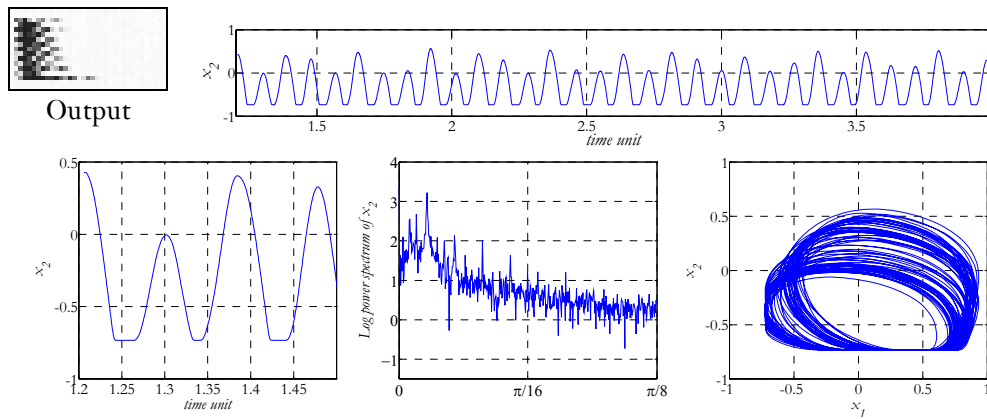
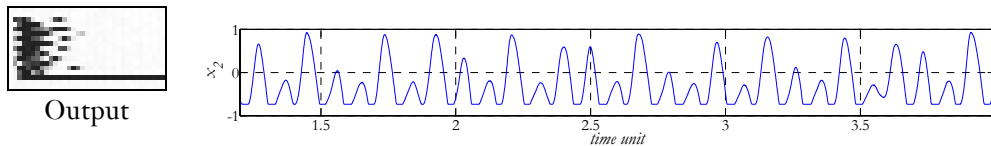


Figure 37. Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the second row ($p=0.42$).



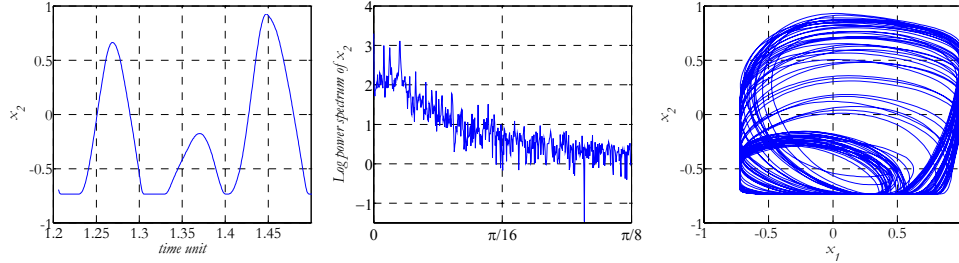


Figure 38. Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the second row ($p=0.6$).

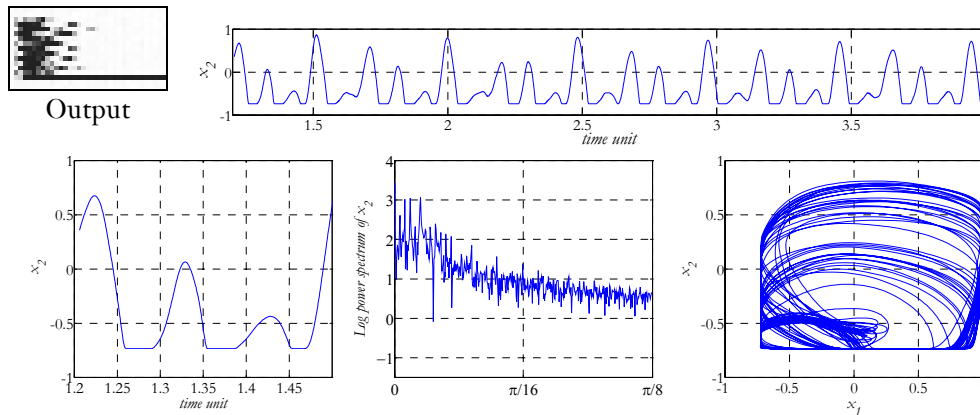


Figure 39. Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the second row ($p=0.68$).

3.3.3 Spatio-temporal signatures

As it can be seen from the results, there is a correlation between the time evolution of the selected cells of the CNN and the dynamic behaviour of the whole array.

Spatio-temporal Signature. A still image that is the snapshot of the output, a descriptor for the dynamic state of the array.

This image reflects the temporal history of the dynamics in space due to the propagating effect; therefore, it can be suitable for characterizing the system without cell data measurements. Observe that, due to the complex dynamics of the patterns, it is difficult to find good characteristic 2D snapshots. Table 7 shows the most characteristic signatures for the state of the CNN.

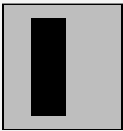
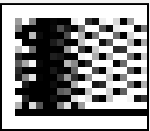
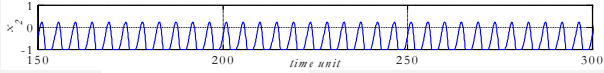
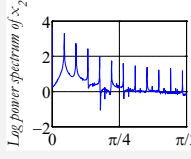
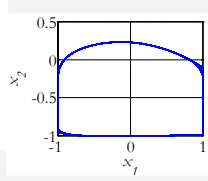
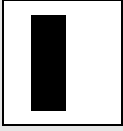

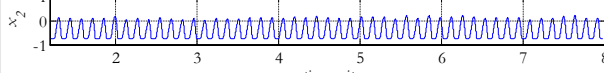
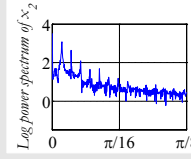
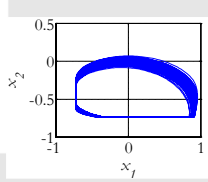
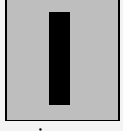

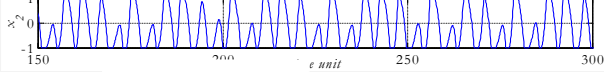
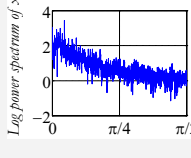
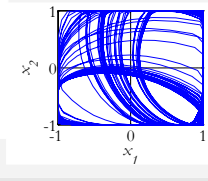
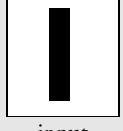
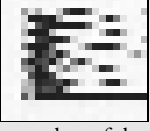
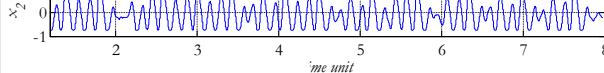
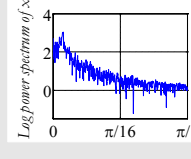
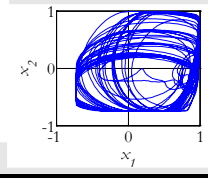
Periodic	Simulation	 input	 snapshot of the output	  
	chip measurement	 input	 snapshot of the output	  
Chaotic	Simulation	 input	 snapshot of the output	  
	chip measurement	 input	 snapshot of the output	  

Table 6. Different dynamical behaviours with the same template but with different input and initial state. The time evolution of one cell from the second row, logarithm of power spectrum of the same cell and the 2D trajectory of the same cell and the neighbor are shown.

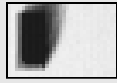




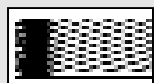
Stable		Chaotic		Periodic	
	$r > 0, p < 0$		$r > 0, p \gg 0$		$r > 0, p > 0$
	$r < 0, p < 0$		$r < 0, p \gg 0$		$r < 0, p > 0$

Table 7. Spatio-temporal signatures for different values of r and p .

3.4 The effect of the constant input and initial state

An inherent property of the chaotic systems is the extreme sensitivity to the initial condition. In this section some results relating to this aspect are presented.

3.4.1 Periodic-chaotic transition

Table 6 shows the effect of the different input patterns in the case of simulation and chip measurements. The applied templates are the same for the two different inputs, i.e. the different behaviour of the system is due to the difference of the input pattern.

The first input is a five pixel wide black vertical bar and the other input is a three pixel wide black vertical bar. The input and the initial state are the same. Template 13 and 14 show the templates for simulation and chip measurements respectively. The gray background color of the input image for the simulation is to mimic the property of the chip, namely, its gray level shifting property. Using these level-shifted images we get similar results to that of the chip.

Template 13:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & 0.7 & -1.1 \\ 0 & -0.3 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.1$$

Template 14:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0.5 & -0.9 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2.1$$

The result shows that if the input is the five pixel wide bar then the transient of the cell is periodic. But if the input is a three pixel wide bar, the transient – and the traveling pattern too – is chaotic. The reason for the difference is that the local oscillators along the left and right border of the bar can influence each other. This can happen only if there is no stable (constant, saturated black) vertical column of cells along the center of the bar. If the input is the three pixel wide bar there is no column of saturated (+1) stable cells in the bar. In the other case there is at least a one pixel wide column of saturated stable cells horizontally. Thus, the oscillators along the left and right border of the bar are uncoupled. The cells were sampled in the second row.

Fig. 40 shows the r - p diagram in that case when the input and initial state are changed to the five pixel wide vertical bar. No chaotic behaviour has been found up to now. However, it cannot be excluded that it is possible to find a certain parameter setting at which the sys-

tem produces chaos. A necessary condition for this seems to be that the stable saturated cells along the center of the bar become unstable.

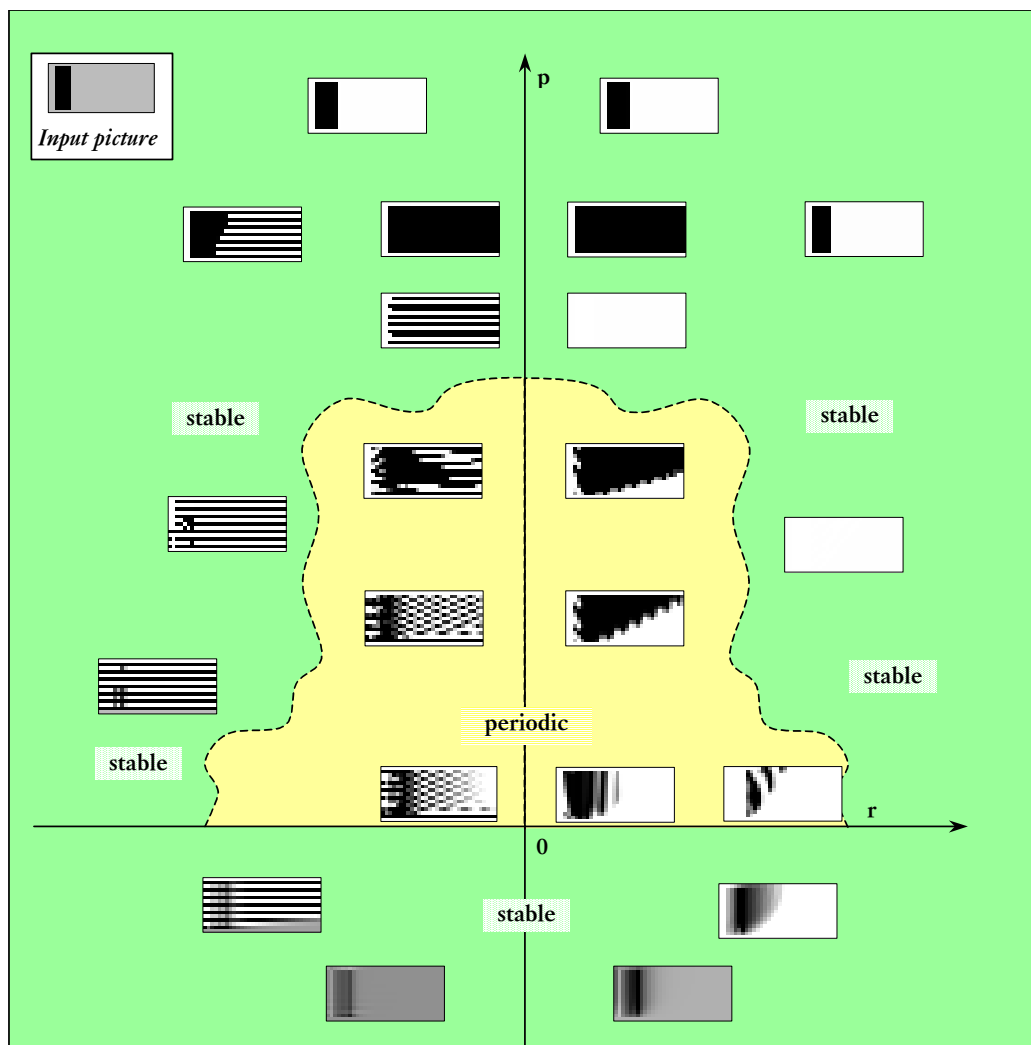


Figure 40. Partitioning of the r - p parameter space when the input is a five pixels wide bar. The input picture is shown in the upper left corner. The pictures in the different regions show few typical snapshots of the output patterns belonging to that region. With this input no chaotic behaviour has been found up to now. The arrangement and size of the different regions give only qualitative information.

3.4.2 Stable-periodic transition

Fig. 41 illustrates that a single pixel perturbation can alter the general dynamic behaviour of the system. The only difference between the three inputs, which are also the initial state, is that a single pixel is changed from black to gray (and from gray to black in the other case) in the middle of the right vertical edge of the bar. In the first case there is no propaga-

tion and no pattern. However, the second and third input produces a periodic pattern (See Fig. 41). The boundary condition is periodic (torus-like left-right and bottom-up connections, this can also be programmed on the chip). The generator template is Template 15.

$$\text{Template 15: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0.5 & -1 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0$$

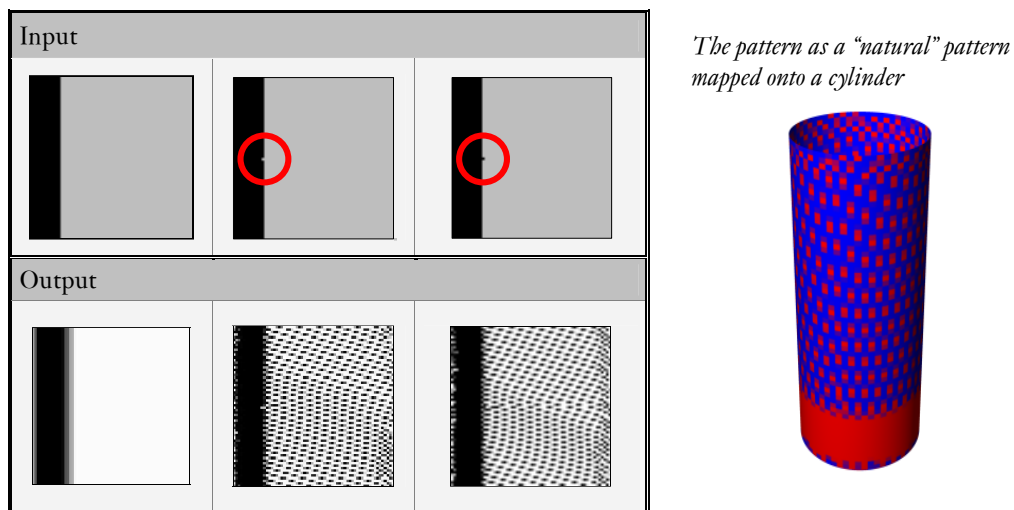


Figure 41. Effect of one pixel perturbation. The first row contains three different inputs. The first one is a vertical bar. The second one is the same except one pixel: a pixel is *clipped off* from the middle of the right border of the pattern. In the third one a pixel is *added* to the middle of the right border of the pattern. The circles denote the location of the difference. The second row contains the corresponding snapshots of the output patterns. In the first column the solution is equilibrium. The second and third column shows periodic (in space and time) solution.

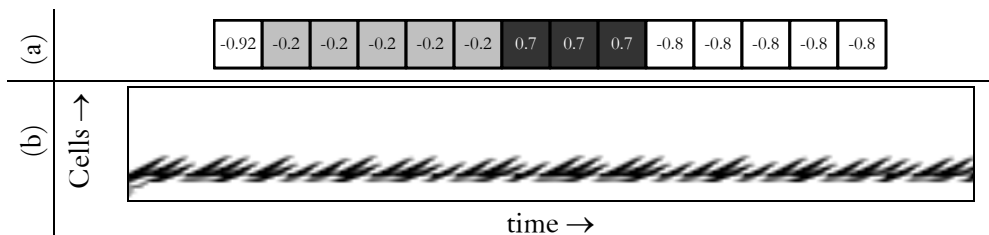


Figure 42. (a) Input and initial state for a 1D chaotic CNN that consists of 14 cells. (b) The generated 1d pattern as a function of time.

3.5 1D chaos

Based on measurements and simulations we can construct probably the simplest template (e.g. Template 16) for a 1D CNN that can exhibit chaos with an appropriate input shown in Fig 42. The minimal number of cells that is necessary to produce chaos appears to be four according to the results of Chapter 4.

The general structure of the templates presented in this thesis shows that a CNN cell in the array does not depend on the cells above it. In the cases above (Templates 13,14) I measured and simulated chaotic signals in the row which has a constant valued row below it. Therefore, if we integrate the effect of the constant valued row into the constant input it is possible to construct a 1D CNN (template, initial state and input) for which the system is chaotic (see Fig 42 and Fig 43). All boundary cells were set to zero.

$$\text{Template 16: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & 0.39 & -1.1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.2$$

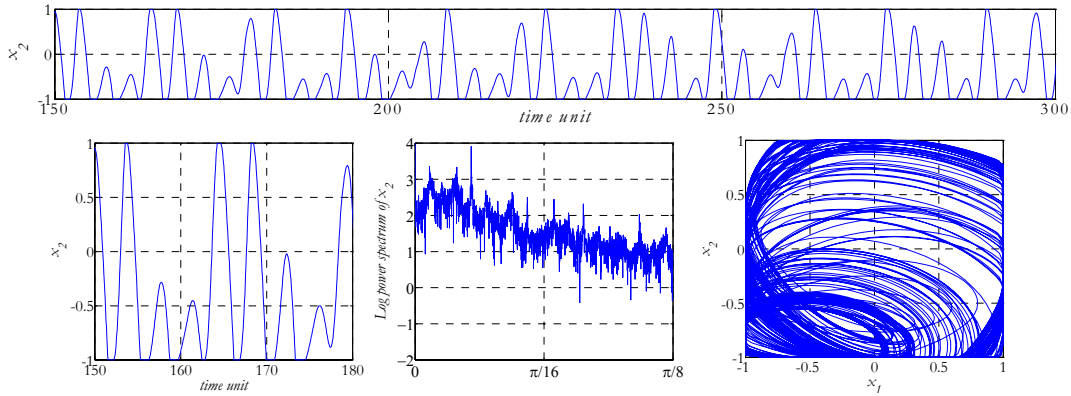


Figure 43. The time evolution, the power spectra of one cell and the trajectory of two neighboring cells.

3.6 Additional traveling pattern examples

This section shows some patterns which are generated with more complicated templates. The basic structure is the same i.e. the antisymmetry is preserved but some nonzero couplings are added. The corresponding templates matrices are shown below the figures.

3.6.1 Wave shadow

This special shadow operator produces different patterns depending on the central element p . The main structure remains the same at different values of p , but the border of the shadow looks like traveling wrinkles. The phenomenon related to the change of p is similar to that of described in Section 3.3. Template 17 and 18 are the generator templates of the patterns in Table 8 for simulation and for chip measurements respectively.

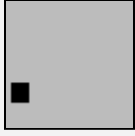



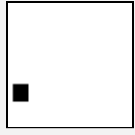



Simulation					
Snapshots of the output					
	Input	p	0.05	0.3	0.7
Chip measurement					
Snapshots of the output					
	Input	p	0.15	0.41	0.94

Table 8. Snapshots of the simulation and of the chip measurements of the complex wave template. Input and initial state are the same. The gray background of the input for the simulation is to mimic the level shift property of the chip.

Template 17:

$$A = \begin{bmatrix} 0 & 0 & -0.6 \\ 1 & P & -1 \\ 0.8 & 0 & -0.1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.9$$

Template 18:

$$A = \begin{bmatrix} 0 & 0 & -0.6 \\ 1 & P & -1 \\ 0.8 & 0 & -0.2 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 4.2$$

With a different template setting (Template 19) a characteristic trajectory is measured on the chip (See Fig. 44 and Fig. 45).

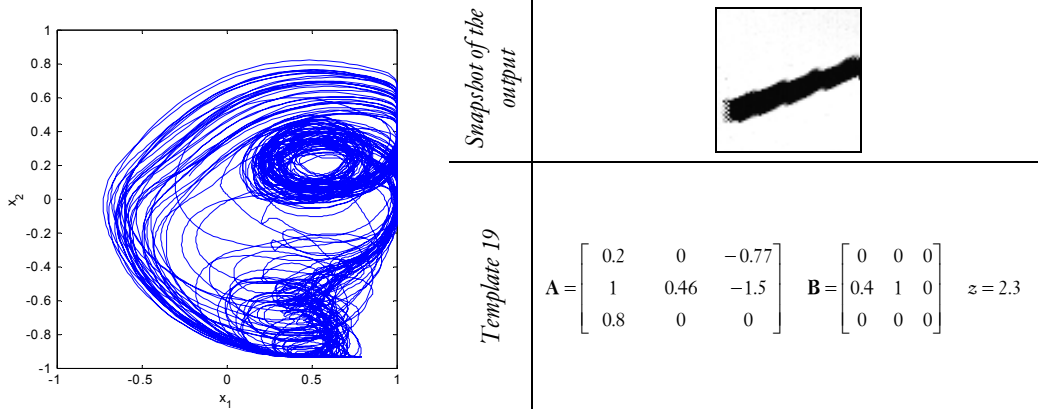


Figure 44. The measured trajectory of the attractor and a snapshot of the traveling pattern. (chip measurement).

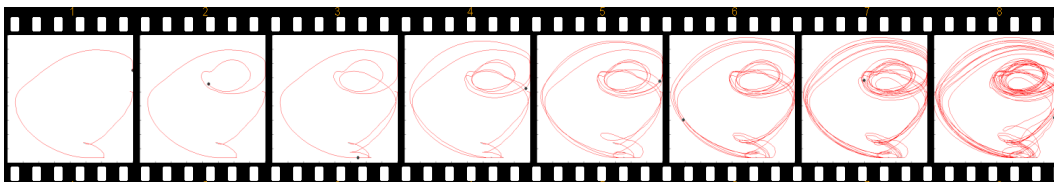


Figure 45. The measured trajectory of the attractor. Snapshots of trajectory of two out of 4096 state variables of the CNN array (*chip measurement*).

3.6.2 “Four pixels” examples

The patterns in this subsection were simulated using the full-range CNN model of eq. (3.1). The next examples show different patterns, all of which were generated from the same input. The “seed” of the patterns is a four-pixel wide horizontal line section. The initial state was the same as the input that is shown in Fig. 46.

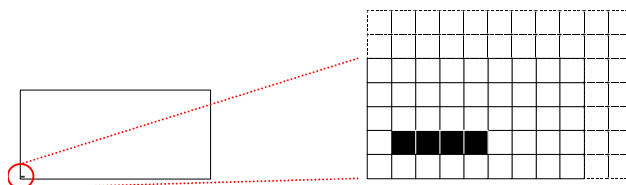


Figure 46. Input and initial state for the traveling patterns.

Bird

This example shows a traveling pattern which makes an impression of distant flying birds. (See Fig. 47 and Template 20)

$$\text{Template 20: } \mathbf{A} = \begin{bmatrix} 0 & 0 & -0.8 \\ 1.1 & 1 & -1 \\ 0.7 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.9$$



Figure 47. Three snapshots of the propagation of the “bird” pattern.

Eruption

This pattern “erupts” periodically during the propagation. It resembles fluid coming out of a pipe. (See Fig. 48 and Template 21)

$$\text{Template 21: } \mathbf{A} = \begin{bmatrix} 0 & 0 & -1 \\ 1.1 & 1.3 & -1 \\ 1.1 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 1.4$$

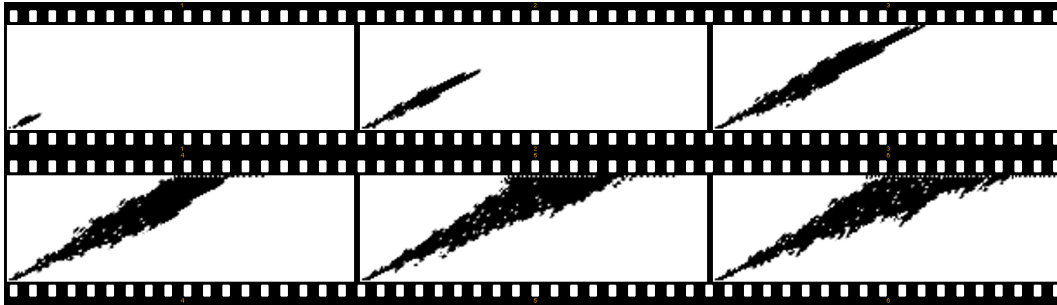


Figure 48. Six snapshots of the propagation of the "eruption" pattern.

3.7 Simulation time vs. real-time measurements

A sophisticated simulation of chaotic systems takes several minutes (or even hours), especially when the dimension of the system is high. Using the programmable ACE4K test bed it is possible to speed up the analysis process of the chaotic system by at least four orders of magnitude. Table 9 shows the comparison of a typical simulation and the real time chip measurement.

Simulation (500Mhz PC)	Chip measurement (with sampling)	Chip measurement (without sampling)
25 min = 1500 sec	200 ms	5 ms

Table 9. Comparison of simulation and chip measurement time.

4. *Complex dynamics in 1D CNN*

Second Thesis

Chaotic dynamics, coexistence of attractors in 1D CNN

I uncovered novel complex dynamics in 1D CNNs with a sign-antisymmetric template class. I investigated the effect of the boundary condition, dimension of the system and the initial state. I found that boundary condition behaves as bifurcation parameter. I found coexistence of attractors depending on the initial state.

II/a. Effect of boundary conditions in 1D CNN

I showed that the dynamic behavior of a 1D CNN with a sign-antisymmetric template class depends on the boundary conditions. I presented examples that demonstrate the rich dynamics of the CNN system such as stable, periodic, quasi-periodic and chaotic dynamics.

II/b. Effect of dimension and initial state, coexistence of attractors

I found that the steady state dynamics depends on the initial state: attractors coexist simultaneously in the same system. I showed that hyperchaotic behavior can occur in a CNN that consist of more than five cells. I demonstrated that the parity of the size of the 1D system drastically changes the dynamics.

FROM the point of view of dynamics the CNN can be divided into two classes: stable CNN in which each trajectory converges to a stable equilibrium point and unstable CNNs which have at least one attractor that is not a stable equilibrium point. Up to now CNNs with this property have been the main concern of research because of the requirement of image processing applications in which usually the steady state of the system yields the desired result. Complete stability of CNNs was proved for symmetric templates in [46] and [109]; for positive cell linking templates in [106]. It was shown in [109] that CNNs with strictly sign-symmetric and acyclic templates are completely stable. Gilli in [110] gave sufficient condition for nonsymmetric CNNs. In [100] Zou investigated 1D CNNs with an opposite sign template. He determined a subset of the parameter range where the CNN is completely stable. He conjectured that opposite sign CNNs with $s < p - 1 < 2s$ are completely stable. De Sandre refined the results of Zou in [111]. Thiran et. al. also studied general 1D templates and introduced the concept of *local diffusion* [101] and *global propagation* [102] and proved some theorems on the number of stable equilibria of general 1D CNNs.

The unstable class of CNNs received much less attention. However, recently CNNs with opposite sign templates were reported in [106] that showed oscillating behaviour. In [100] Zou analyzed an opposite sign template class $[s \ p \ -s]$ in detail and presented an example of periodic limit cycle. Further examples of CNNs exhibiting limit cycles are shown in [102-115,158,159]. Complex dynamic behaviour has been observed with different CNN models:

- chaotic attractor in a nonautonomous two-cell system [116]
- autonomous CNN with space variant and space invariant templates of three cells [103] organized into 2D grid.
- autonomous fully connected CNN composed of three cells [160].
- delayed CNN [117]
- state controlled CNN [105]
- autonomous CNN with space invariant templates composed of arbitrary cells [104][93]
- One- and two-dimensional CNNs with constant input [118]

4.1 Effect of boundary condition

The former works with the exception of some general results concerning CNN stability concentrated mainly on the effect of the template parameters. In [119] Thiran analysed the effect of the boundary condition on the dynamics of the CNN and established three classes: always stable, always unstable, conditionally stable or unstable depending on the boundary conditions. He presented an example of a CNN with opposite sign template of

form $[s \ p \ -s]$ that is stable if the boundary condition is ± 1 and unstable with zero boundary condition.

From the work of Zou [100] it is known that 1D CNNs without input, with $p-1 < s$ possess no equilibrium point. He proposed to check whether complex dynamics is found. This is a realistic expectation since the CNN is a high dimensional coupled nonlinear system.

I investigated this possibility inspired by the work done with 2D spatio-temporal chaos that was described formerly. I discovered surprisingly rich dynamics and bifurcation process. Indeed, constant boundary conditions behave as bifurcation parameters from the point of view of global dynamics of the system. They determine the equilibrium point patterns that consist of permutations of ± 1 . They are responsible for the transition from stability to instability. They also determine the type of instability such as periodic, quasi-periodic, and chaotic.

For a better understanding of dynamics I repeat here four theorems that help to localize the possible interesting parameter region.

Let us consider the following 1D version of (2.4):

$$\dot{x}_i = -x_i + wf(x_{i-2}) + sf(x_{i-1}) + pf(x_i) + rf(x_{i+1}) + vf(x_{i+2}) \quad (4.1)$$

where f is defined as:

$$f(x) = \frac{1}{2}(|x+1| - |x-1|)$$

This CNN has a general 5×1 template: $\mathbf{A} = [w \ s \ p \ r \ v]$. The effect of \mathbf{B} template is integrated into β .

Theorem 1 [Gilli & Petrás, 120]: If the template elements of equations (4.1) satisfy the constraints $wv \geq 0$ and $sr \geq 0$, then the corresponding CNN is completely stable (i.e. all trajectories converge towards an equilibrium point) for any external constant inputs and boundary conditions.

Proof: it is derived from Theorem 1 and Theorem 2 of [110].

Consider now a 3×1 system:

$$\dot{x}_i = -x_i + sf(x_{i-1}) + pf(x_i) + rf(x_{i+1}) + \beta \quad (4.2)$$

Theorem 2 [Setti] The number $S(N)$ of stable equilibria of CNN (4.2) of $N \geq 3$ cells, satisfying $p-1 < |s-r|$, $p > 1$, $\beta = 0$ and with zero boundary condition, is equal to

$$\begin{aligned} 2N, & \quad \text{if } \max\{|r|, |s|\} < p-1 < |s-r| \\ 2, & \quad \text{if } \min\{|r|, |s|\} < p-1 < \max\{|r|, |s|\} \\ 2, & \quad \text{if } p-1 < \max\{\min(r,s), \min(-r,-s)\} \\ 0, & \quad \text{if } p-1 < \max\{\min(-r,s), \min(r,-s)\} \end{aligned}$$

Proof: See [102].

Theorem 3 [Zou] The CNN described in (4.2) with $p > 1$, $s > 0$ has no stable equilibrium point, if

$$\beta = 0 \text{ and } s > p - 1$$

Proof: See [100].

Theorem 4 [Zou] The system defined by (4.2) with $p > 1$, $s > 0$ is completely stable if

$$\beta = 0 \text{ and } s < (p - 1) / 2$$

Proof: See [100].

These theorems restrict the possible parameter space of complex dynamics. Without loss of generality suppose that $s > r$. Summarizing the theorems yields that the following should be fulfilled so as the system does not have any stable equilibrium points:

$$s \cdot r < 0 \quad (\text{Theorem 1})$$

$$p - 1 < |s - r|, p - 1 < \min\{s, -r\} \quad (\text{Theorem 2})$$

$$\text{if } s = -r, p - 1 < s \quad (\text{Theorem 3})$$

I show that if the template parameters fulfill $s > p - 1$ derived from these inequalities the boundary conditions behave as bifurcation parameters.

For the sake of simplicity I chose a one-dimensional opposite sign template class as a case study:

$$A = [s \ p \ -s].$$

For practical and theoretical reasons the dimension of the system under study is from four by one to eight by one.

Table 10 shows the possible equilibrium point patterns in a compact, regular expression form as a function of p , s and boundary conditions. They are composed of sequences of $\{+1, -1\}$. The string expression $\{a, b\}^0$ represents the null string, whereas the expression $\{a, b\}^n$ denotes a string obtained by repeating n times the symbols a and b , e.g. $\{a, b\}^3 = a, b, a, b, a, b$. The expressions were obtained using Deterministic Finite Automaton (DFA).

The possible stable equilibrium patterns of a CNN are represented in such a way that a stable pattern constitutes one node of a graph (see Fig. 49). The stability of a pattern around a cell is determined by the actual state of that cell and by its one or two neighbors. A cell has one neighbor, if it is a left- or rightmost cell and two neighbors if it is an inner cell. The nodes are organized into three columns. In the first column there are the stable patterns for the leftmost cells. On their left is a constant cell state of which is always equal to the left boundary condition. In the second (inner) column are the stable patterns for the inner cells. The third column contains the stable patterns for the rightmost cells. Their right neighbor is a constant cell with the right boundary condition.

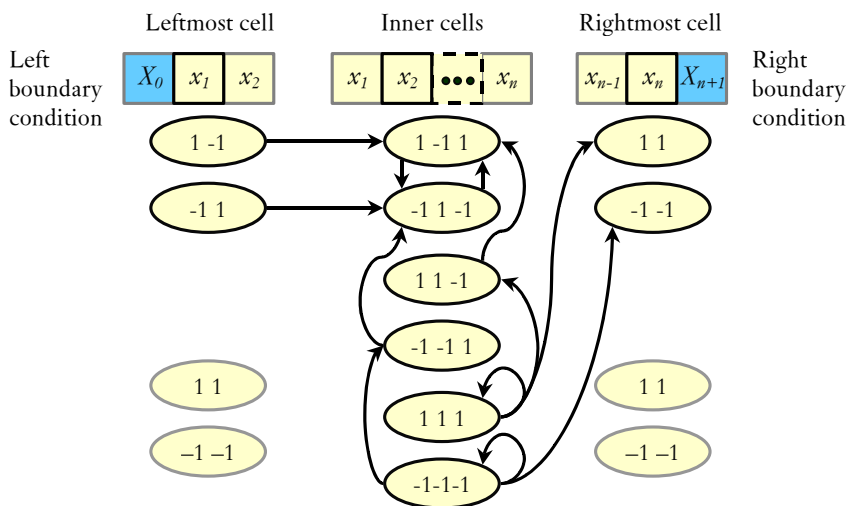


Figure 49. The DFA of the stable equilibrium patterns. With $X_0=0$, $X_{N+1}=0$ there is no stable equilibrium point of the CNN characterized by $A=[s \ p \ -s]$, $p-1 < s$.

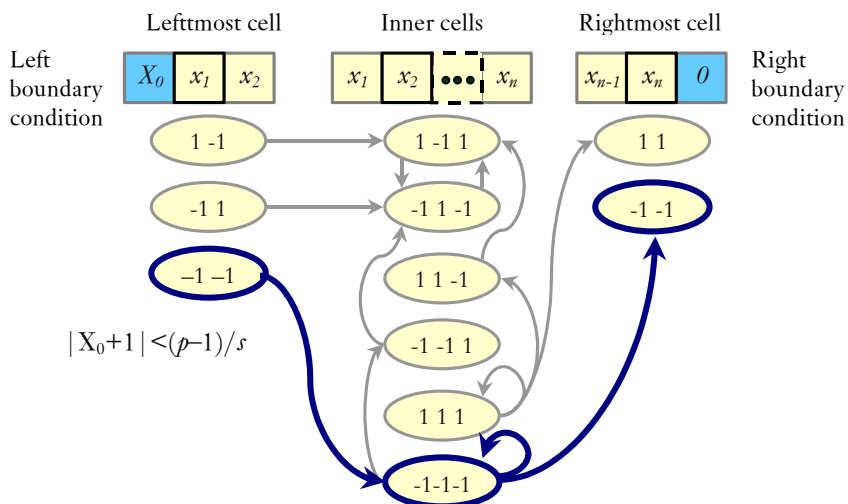


Figure 50. The DFA of the stable equilibrium patterns. In the region $\{|X_{N+1}| < 1-(p-1)/s, |X_0+1| < (p-1)/s\}$ is one stable equilibrium pattern: $-1, \{-1\}^n, -1$. Thus a path exists from the left to the right in the DFA graph (bold line). $A=[s \ p \ -s]$, $p-1 < s$.

<i>Equilibrium point patterns</i>	$X_{N+1} < \frac{-(p-1)+s}{s}$	$ X_{N+1} < \frac{p-1}{s}$	$ X_{N+1} < \frac{-(p-1)+s}{s}$	$ X_{N+1}-1 < \frac{p-1}{s}$	$X_{N+1} > \frac{p-1+s}{s}$
$X_0 < \frac{-(p-1)+s}{s}$	$-1, \{-1\}^n, +1, -1, \{-1, -1\}^m, +1$ $-1, \{-1\}^n, +1$	$-1, \{-1\}^n, +1, -1, \{-1, -1\}^m, +1$ $-1, \{-1\}^n, +1$	$-1, \{-1\}^n, +1, -1, \{-1, -1\}^m, +1$ $-1, \{-1\}^n, +1$	$-1, \{-1\}^n, +1, \{-1, -1\}^m, -1$ $-1, \{-1\}^n, -1$	$-1, \{-1\}^n, +1, \{-1, -1\}^m, -1$ $-1, \{-1\}^n, -1$
$ X_0+1 < \frac{p-1}{s}$	$-1, \{-1\}^n, +1, -1, \{-1, -1\}^m, +1$ $+1, -1, \{-1, -1\}^n, +1$ $-1, \{-1\}^n, +1$	$-1, \{-1\}^n, +1, -1, \{-1, -1\}^m, -1$ $+1, -1, +1, \{-1, -1\}^n, -1$ $-1, \{-1\}^n, -1$	$-1, \{-1\}^n, +1, -1, \{-1, -1\}^m, -1$ $+1, -1, +1, \{-1, -1\}^n, -1$ $-1, \{-1\}^n, -1$	$+1, -1, +1, \{-1, -1\}^m, -1$ $-1, +1, \{-1, -1\}^n, -1$	$+1, -1, +1, \{-1, -1\}^n, -1$ $-1, +1, \{-1, -1\}^n, -1$
$ X_0 < \frac{-(p-1)+s}{s}$	$-1, +1, -1, \{-1, -1\}^n, +1$ $+1, -1, \{-1, -1\}^n, +1$	$-1, +1, -1, \{-1, -1\}^n, +1$ $+1, -1, \{-1, -1\}^n, +1$	<i>No equilibrium points</i>	$-1, -1, +1, \{-1, -1\}^n, -1$ $-1, \{-1\}^n, -1$	$-1, \{-1\}^n, +1, \{-1, -1\}^m, -1$ $-1, \{-1\}^n, -1$
$ X_0-1 < \frac{p-1}{s}$	$+1, \{+1\}^n, -1, \{+1, -1\}^m, +1$ $-1, +1, -1, \{+1, -1\}^n, +1$ $+1, \{+1\}^n, +1$	$+1, \{+1\}^n, -1, \{+1, -1\}^m, +1$ $-1, +1, -1, \{+1, -1\}^n, +1$ $+1, \{+1\}^n, +1$	$+1, \{+1\}^n, +1$	$+1, \{+1\}^n, -1, +1, \{-1, -1\}^m, -1$ $-1, +1, -1, \{-1, -1\}^n, -1$ $+1, \{+1\}^n, -1$ $+1, \{+1\}^n, +1$	$+1, \{+1\}^n, -1, +1, \{-1, -1\}^m, -1$ $-1, +1, -1, \{-1, -1\}^n, -1$ $+1, \{+1\}^n, -1$
$X_0 > \frac{p-1+s}{s}$	$+1, \{+1\}^n, -1, \{+1, -1\}^m, +1$ $+1, \{+1\}^n, +1$	$+1, \{+1\}^n, -1, \{+1, -1\}^m, +1$ $+1, \{+1\}^n, +1$	$+1, \{+1\}^n, +1$	$+1, \{+1\}^n, -1, +1, \{-1, -1\}^m, -1$ $+1, \{+1\}^n, -1$ $+1, \{+1\}^n, +1$	$+1, \{+1\}^n, -1, +1, \{-1, -1\}^m, -1$ $+1, \{+1\}^n, -1$

Table 10. The Table reports a compact expression for all the possible stable equilibrium point patterns in a CNN composed by N cells and described by a 1D template $[s, p, -s]$ ($p-1 < s$ and $s > 0$) with boundary conditions X_0 and X_{N+1} . The patterns have been determined by exploiting a rigorous technique, based on DFA (Deterministic Finite Automata). The two parameters n and m are nonnegative integer numbers. The string expression $\{a, b\}^0$ represent the null string, whereas the expression $\{a, b\}^n$ denotes a string obtained by repeating n times the symbols a and b , e.g. $\{a, b\}^3 = a, b, a, b, a, b$.

Definition. Oriented connection exists between two nodes if the second node is the continuation of the first node. This means that the two-element vector formed from the two rightmost elements of the first node and the two-element vector formed from the two leftmost elements of the second (inner) node are the same (Fig. 49). Formally:

Oriented connection exists between two nodes: $n_1 \rightarrow n_2$

$$n_1 = \{S_1, S_2, S_3\} \text{ and } n_2 = \{Q_1, Q_2, Q_3\} \text{ iff. } S_2 = Q_1 \text{ and } S_3 = Q_2, \text{ where } S_i, Q_i \in \{-1, 1\}. \quad (\text{Inner Node})$$

$$n_1 = \{X_0, L_2, L_3\} \text{ and } n_2 = \{S_1, S_2, S_3\} \text{ iff. } L_2 = S_1 \text{ and } L_3 = S_2, \text{ where } L_i, S_i \in \{-1, 1\}. \quad (\text{Left Node})$$

$$n_1 = \{S_1, S_2, S_3\} \text{ and } n_2 = \{R_1, R_2, R_3\} \text{ iff. } S_2 = R_1 \text{ and } S_3 = R_2, \text{ where } S_i, R_i \in \{-1, 1\}. \quad (\text{Right Node})$$

Definition. A path is a sequence of connections.

A pattern is a stable equilibrium point if a path exists from a node in the first column to a node in the last column (Fig. 50).

Using this graph based rigorous technique all the stable equilibrium patterns can be represented as directed paths. It is revealed that the parameter space can be divided into twenty-five sub-regions. Except the central unstable region $\{|X_0| < 1 - (p-1)/s, |X_{N+1}| < 1 - (p-1)/s\}$ all the sub-regions exhibit at least one stable equilibrium point.

This central unstable region is the most probable parameter sub-space for finding complex behaviour. However, it cannot be excluded that in the regions with one or more stable equilibrium points are other attractors such as limit cycles or chaotic attractors. In fact, later I will show that this is the case.

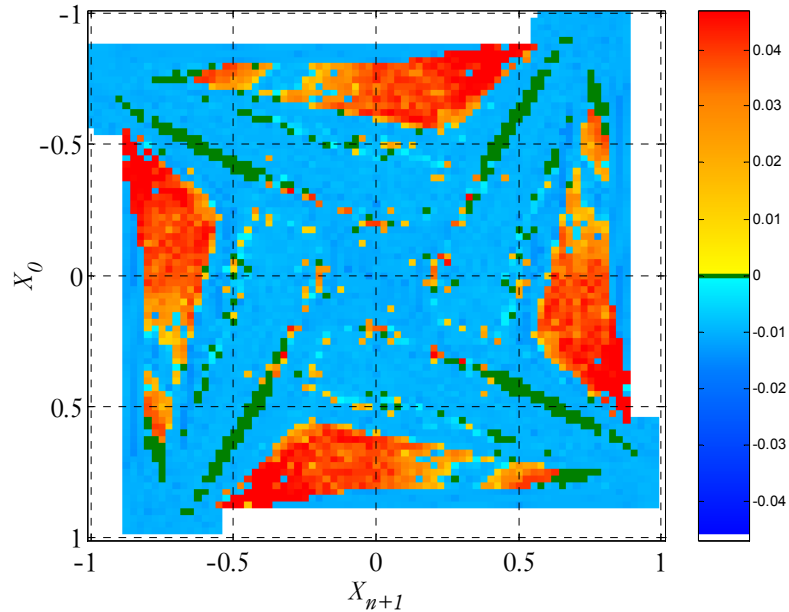


Figure 51. The parameter space of a 4x1 CNN with $A=[0.9 \ 1.1 \ -0.9]$. Yellow to red colors mean positive Lyapunov exponents, green means two zero exponents (torus or quasi-periodic dynamics), blue denotes periodic behaviour with single zero exponent as the greatest one. White means stable region with all exponents being zero.

4.2 Nonlinear dynamics

4.2.1 Study of 1D CNN with a selected template

I used first order 1D CNN with linear templates (2.4). I considered the following sign anti-symmetric template without any input:

$$A=[0.9 \quad 1.1 \quad -0.9]$$

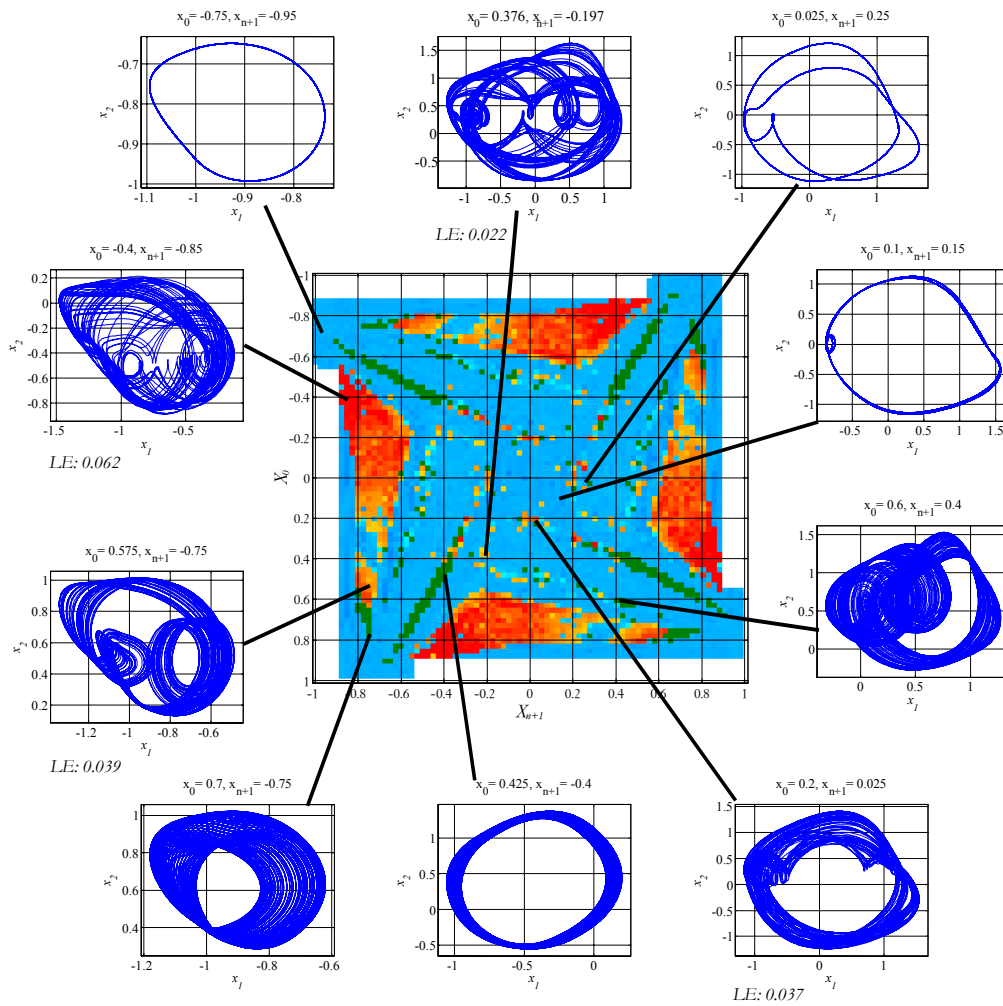


Figure 52. Lyapunov map of a 4x1 CNN with $A=[0.9 \quad 1.1 \quad -0.9]$. Some observed attractors are shown. Color code. Yellow-red: chaotic attractor; green: torus (quasi-periodic); blue: limit cycle (periodic); white: equilibrium point.

I used numerical technique to explore the unstable region of parameter space. Because of the need for computation of the Jacobian I replaced the PWL transfer function by function f_ε a C^1 function that is equal to the original in the limit [93].

$$f_\varepsilon(x) = \begin{cases} -1 & x \leq -(1+\varepsilon) \\ \frac{1}{4\varepsilon} [x^2 + 2(1+\varepsilon)x + (1-\varepsilon)^2] & |x+1| \leq \varepsilon \\ x & |x| \leq 1-\varepsilon \\ -\frac{1}{4\varepsilon} [x^2 - 2(1+\varepsilon)x + (1-\varepsilon)^2] & |x-1| \leq \varepsilon \\ 1 & x \geq 1+\varepsilon \end{cases}$$

I considered a 4x1 CNN and I discovered several well-separable areas with characteristic dynamics. With zero initial state for all cells the unstable region turned to be centrally symmetric. Limit cycles were found in most part of the parameter space separated by areas of chaotic and quasi-periodic attractors (Fig. 51). Limit cycles are denoted by blue color, yellow-red color represents chaotic behaviour while green means the presence of quasi-periodic dynamics. The colors reflect the magnitude of the highest Lyapunov exponents of the system; of course in the case of quasi-periodic dynamics there are two highest exponents: two zeros. Exponents were computed using the algorithm described in [121] that is based on the QR decomposition of the evolution of a perturbation. The perturbation is computed using the time dependent Jacobian of the system making use of the Lyapunov exponent toolbox (LET) for MATLAB by S. W. Kam.

Fig. 52 shows the map of a 4x1 CNN with $A=[0.9 \ 1.1 \ -0.9]$ with few characteristic attractors projected into 2D. Observe the periodic, quasi-periodic and chaotic attractors.

In the following, I choose a line in the parameter space of Fig. 51 and show some typical route from one dynamical state to another. Fig. 53 shows the bifurcation diagram corresponding to $\{X_0; X_{n+1}\} = \{0.66; -0.78 \dots 0.2\}$. Indeed, this diagram is equivalent to the line $X_0=0.66$ of Fig. 51.

Fig. 54 shows the zoomed part of Fig. 53 with period doubling bifurcation $X_{n+1} = \{-0.3665 \dots -0.361\}$. Fig. 55 presents the corresponding 2D projected trajectories.

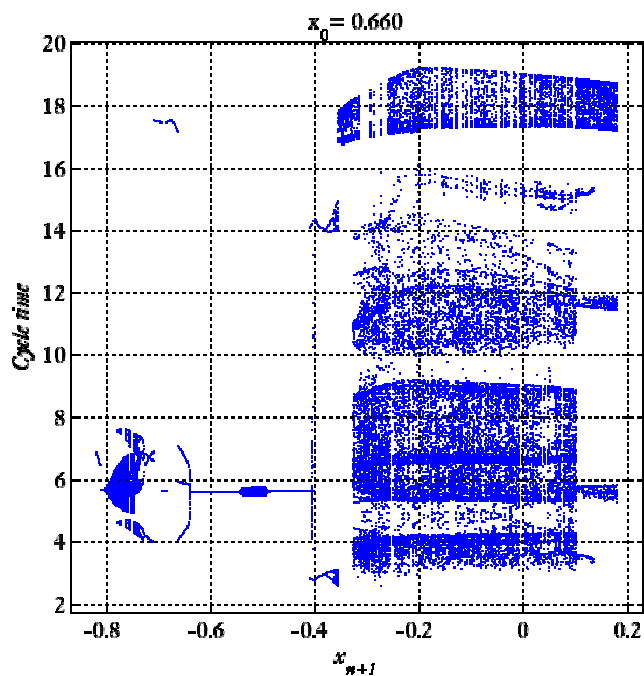


Figure 53. Bifurcation diagram with all kind of dynamic behaviour. Compare diagram with the corresponding row of Fig. 51. $\{X_0; X_{n+1}\}=\{0.66, \dots\}$; $A=[-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

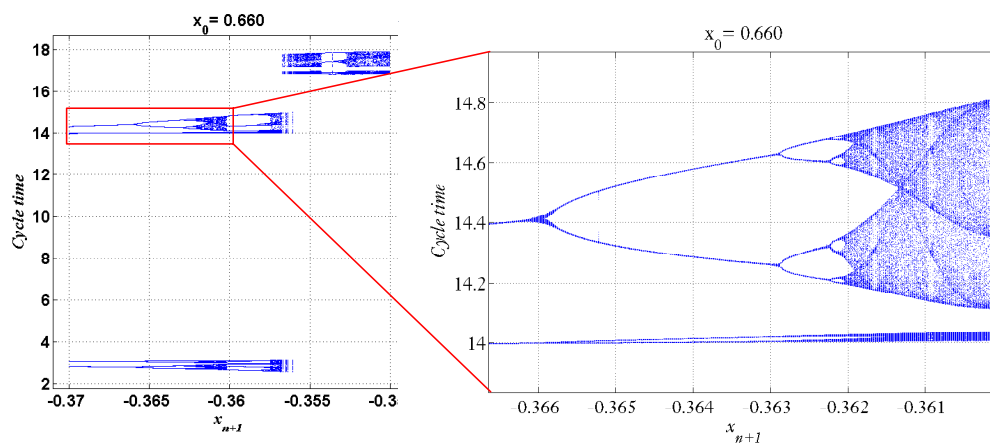


Figure 54. Zoomed part of Fig. 53 with period doubling. $\{X_0; X_{n+1}\}=\{0.66, \dots\}$; $A=[-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

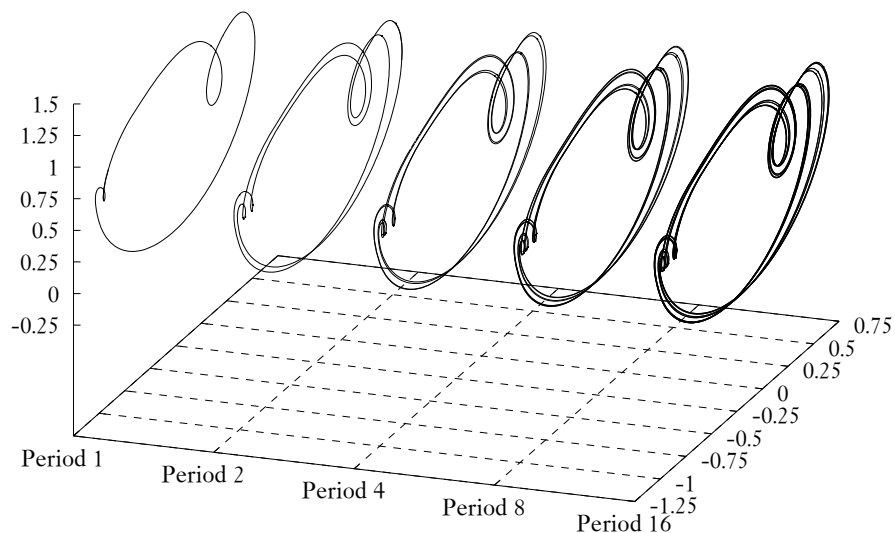


Figure 55. Trajectories with period doubling. $\{X_0; X_{n+1}\} = \{0.66, -0.3665 \dots - 0.361\}$; $A = [-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

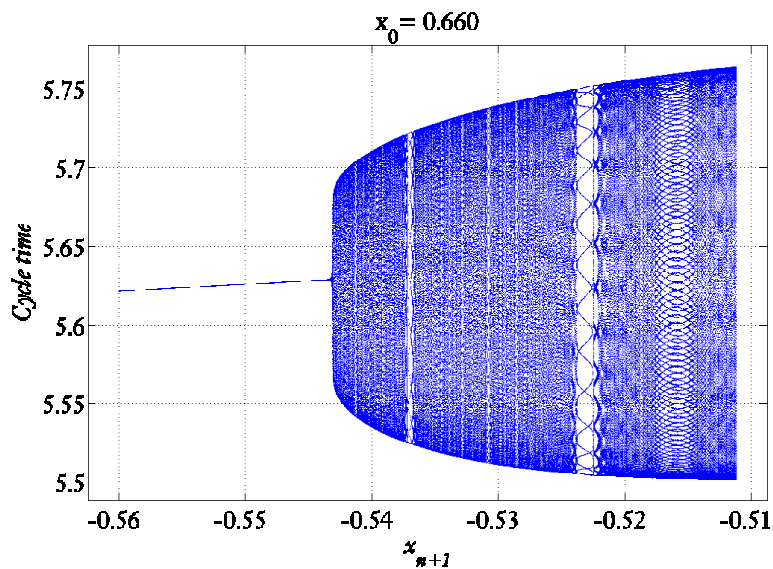


Figure 56. Bifurcation diagram. Naimark-Sacker bifurcation. $\{X_0; X_{n+1}\} = \{0.66, -0.56 \dots -0.51\}$; $A = [-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

Fig. 56 shows another zoomed part from $X_{n+1} = \{-0.56 \dots -0.51\}$ presenting a periodic to quasi-periodic transition process as we increase X_{n+1} . Observe the periodic windows in the quasi-periodic region. The corresponding trajectories are shown in Fig. 57.

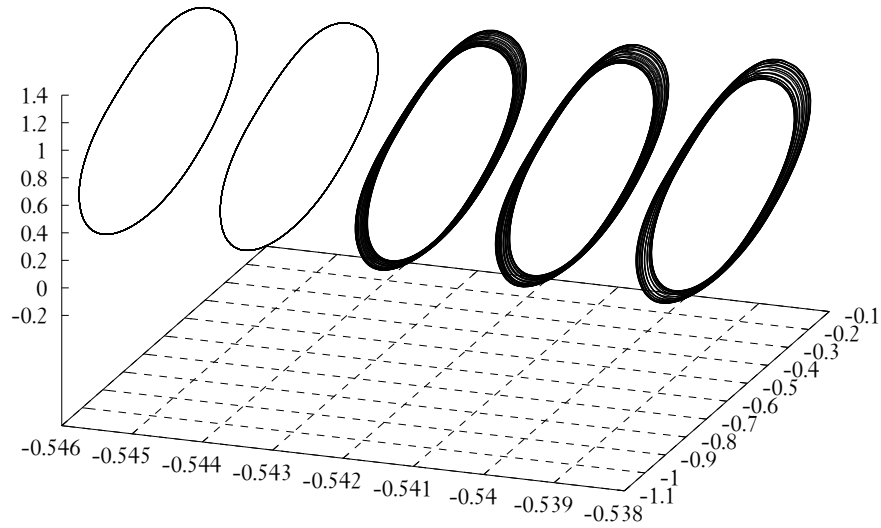


Figure 57. Trajectories of limit cycle to torus transition. $\{X_0; X_{n+1}\}=\{0.66, -0.546\dots-0.538\}$; $A=[-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

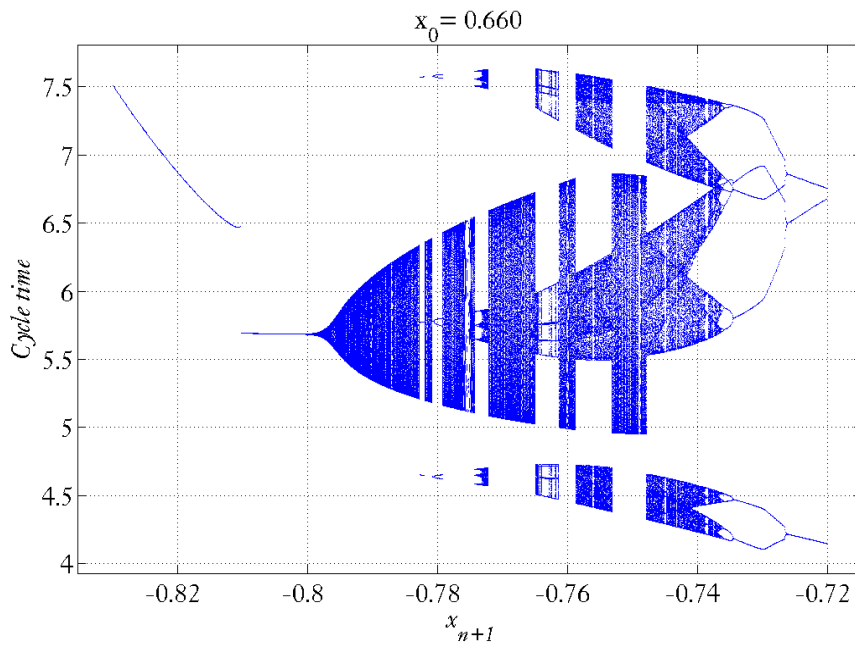


Figure 58. Bifurcation diagram. Alternating Torus-Chaos windows. $\{X_0; X_{n+1}\}=\{0.66, \dots\}$; $A=[-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

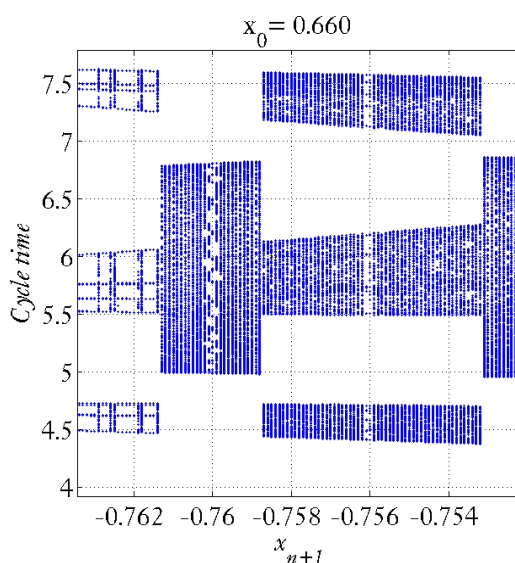


Figure 59. Bifurcation diagram. Torus-Chaos windows - zoomed region. $\{X_0; X_{n+1}\}=\{0.66, \dots\}$; $A=[-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

Fig. 58 presents a region of the bifurcation diagram of Fig. 53 in which periodic, quasi-periodic and chaotic regions are alternating. Fig. 59 shows a zoomed part in which torus-chaos transition is shown. Some corresponding trajectories are presented in Fig. 60. Fig. 61 shows the 2D projection of Poincare map of a quasi-periodic attractor of the 4×1 CNN with $A=[-0.9 \ 1.10 \ 0.9]$ at $\{X_0; X_{n+1}\}=\{0.66, -0.53\}$.

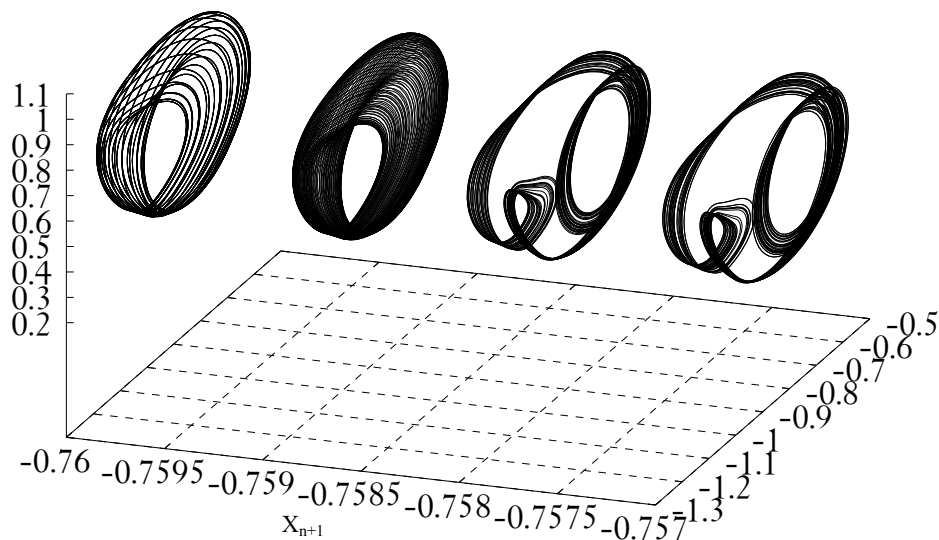


Figure 60. Torus to Chaos transition. Trajectories. $\{X_0; X_{n+1}\}=\{0.66, -0.76\dots-0.757\}$; $A=[-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

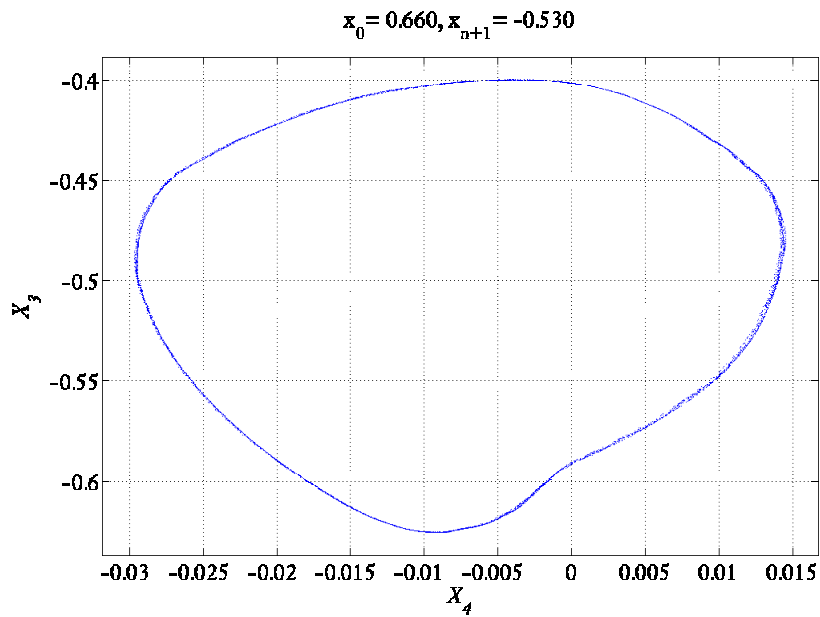


Figure 61. 2D projection of the Poincaré map of a torus at $\{X_0; X_{n+1}\} = \{0.66, -0.53\}$; $A = [-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

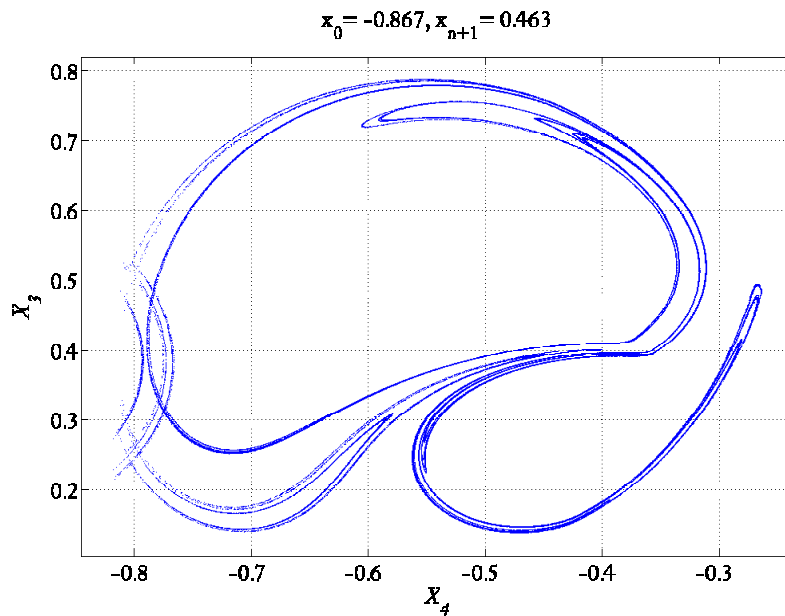


Figure 62. The "Brain". 2D projection of the Poincaré map of four cell CNN at $\{X_0; X_{n+1}\} = \{-0.867, 0.463\}$; $A = [-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

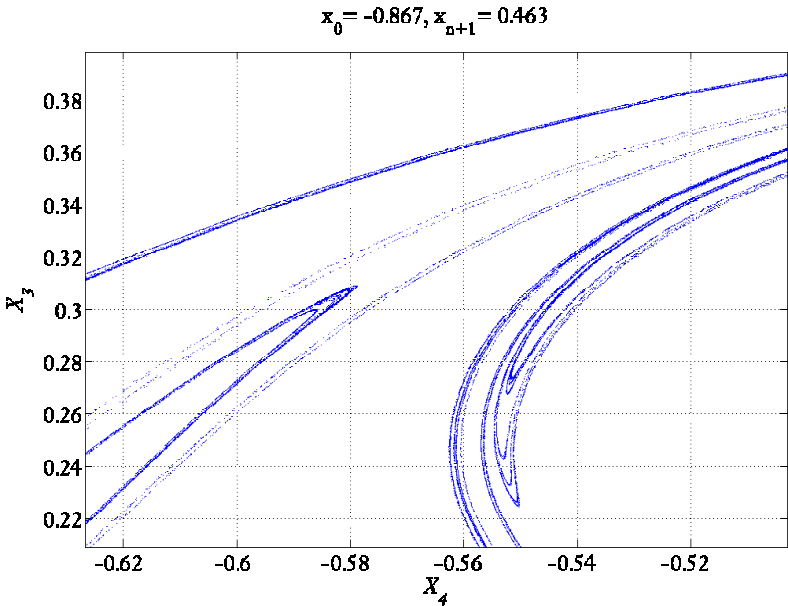


Figure 63. Zoomed part of the Brain map. Observe the fractal structure. $\{X_0; X_{n+1}\} = \{-0.867, -0.463\}$; $A = [-0.9 \ 1.10 \ 0.9]$; Dimension: 4×1 .

Fig 62 and 63 show the 2D projection of the first return map of the 4×1 CNN with $A = [-0.9 \ 1.10 \ 0.9]$ at $\{X_0; X_{n+1}\} = \{-0.867, -0.463\}$.

*Sub thesis II/b.***Chaotic dynamics, coexistence of attractors in 1D CNN**

I uncovered novel complex dynamics in 1D CNNs with a sign-antisymmetric template class. I investigated the effect of the boundary condition, dimension of the system and initial state. I found that boundary condition behaves as bifurcation parameter. I found coexistence of attractors depending on the initial state.

II/b. Effect of dimension and initial state, coexistence of attractors

I showed that the steady state dynamics depends on the initial state: attractors coexist simultaneously in the same system. I showed that hyperchaotic behavior can occur in a CNN that consist of more than five cells. I demonstrated that the parity of the size of the 1D system drastically changes the dynamics.

4.2.2 Coexistence of attractors in a 4×1 CNN

Perhaps the most important result in this thesis is the simultaneous existence of different attractors. This could be exploited by further research for e.g information storage.

As I noted before the existence of stable equilibrium points does not necessarily exclude the presence of other types of attractors. Indeed, in regions

- a) $\{ |X_0|s < -(p-1)+s, |X_{N+1}+1|s < p-1 \}$
- b) $\{ |X_0+1|s < p-1, |X_{N+1}|s < -(p-1)+s \}$
- c) $\{ |X_0|s < -(p-1)+s, |X_{N+1}-1|s < p-1 \}$
- d) $\{ |X_0-1|s < p-1, |X_{N+1}|s < -(p-1)+s \}$

I found the coexistence of stable equilibrium points, limit cycles and chaotic attractors that are another example to that of the authors in section V. of [122].

The presence of different attractors is also valid in the unstable region $\{ |X_0| < 1-(p-1)/s, |X_{N+1}| < 1-(p-1)/s \}$.

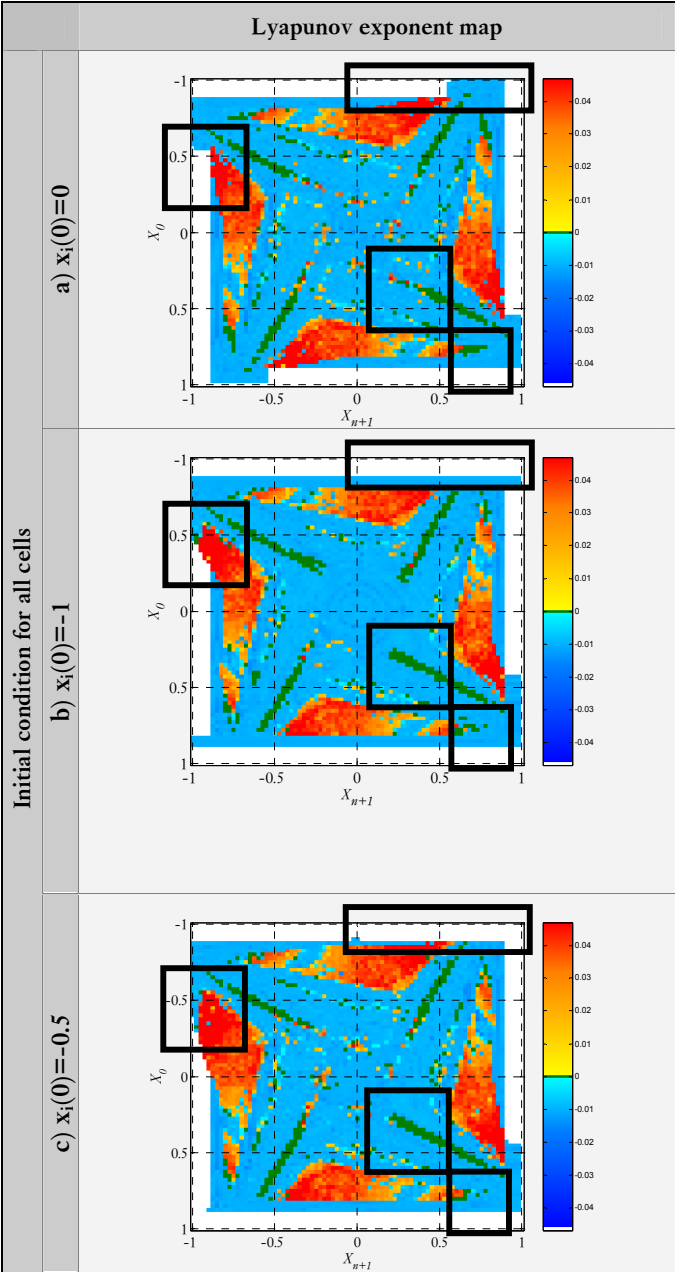


Figure 64. Coexistence of attractors in a 4x1 CNN with a) $x_i(0)=0$, b) $x_i(0)=-1$, c) $x_i(0)=-0.5$, $i=1..4$ (all cells are set to the same value). Observe the coexistence of chaotic \leftrightarrow limit cycle, chaotic \leftrightarrow torus and torus \leftrightarrow limit cycle attractors.

4.2.3 Effect of the perturbation of template parameters on the exponents

As we change the template parameters the map obviously changes . Fig 65 shows the effect of s while Fig. 66 shows the effect of p .

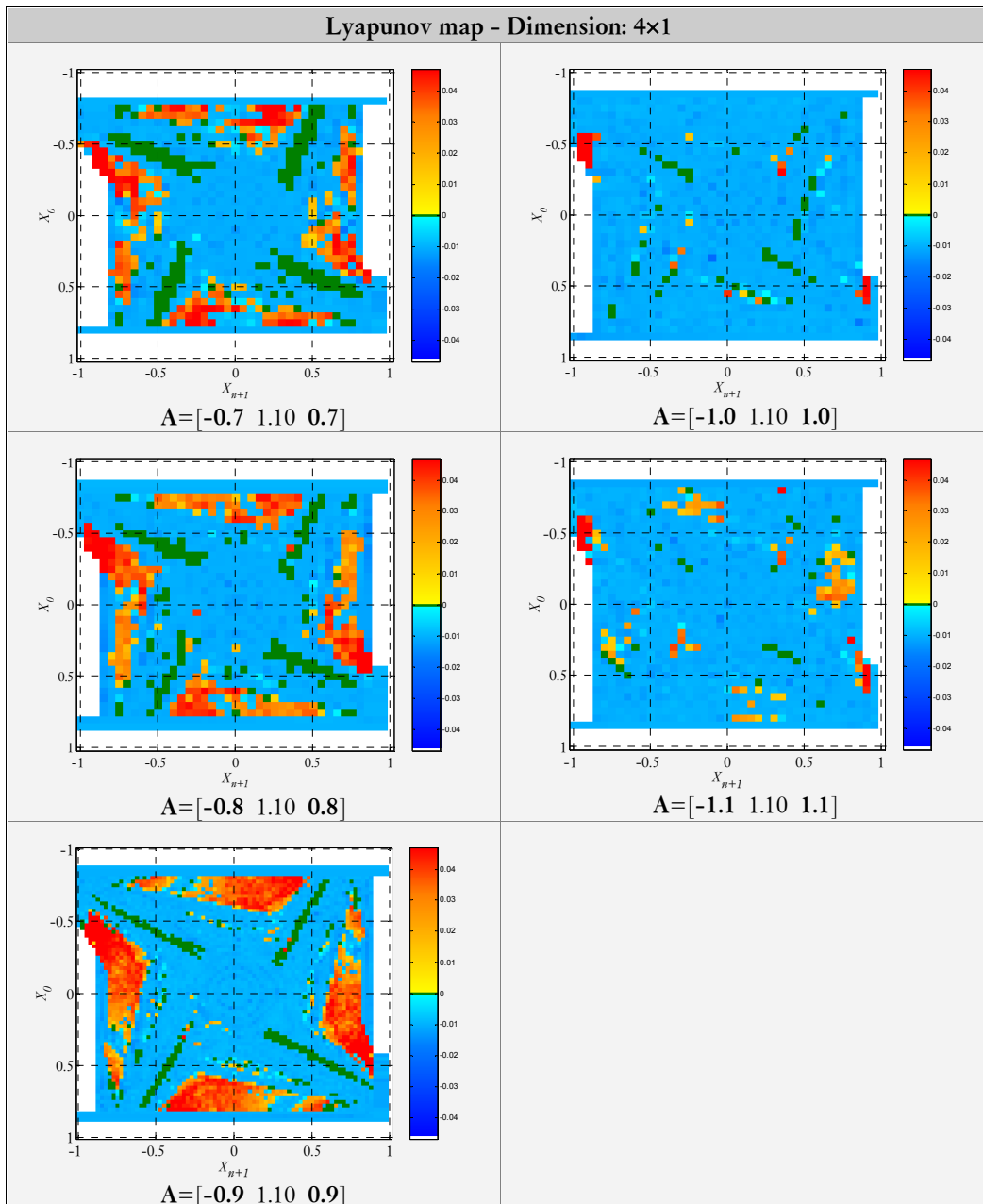


Figure 65. Effect of s on the Lyapunov map in a 4×1 CNN . Initial state is -1

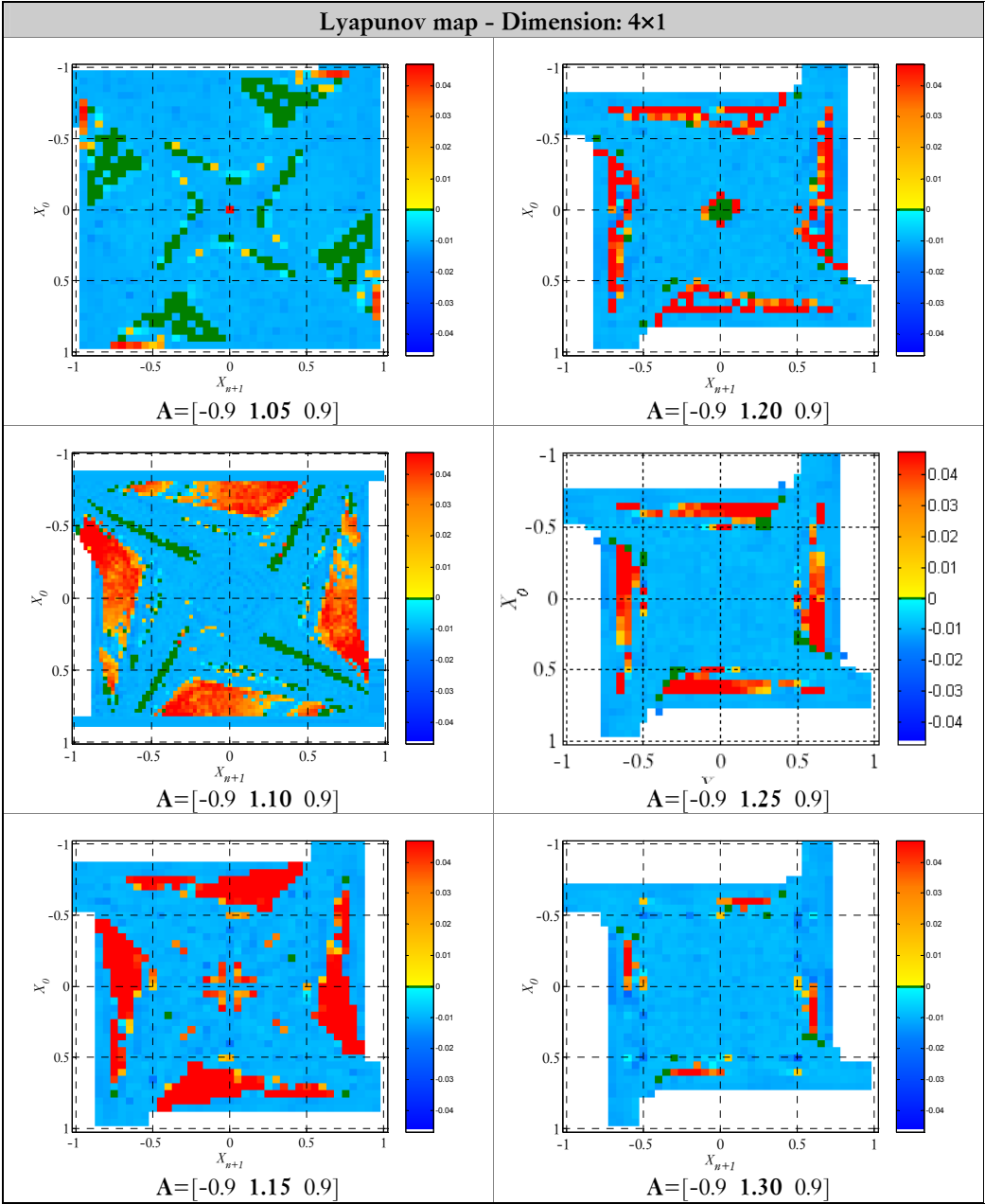


Figure 66. Effect of p on the Lyapunov map in a 4×1 CNN .

4.2.4 Higher dimension - Hyperchaos

I showed that hyperchaotic behaviour can occur even in a 6×1 size CNN. Hyperchaos means that the system has more than one positive Lyapunov exponents.

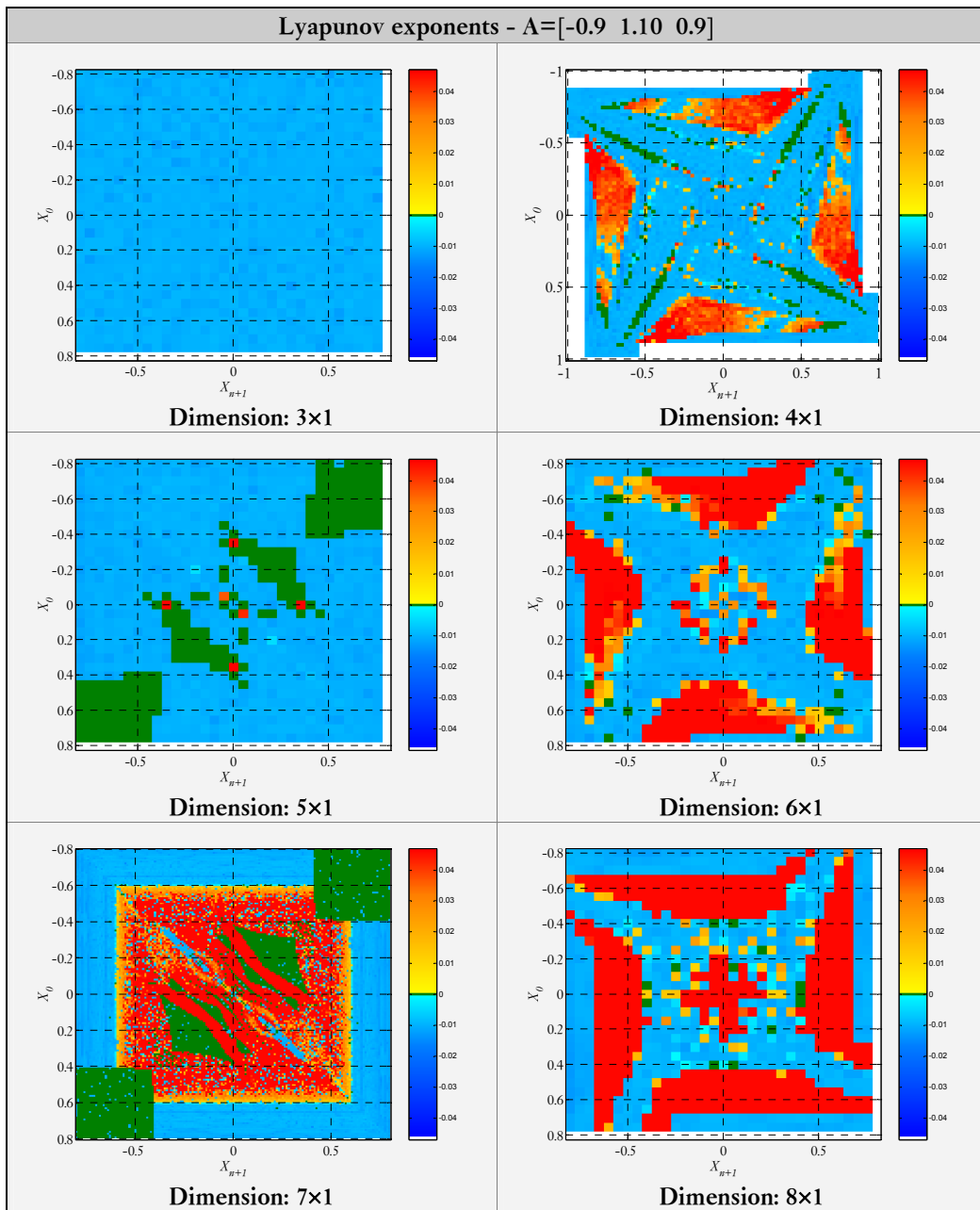


Figure 67. Effect of the dimension of the CNN on the Lyapunov map. Dimensions of the CNN are from 3×1 to 8×1 . $A=[-0.9 \ 1.10 \ 0.9]$

This can only appear in dynamical systems having dimension $N > 3$. In the case of CNN the presence of hyperchaos was found with $N \geq 6$. Fig. 67 shows the effect of the size of the system on the dynamics.

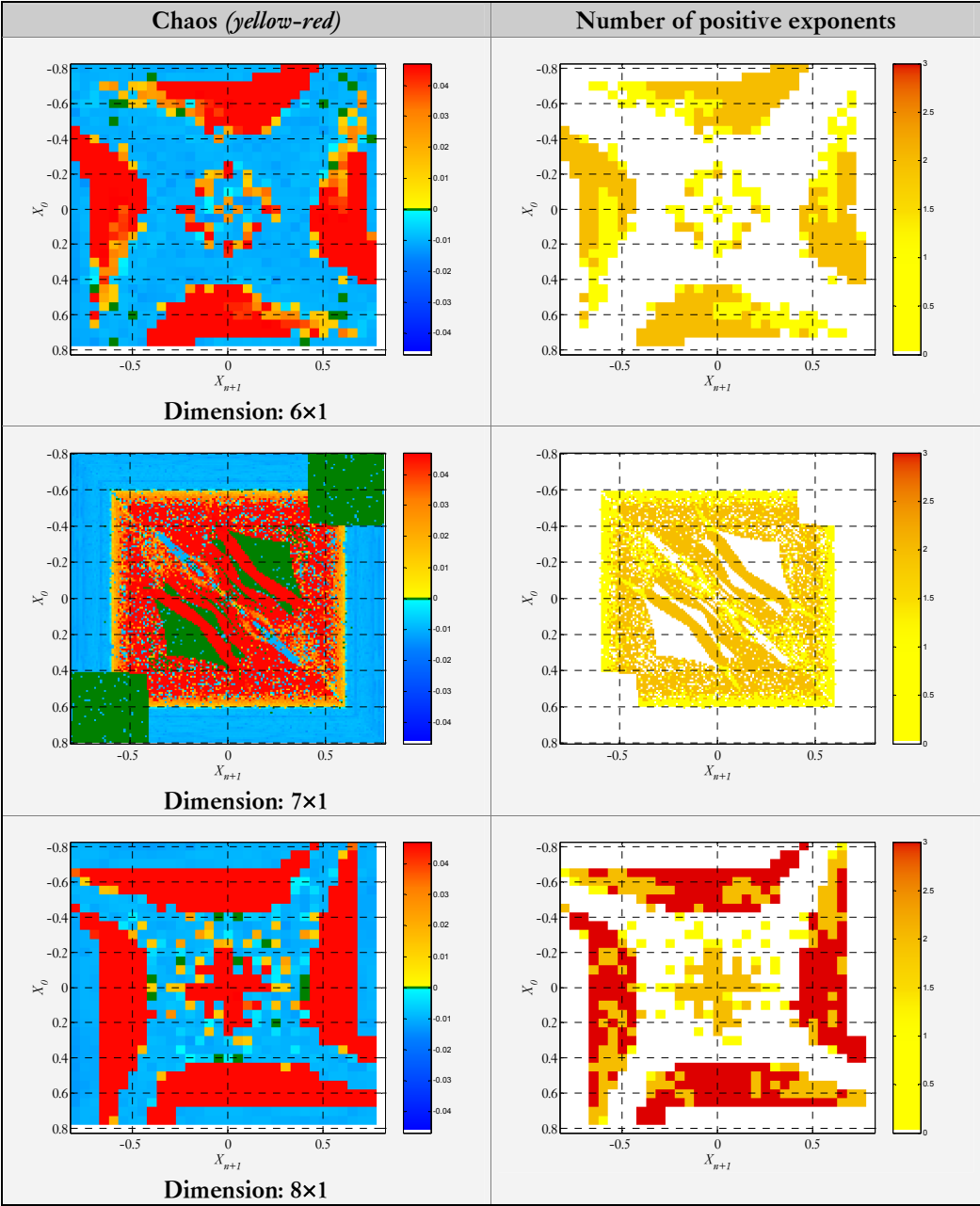


Figure 68. Hyperchaos in 6x1, 7x1 and 8x1 CNNs. In the right column the color denotes the number of positive Lyapunov exponents.

Observe that parity of the number of cells can play an important role. Maps of the systems having the same parity of number of cells are similar. Fig. 68 shows the systems that

4. COMPLEX DYNAMICS IN 1D CNN

exhibit more than one positive Lyapunov exponents. The color-coded number of positive exponents are shown in the right column.

5. *Special wave operator*

Third Thesis

Curvature controlled trigger wave

I showed that the CNN with a nonlinear diffusion type template class computes the approximation of the so-called “Curve Shortening Flow” of shapes. Indeed, the process can be described as local curvature controlled trigger wave propagation in the CNN array. In this framework the curve is represented as the boundary region of a solid black shape against white background. As a result of the diffusion-like evolution of the CNN array, a smooth 2D transition zone called *boundary region* is formed between black and white regions. The width of the region is controlled by the template parameters. The evolution of the shape of this region corresponds to the evolution of the curve. To show this, at first I analyzed the characteristics of the boundary region of the evolving shape than I studied the dynamics of this transition region.

III/a. Stability of the boundary

I proved that the boundary region is stationary provided that the region is horizontal or vertical. I gave relation on the number of linear cells in a horizontal or vertical boundary region. I gave approximation on the minimal number of linear cells of arbitrary oriented straight boundaries.

III/b. Dynamics of the boundary region

I showed that the temporal derivative of the state variable of a linear cell being in the boundary region is the quasi-linear function of the local curvature of the iso-intensity line to which the cell belongs provided that the state variable of the cell is close to zero. This is fulfilled in the centerline of the boundary region. The boundary region changes so that its local curvature is zero. In other words the shape contours get smoothed provided that spatial constraints make this possible. I gave method to compute the convex hull of a shape in one transient. Moreover, I gave method to separate two-dimensional clusters of points.

NONLINEAR (active) waves are natural phenomena that can be found everywhere around us in Nature and described by various fields of sciences. They can be observed in e.g. chemistry [4], or biology [28,29]. Their basic property is that the wave propagates in an excitable (bistable) medium in a way that a small unit of the medium transits from one stable state to another at the expense of energy stored in the medium. The transition triggers the local neighborhood thus the propagation is self-sustaining; from here comes the name: auto(nomous)-wave (R. V. Khorhlov). Such nonlinear active waves are autowaves [31] (traveling waves, trigger waves, target (or concentric) waves [4]) and spiral waves [29].

Due to their local dynamics nature, considering continuous descriptions in space and time, reaction-diffusion PDEs are the plausible mathematical models for these phenomena.

PDEs have proved to be useful as a general framework for image processing [32,33]. Multiscale analysis originally based on linear PDE filters [34]; later nonlinear diffusion filters were developed to avoid the artifacts [35-38]. In the active contour (or snakes) approach for object detection the contour propagation is also governed by PDEs [39,40]. In the level-set approach a PDE controls the evolution of a set of iso-intensity contours of the image [41,42].

Till now this field remained a focused and intensive research area. Several works revealed the connection between the different PDE related image-processing methods [32,43-45].

The main drawback of the approach is the increased computing power requirement because of the need of the numerical solution of PDEs. With parallel computing architecture this requirement can be overcome. This implies spatial discretization that gives an approximation of the original PDE. However with an appropriate spatial constant the original dynamics can be reproduced. Thus the PDE dynamics is described by a set of ODEs.

For parallel computing of PDEs obviously we need programmable processors. However, they do not need to be as complex and general as a microprocessor or DSP. It is sufficient if they can 'only' compute ODEs. At first sight this might seem to make no sense, however considering analog implementation this approach shows itself much more attractive.

A suitable computing model that is hardware feasible can be the Cellular Nonlinear Networks [46-49] introduced in Chapter 2. The most widely applied variant of it consists of first order nonlinear dynamical systems that are locally coupled and organized into a two-dimensional array. An intensive research has recently emerged in the CNN field. Several linear, nonlinear filtering methods and wave computing algorithms were developed [56,64-74]. The structure of the CNN naturally makes it a possible tool as spatially discretized PDE computer [75,76].

The CNN as an active medium has been studied extensively. Several types of nonlinear waves have been reproduced and analyzed on this structure depending on the complexity of the elementary cells [31,59,77-82].

5.1 Shape deformation

Recently several papers have dealt with the CNN as a wave-computing device [57,59,64,65,75,83,84]. It has been proved to be a suitable computing frame for wave-based image processing. Some of these algorithms used autowaves for specific tasks [74], others used trigger waves for some kind of morphological processing in which the shape is deformed by the wave.

These trigger wave-based processes were one-way: if the spreading wave triggered a cell then it remained triggered [64]. However, using an appropriate nonlinear diffusion operator this property can be violated yielding a qualitatively different behaviour [85].

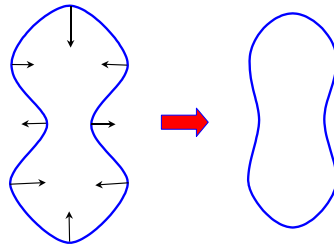


Figure 69 Curve shortening flow

The phenomenon that is well-known from the image processing literature [167,168] is described by

$$\frac{\partial C(l,t)}{\partial t} = -\kappa \vec{N} \quad (5.1)$$

where C is the boundary vector of coordinates (the planar curve), κ is the local curvature, \vec{N} is the outward normal, l is the curve parameter, t is the scale parameter (Fig. 69).

This equation is known as the geometric heat equation or *curve shortening flow* in differential geometry. Several work dealt with this flow (see for review e.g. [33]). It had been proved [86,87] that all closed contours collapse into a single point under this flow.

During the evolution of this flow the shape (closed contour) propagates along its outward normal direction and collapses under its curvature, which gives rise to obtain a hierarchy of shapes and thus a scale space for shapes can be formed [88-90]. These scale spaces

and their enhancements gave a suitable framework for recognition, classification and retrieval of shapes. Due to numerical and theoretical reasons Osher and Sethian proposed an approach in which the curve flow is embedded as the zero level set of a surface [42]. The surface is evolved by its mean curvature and due to the special shape of the surface the mean curvature of the surface corresponded to the curvature of the embedded curve at the zero level set and the evolution of the surface contained the evolution of the original curve. See [166] for on-line reference.

The CNN based nonlinear diffusion works similar way. The shape is given as the initial state of the array. The zero level set (or isointensity line) of the state is considered as the evolving contour of the shape during the evolution (see Fig. 70). Black (+1) color means “inside”, white means “outside” of the shape.

The *boundary region* of a shape is a one- or two-dimensional transition zone between opposite-signed saturated cells (1D) or between regions (2D; see Fig. 73, 75 and 79). The state variables of cells that constitute the boundary are in the linear part of transfer function (5.3) i.e. $-1 < x_{ij} < +1$. Please do not to mistake *boundary region* for *boundary condition* of the CNN array in this thesis.

It will be shown that due to the flat shape¹ of the boundary region (see Fig. 78) the mean curvature of the surface and the curvature of the embedded curve are practically the same. Therefore the boundary region evolves according to the curvature of the embedded shape.

Using the discretized version of the following nonlinear diffusion operator $A(\cdot) = D\nabla^2 f(\cdot) + a_f f(\cdot)$ the CNN computes the approximation of the curve shortening flow in one transient.

$$\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{k=-q}^q \sum_{l=-q}^q A_{kl} f(x_{i+k, j+l}(t)), \quad A = \begin{bmatrix} 0 & D/4 & 0 \\ D/4 & a_c - D & D/4 \\ 0 & D/4 & 0 \end{bmatrix} \quad (5.2)$$

where $q=1$, x is the state of the CNN, $f = \lim_{\varepsilon \rightarrow 0} f_\varepsilon$ is a nonlinear transfer function [93] defined by (5.3).

$$f_\varepsilon(x) = \begin{cases} -1 & x \leq -(1+\varepsilon) \\ \frac{1}{4\varepsilon} [x^2 + 2(1+\varepsilon)x + (1-\varepsilon)^2] & |x+1| \leq \varepsilon \\ x & |x| \leq 1-\varepsilon \\ -\frac{1}{4\varepsilon} [x^2 - 2(1+\varepsilon)x + (1-\varepsilon)^2] & |x-1| \leq \varepsilon \\ 1 & x \geq 1+\varepsilon \end{cases} \quad (5.3)$$

At the beginning of the evolution of the CNN the shape is implicitly embedded into a surface through the blurring of the diffusion process (Fig. 71). As a result, a linear transition

¹ the linearity of the cross section of the surface along the gradient direction. This shape is a special characteristic of the CNN and maintained during the evolution.

zone is formed between the black and white regions. Cells belonging to the zone are in the linear region of (5.3). Cells being black or white are said to be saturated. They are in the saturation region of (5.3). The width of the linear region can be influenced by parameters D and a_c (Fig. 72). However the blur is spatially limited: the nonlinear character of the diffusion maintains the constant width and the gradient of the linear zone. This width is independent of the orientation and the shape of the zone throughout the transient. When the desired curve smoothing is reached the transient is stopped.

The CNN may be used as an effective parallel “curve shortening flow computer” and we can use the results developed for scale-spaces of shapes ([88-90]).


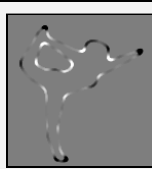


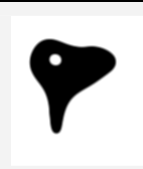
Output					
time	t: 0	Local curvature at t:1	t: 4	14	34

Figure 70. $s=0.6, p=-1$, boundary condition of the array is zeroflux

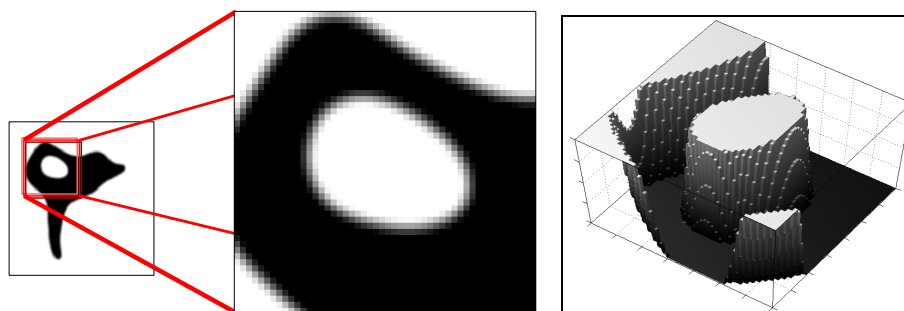


Figure 71. Linear zone of the interface between the black and white regions.

5.2 Number of linear cells

In this section I derive some relations on the boundary zone of the embedded shapes. At first I consider straight boundaries that are horizontal or vertical. In the second subsection boundaries are considered that are oriented along the $\pi/4$ direction. I derive a conjecture on the length of the linear zone along the gradient.

5.2.1 Horizontally or vertically aligned boundary region

Several authors investigated one-dimensional CNNs. In [91] Thiran et al. proved that the stable pattern of a 1D CNN with $p, s > 0$, $-1 < (p-1)/2s < 1$ consists of a succession of alternately black and white pair of adjacent saturated cells, separated by boundaries of B linear pixels. He gave conditions necessary for the stability of a pattern and also gave the range of stable patterns. This theorem can be easily extended into 2D CNN having hori-

zontally or vertically oriented patterns (Fig 73). In 2D the linear cells form the *boundary region* between the black and white regions.

	Stable pattern	Stable pattern	Unstable pattern	Unstable pattern	Stable pattern
initial state					
snapshot					
	$a_e=2.39,$ $D=2.4$ $B_{1D}=0$	$a_e=2.39,$ $D=2.4$ $B_{1D}=0$	$a_e=1.3, D=2.4$ $B_{1D}=3$	$a_e=1.06,$ $D=2.4$ $B_{1D}=8$	$a_e=2.45, D=2.4$ $B_{1D}=0$

Figure 72. The effect of the self-feedback on the propagation; B_{1D} is the number of linear cells.



Figure 73. Extension of 1D stable pattern into 2D. $B=7$

The following proposition shows the relation of the one- and two-dimensional stable patterns provided that in 2D the boundary regions are oriented vertically or horizontally (i.e. the gradient vector of the 2D image is parallel with or perpendicular to the x axis of the image respectively).

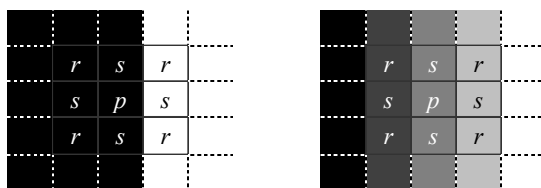


Figure 74. 2D vertically oriented boundary regions with and without linear cells. The dynamics of all rows are the same.

Proposition:

$$\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{k=-q}^q \sum_{l=-q}^q A_{kl} f(x_{i+k, j+l}(t)), \quad \mathbf{A} = \begin{bmatrix} r & s & r \\ s & p & s \\ r & s & r \end{bmatrix} \quad (5.4)$$

where $q=1$. Let $s, r > 0, p < 1, 4s + 4r + p > 1$. Moreover, let the initial state of the infinite or periodic 2D CNN (5.4) consist of identical rows or columns (Fig. 74).

Then the one-dimensional equivalent CNN has the form

$$\dot{x}_i = -x_i + \hat{s}f(x_{i-1}) + \hat{p}f(x_i) + \hat{s}f(x_{i+1}), \quad \mathbf{A} = \begin{bmatrix} \hat{s} & \hat{p} & \hat{s} \end{bmatrix} \quad (5.5)$$

$$\hat{s} = 2r + s, \quad \hat{p} = 2s + p$$

and the number of linear cells [91] in the boundary zone is an integer B:

$$\boxed{\frac{\pi}{\pi - \arccos(\mu)} - 2 < B < \frac{\pi}{\pi - \arccos(\mu)} - 1} \quad \text{where } \mu = \frac{\hat{p} - 1}{2\hat{s}} \quad (5.6)$$

Proof:

The dynamics of all rows are the same therefore the array can be written as a vector:

$$\left. \begin{array}{l} rx_{i-1} \quad sx_i \quad rx_{i+1} \\ sx_{i-1} \quad px_i \quad sx_{i+1} \\ rx_{i-1} \quad sx_i \quad rx_{i+1} \end{array} \right\} \dot{x} = -x + (2r + s)f(x_{i-1}) + (2s + p)f(x_i) + (2r + s)f(x_{i+1}) \quad (5.7)$$

Thus, the 2D array is transformed to 1D provided that the initial state is made up of black and (gray and) white regions separated by vertical regions. Doing so, we can use the theory from [91] for the number of linear cells equation (5.6). The same results are hold for horizontally aligned boundary regions as well. ■

5.2.2 Boundary region aligned with $\pi/4$

Let us consider an infinite 2D CNN controlled by template \mathbf{A} . The governing equation is the same as in (5.4) with the following:

$$s > 0, p < 1, 4s + p > 1, \quad \mathbf{A} = \begin{bmatrix} 0 & s & 0 \\ s & p & s \\ 0 & s & 0 \end{bmatrix} \quad (5.8)$$

To gain a relationship between p, s and B , I use a way of thinking similar to that of in [91] but I extend it into 2D with restrictions. Compared to the 1D case, the degree of freedom of the parameters increases greatly. The number of possible stable patterns also increased a lot. Due to these only few things can be stated.

In spite of the fact that the linear analysis is not valid for equation (5.4) it is still useful for deriving some results for the number of linear cells in 2D.

The two-dimensional discrete Fourier transform of the template matrix is:

$$\tilde{A}[\omega] = p - 1 + 2s(\cos\omega_1 + \cos\omega_2) \quad (5.9)$$

This describes the filtering characteristic of the CNN in the linear region.

Let $\omega_1 = \omega_2$. This means that the boundary zone is aligned with $\pi/4$ direction (see Fig. 75). Since we only consider straight zones this assumption can be valid.

The cut-off frequency ω_c is given by:

$$0 = p - 1 + 2s(\cos\omega_1 + \cos\omega_2)$$

$$0 = \frac{p-1}{2s} + (\cos\omega_1 + \cos\omega_2)$$

Let $\mu = \frac{p-1}{2s}$ and $\omega_1 = \omega_2 = \omega_c$

$$-\frac{\mu}{2} = \cos\omega_c$$

$$\omega_c = \pi - \arccos\left(\frac{\mu}{2}\right) \quad (5.10)$$

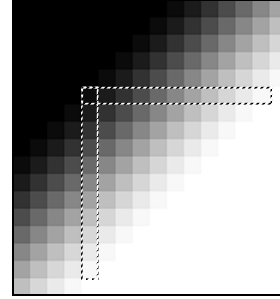


Figure 75. The boundary region is aligned with $\pi/4$ direction. In an infinite CNN are equal the two spatial frequencies of the pattern.

Let us suppose that the stable pattern in equilibrium should have spatial frequency less than or equal to

$$\omega_{\max} = \frac{2\pi}{2(B+2)}$$

This assumption is based on the 1D case. While in 1D we have a 'nice'-shaped region in 2D this is never true except the horizontal or vertical contour case. Therefore the 2D spatial frequencies of the stable pattern are rather uncertain if any stable patterns exist. Due to these difficulties in the following we consider only straight contours oriented at $\pi/4 + k\pi/2$, $k=1..3$.

In two-dimension we have two independent spatial frequencies.

$$\omega_{\max 1} = \frac{\pi}{B_x + 2}, \omega_{\max 2} = \frac{\pi}{B_y + 2}$$

Condition $\omega_{\max 1} = \omega_{\max 2}$ means that the contour zone is straight and oriented at $\pi/4$. Thus the spatial period and therefore the length of linear cells are the same along x and y directions. Suppose that only patterns whose spatial frequency is lower than ω_{\max} are stable as in 1D case. Using equation (5.10):

$$\omega_{\max} = \frac{\pi}{B+2} < \omega_c$$

$$\frac{\pi}{\text{Arccos} \frac{-1 + \sqrt{1-2\mu}}{2}} - 2 < B_{xy} \quad (5.11)$$

where B_{xy} is the closest greater integer.

This gives a lower bound on the number of linear cells if the boundary between the black and white regions is oriented at $\pi/4$.

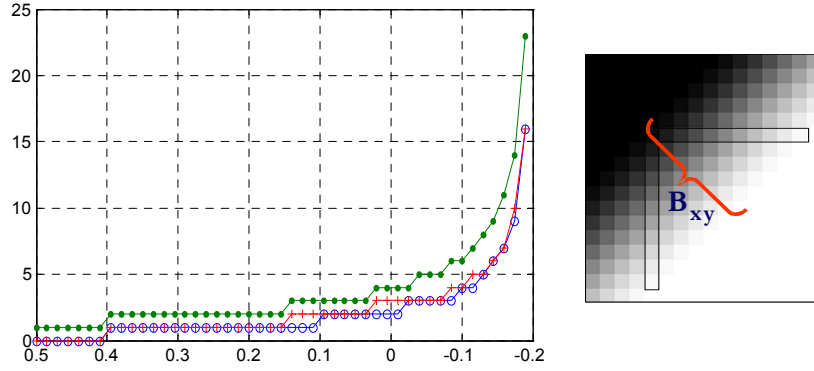


Figure 76. The length of linear cells in 2D as a function of p . dotted: B_{xy} ; + sign: $B_{rot} = B_{xy} \cdot \cos(45^\circ)$; circle: B_{ID}

Although relation (5.11) is obtained without correct mathematical derivation it still gives a reliable approximation and corresponds to the simulations.

In Table 11, B_{ID} is computed using (5.6); B_{xy} comes from (5.11); $B_{rot} = \text{round}(B_{xy} \cos(\pi/4))$. Indeed, experiments show that $B_{rot} = \text{round}(B_y \cos(\alpha))$ if $\alpha < \pi/2$ and $B_{rot} = \text{round}(B_x \sin(\alpha))$ if $\alpha > 0$, where B_x and B_y are the number of linear cells along x and y direction obtained through simulation. Fig. 76 shows the length of the linear cells in 2D as a function of p .

Results suggest corresponding with experiments that the number of linear cells perpendicular to the edge (along the gradient direction) is independent of orientation.

p	0.5	0.2	0.02	-0.06	-0.1	-0.13	-0.148	-0.16	-0.168	-0.174	-0.178	-0.181
B_{ID}	0	1	2	3	4	5	6	7	8	9	10	11
$B_{xy}, \alpha = \pi/4$	1	2	4	5	6	8	9	11	12	14	15	16
B_{rot}	0	1	2	3	4	5	6	7	8	9	10	11

Table 11. The Length of linear cells with differently aligned boundaries. B_{ID} is the horizontally (vertically) oriented one, B_{xy} is the boundary oriented along $\pi/4$, B_{rot} is the 1D equivalent rotated length computed from B_{xy} . $B_{rot} = \text{round}(B_{xy} \cos(\pi/4))$.

Sub thesis III/b.

Curvature controlled trigger wave

I showed that the CNN with a nonlinear diffusion type template class computes the approximation of the so-called “Curve Shortening Flow” of shapes. Indeed, the process can be described as local curvature controlled trigger wave propagation in CNN array. In this framework the curve is represented as the boundary region of a solid black shape against white background. As a result of the diffusion-like evolution of the CNN array, a smooth 2D transition zone called *boundary region* is formed between black and white regions. The width of the region is controlled by the template parameters. The evolution of the shape of this region corresponds to the evolution of the curve. To show this at first I analyzed the characteristics of the boundary region of the evolving shape than I studied the dynamics of this transition region.

III/b. Dynamics of the boundary region

I showed that the temporal derivative of the state variable of a linear cell being in the boundary region is the quasi-linear function of the local curvature of the iso-intensity line to which the cell belongs provided that the state variable of the cell is close to zero. This is fulfilled in the centerline of the boundary region. The boundary region changes so that its local curvature is zero. In other words the shape contours get smoothed provided that spatial constraints make this possible. I gave method to compute the convex hull of a shape in one transient. Moreover, I gave method to separate two-dimensional clusters of points.

5.3 The CNN as a curve evolution computer

In the following I present validation of the relation between the local curvature of the boundary of the evolving shape and the dynamics of the CNN.

Let the CNN have the same form as (5.8):

$$\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} f(x_{i+k, j+l}(t)) \quad \mathbf{A} = \begin{bmatrix} 0 & s & 0 \\ s & p & s \\ 0 & s & 0 \end{bmatrix} \quad (5.12)$$

where $s > 0$, $p < 1$, $4s + p - 1 > 0$, f is the same as (5.3). Let $s = D/4$, $p = a_e - D$

Proposition:

The temporal derivative of the state variable of a linear cell being in the boundary region is the quasi-linear function of the local curvature of the iso-intensity line to which the cell belongs provided that the state variable of the cell is close to zero and that the number of linear cells is greater than 3.

$$\dot{x}_{ij} = F(\kappa), \quad \text{where } \kappa \text{ is the local curvature of the iso-intensity line} \quad (5.13)$$

Validation:

Horizontal and vertical boundary regions are stationary based on Theorem 1. In these cases the curvature is zero and the state derivative is also zero since the pattern (the boundary region) is stable. Let us examine now the case when the orientation of the region contour is arbitrary.

Consider the PDE equivalent of the autonomous CNN of eq. (5.12). Let us denote $u = u(x, y, t)$, $u \in \mathbf{R}^2 \times \mathbf{Z} \rightarrow \mathbf{R}$

$$\frac{\partial u}{\partial t} = -u + D\nabla^2 f(u) + a_e f(u), \quad \text{where } D=s, a_e=p+4s \text{ is constant} \quad (5.14)$$

Let $f = \lim_{\varepsilon \rightarrow 0} f_\varepsilon$ is as in (5.3) [93]:

$$f'_\varepsilon(x) = \begin{cases} 0 & x \leq -(1+\varepsilon) \\ \frac{1}{2\varepsilon}[x+(1+\varepsilon)] & |x+1| \leq \varepsilon \\ 1 & |x| \leq 1-\varepsilon \\ -\frac{1}{2\varepsilon}[x-(1+\varepsilon)] & |x-1| \leq \varepsilon \\ 0 & x \geq 1+\varepsilon \end{cases} \quad f''_\varepsilon(x) = \begin{cases} 0 & x \leq -(1+\varepsilon) \\ \frac{1}{2\varepsilon} & |x+1| \leq \varepsilon \\ 0 & |x| \leq 1-\varepsilon \\ -\frac{1}{2\varepsilon} & |x-1| \leq \varepsilon \\ 0 & x \geq 1+\varepsilon \end{cases} \quad (5.15)$$

Let us expand $\nabla^2 f(u)$:

$$\begin{aligned} \nabla^2 f(u) &= \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial u} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial u} \frac{\partial u}{\partial y} \right) = \\ &= \frac{\partial^2 f}{\partial u^2} \left(\frac{\partial u}{\partial x} \right)^2 + \frac{\partial f}{\partial u} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 f}{\partial u^2} \left(\frac{\partial u}{\partial y} \right)^2 + \frac{\partial f}{\partial u} \frac{\partial^2 u}{\partial y^2} = \\ &= \frac{\partial^2 f}{\partial u^2} \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 \right) + \frac{\partial f}{\partial u} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = \end{aligned}$$

$$= f''(u)(\|\nabla u\|^2) + f'(u)\nabla^2 u \quad (5.16)$$

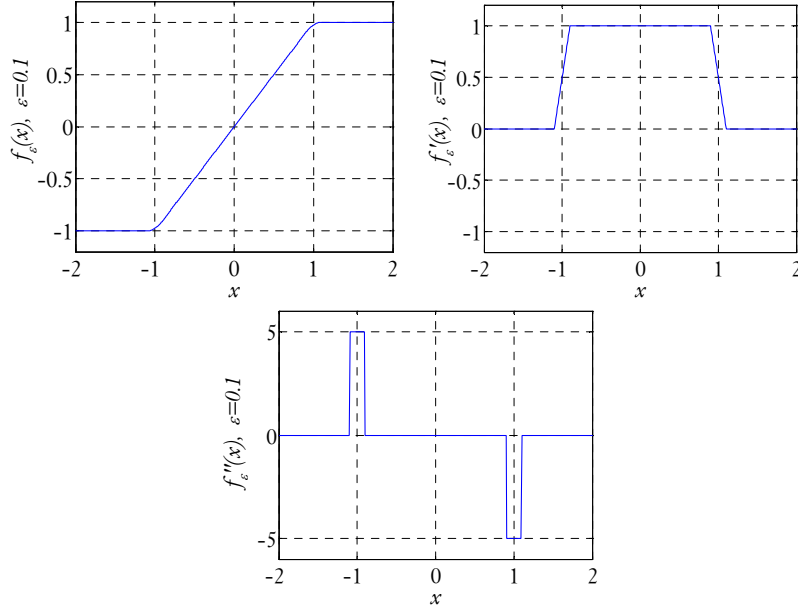


Figure 77. The $f_\varepsilon(x)$ and its first and second derivatives.

Substituting (5.16) for (5.14) we get:

$$\frac{\partial u}{\partial t} = -u + D(f''(u)\|\nabla u\|^2 + f'(u)\nabla^2 u) + a_\varepsilon f(u) \quad (5.17)$$

Observe that now we have two terms inside of the parenthesis: the first one is the product of the square of the vector norm of the gradient and of the second derivative of the nonlinearity. The second term is the product of the derivative of the nonlinearity and of the Laplacian of u .

Let $u(x,y)$ be the intensity image (*surface*), moreover, let $\Gamma: u(x,y)=c$ be an iso-intensity line (or *level contour*) of the boundary region in the spreading wave front (see Fig. 78). Let us introduce local moving orthonormal coordinate system (ξ, η) at any point P_0 in the iso-intensity lines of the boundary region. Let θ be the angle of the gradient vector:

$$\theta = \arctan\left(\frac{\partial u}{\partial \eta} / \frac{\partial u}{\partial \xi}\right) \quad (5.18)$$

The η -axis is parallel to the gradient direction at P_0 , and the ξ -axis is perpendicular, i.e.

$$\mathbf{e}_\xi = (\sin\theta, -\cos\theta)^T \quad \text{and} \quad \mathbf{e}_\eta = (\cos\theta, \sin\theta)^T = \frac{\nabla u}{\|\nabla u\|}.$$

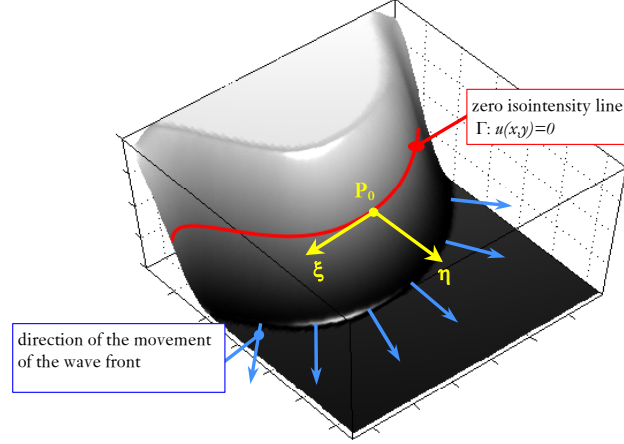


Figure 78. A boundary region and its moving local coordinate systems.

In terms of Cartesian coordinates the local directional derivatives can be expressed [33]:

$$\partial_{\xi} = \sin \theta \partial_x - \cos \theta \partial_y, \quad \partial_{\eta} = \cos \theta \partial_x + \sin \theta \partial_y. \quad (5.19)$$

In this local coordinate system one of the first-order directional derivatives, u_{ξ} is zero. Each coordinate system moves together with the propagating wave front (Fig. 78). The Laplace operator can be written as

$$\nabla^2 u = \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial \eta^2} + \frac{\partial^2 u}{\partial \xi^2}. \quad (5.20)$$

Let $u_{\eta} = \frac{\partial u}{\partial \eta}$, $u_{\eta\eta} = \frac{\partial^2 u}{\partial \eta^2}$. The second derivative along the ξ -axis can be written as

$$u_{\xi\xi} = \mathbf{H}u(\xi, \xi) = \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{\left(\sqrt{u_x^2 + u_y^2}\right)^2} \quad \text{where } \mathbf{H} \text{ is the Hessian.}$$

Since the curvature of a plane curve defined by an implicit function $\Gamma: u(x,y)=0$ is

$$\kappa = \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{(u_x^2 + u_y^2)^{3/2}}. \quad (\text{See Appendix B and Table 12})$$

We get [96]

$$u_{\xi\xi} = \kappa |\nabla u| = \kappa u_{\eta} \quad (5.21)$$

Table 12 shows different forms of operators in different coordinate systems. See tutorial web pages [164,165,166] on the relation to anisotropic diffusion, the motion of the level sets and their curvature.

Name	Cartesian	Local moving (gague)
Intensity	u	u
Gradient ²	$u_x^2 + u_y^2$	u_η^2
Laplacian	$u_{xx} + u_{yy}$	$u_{\xi\xi} + u_{\eta\eta}$
Curvature of an isointensity line	$\frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{(u_x^2 + u_y^2)^{3/2}}$	$\frac{u_{\xi\xi}}{u_\eta}$

Table 12. Different forms of two-dimensional differential invariants [33]

Using notation $u_\eta = \frac{\partial u}{\partial \eta}$, $u_{\eta\eta} = \frac{\partial^2 u}{\partial \eta^2}$ and substituting eq. (5.21) for (5.20) we get the following [96]:

$$\nabla^2 u = u_{\xi\xi} + u_{\eta\eta} = u_{\eta\eta} + \kappa u_\eta. \quad (5.22)$$

This curvature κ is the same for the surface u and for the zero isointensity curve. This is because u has a flat shape around the zero level set ($u_{\eta\eta} \approx 0$), see Fig. 78 for three-dimensional view and Fig. 73 for one-dimensional cross section of the spreading wave front.

We know that

$$\|\nabla u\|^2 = u_x^2 + u_y^2 = u_\eta^2, \quad (5.23)$$

where u_x, u_y denotes the derivative along the x and y axis.

Substituting eq. (5.22) for (5.17) and using (5.23) we get the following one-dimensional equation:

$$\frac{\partial u}{\partial t} = -u + D(f''(u)u_\eta^2 + f'(u)(u_{\eta\eta} + \kappa u_\eta)) + a_c f(u). \quad (5.24)$$

Now we see that the behaviour of the PDE depends on the local curvature κ . Inside of the linear region the influence of the curvature is dominant. Close to the edges of the linear region the importance of the curvature decreases while the square of the gradient determines the state derivative.

Let us now suppose that u is in the linear region of the nonlinearity. Inside of this region ($|u| \leq 1 - \varepsilon$) $f''(u) = 0$ and $f'(u) = 1$. Using this we get

$$\frac{\partial u}{\partial t} = -u + D(u_{\eta\eta} + \kappa u_\eta) + a_c u. \quad (5.25)$$

Let consider the case when u is zero, which means it is on the zero isointensity line. Thus we can write:

$$\boxed{\frac{\partial u}{\partial t} = Du_{\eta\eta} + D\kappa u_{\eta}}. \quad (5.26)$$

On the other hand, equation (5.26) can be also written as:

$$\frac{\partial u}{\partial t} = D(u_{\xi\xi} + u_{\eta\eta}).$$

Since $u_{\eta\eta} \approx 0$ we get

$$\boxed{\frac{\partial u}{\partial t} = Du_{\xi\xi}}. \quad (5.27)$$

The speed of a point of the isointensity line along the gradient is

$$\frac{\partial \eta}{\partial t} = \frac{\partial u / \partial t}{\partial u / \partial \eta}. \quad (5.28)$$

Thus we get

$$\frac{\partial \eta}{\partial t} = D \frac{u_{\eta\eta}}{u_{\eta}} + D\kappa.$$

In this equation $[u_{\eta\eta} / u_{\eta}]$ is a small number. As a result we see that the curve is moving according to the local curvature since η is the component of the basis vector of our local coordinate system along the gradient.

$$\frac{\partial \eta}{\partial t} = D\kappa. \quad (5.29)$$

Equation (5.27) shows that smoothing occurs along the boundary. This equation is well-known in literature and it was shown that its meaning is that the level sets of u move in the normal direction with a speed proportional to their mean curvature (Alvarez [37], Evans [41], Osher [42]). Equation (5.29) also shows this. In our case, the curvature of surface u and the embedded curve are the same when u is on the zero isointensity line. Of course, the relation (5.27) is valid only in the linear region of the CNN, however this linear region always exists during the transient if p and s are chosen according to equations (5.8) and (5.11). ■

From PDE to CNN

To obtain the CNN representation we make the spatial discretisation of (5.25) with $b=1$, $(\cdot)_{\eta} = [-1/2 \ 0 \ 1/2]$, $(\cdot)_{\eta\eta} = [1 \ -2 \ 1]$:

$$\frac{dx_i}{dt} = -x_i + \frac{D}{4} \left(x_{i-1} - 2x_i + x_{i+1} + \frac{1}{2} (\kappa x_{i-1} - \kappa x_{i+1}) \right) + a_e x_i. \quad (5.30)$$

Let $a_e = p+4s$ and $D = 4s$. Substituting this for (5.30) we get

$$\frac{dx_i}{dt} = -x_i + s \left(x_{i-1} - 2x_i + x_{i+1} + \frac{1}{2}(\kappa x_{i-1} - \kappa x_{i+1}) \right) + (p + 4s)x_i, \quad (5.31)$$

$$\boxed{\frac{dx_i}{dt} = -x_i + s \left(1 + \frac{\kappa}{2} \right) x_{i-1} + (p + 2s)x_i + s \left(1 - \frac{\kappa}{2} \right) x_{i+1}} \quad (5.32)$$

■

The Effect of κ

In the following let us investigate the effect of κ .

Case 1: $\kappa = 0$

If $\kappa = 0$ equation (5.32) is the same as equation (6) in [91] in the linear region:

$$\frac{dx_i}{dt} = -x_i + \hat{s}^+ x_{i-1} + \hat{p} x_i + \hat{s}^- x_{i+1}. \quad (5.33)$$

where $\hat{p} = p + 2s$ and $\hat{s}^+ = \hat{s}^- = s$. Thus theorems and results developed in that case can be used, too.

Case 2: $\kappa \neq 0$

If $\kappa \neq 0$ then in equation (5.33)

$$\hat{s}^+ = s \left(1 + \frac{\kappa}{2} \right) \text{ and } \hat{s}^- = s \left(1 - \frac{\kappa}{2} \right). \quad (5.34)$$

In template form:

$$\boxed{A = \begin{bmatrix} s \left(1 + \frac{\kappa}{2} \right) & p + 2s & s \left(1 - \frac{\kappa}{2} \right) \end{bmatrix}} \quad (5.35)$$

This template is asymmetric. Asymmetry depends on the curvature. The eigenvalues of this corresponding system of the linear region are [94]:

$$\lambda_i = p - 1 + 2s + 2\sqrt{\hat{s}^+ \hat{s}^-} \cos\left(\frac{i\pi}{B+1}\right). \quad (5.36)$$

Substituting (5.34) we get

$$\lambda_i = p - 1 + 2s + 2s \sqrt{1 - \frac{\kappa^2}{4}} \cos\left(\frac{i\pi}{B+1}\right) \quad (5.37)$$

In [91] Thiran proved that the exact values of the state of a CNN in equilibrium is

$$\text{if } B \text{ is even} \quad x_i^* = (-1)^i \frac{\cos((B+1-2i)\phi/2)}{\cos((B+1)\phi/2)}, \quad (5.38)$$

$$\text{if } B \text{ is odd} \quad x_i^* = (-1)^i \frac{\sin((B+1-2i)\phi/2)}{\sin((B+1)\phi/2)}. \quad (5.39)$$

where ϕ is $\arccos\left(\frac{p-1+2s}{2s}\right)$, B is the number of linear cells provided that the template describing the system is symmetric. If we introduced $\varepsilon = \frac{\kappa}{2}$ ($\varepsilon > 0$) small perturbation of the template then the equilibrium would change:

$$\phi = \arccos\left(\frac{p-1+2s}{2s\sqrt{1-\varepsilon^2}}\right).$$

Therefore the equilibrium x_i^* should change. The greater κ the greater $|x_i^*|$ is.

$$\kappa > 0 \rightarrow \dot{x}_i > 0$$

$$\kappa < 0 \rightarrow \dot{x}_i < 0$$

It follows from the definition that small perturbation (curvature) does not necessarily change B , however it does change x_i^* and consequently the location of the boundary region.

Corollary

- a) If the curvature is zero then the 2D problem is reduced into orientation-independent 1D problem and stability results developed in the 1D case are valid.
- b) Moreover, the result strongly conjectures that if the curvature is not zero the boundary region will move according to the sign and value of the curvature. Experiments support the conjecture.

5.3.1 Examples

Disc deformation

In this experiment I measured the change of state derivative x of the CNN during the evolution. The initial state was a black disc. During the transient the disc was continuously decreasing with increasing speed until it disappeared. Meanwhile, the magnitude of the state derivative in the linear transition zone got larger and larger. At the same time the real curvature of the disc was computed according to the relation $\kappa=1/R_t$, where R_t is the actual radius of the disc. Results are shown in Fig. 79,80 and Fig. 81.

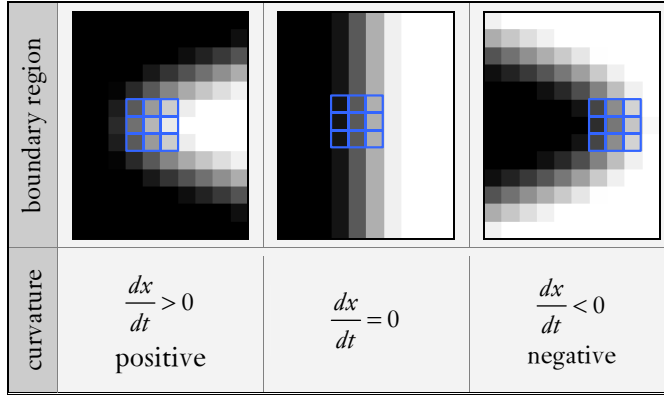


Figure 79. The interpretation and the sign of the local curvature of a boundary zone.

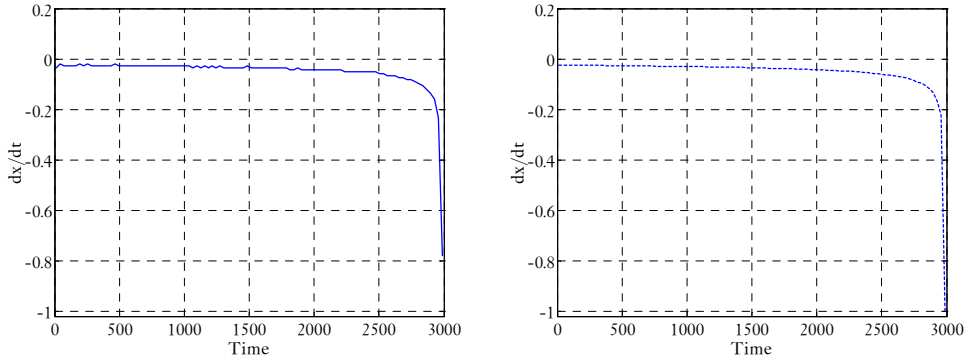


Figure 80. The real curvature= $1/R_t$ (solid) and the state derivative of the CNN (dashed). ($p=-1.25, s=0.6, r=0$)

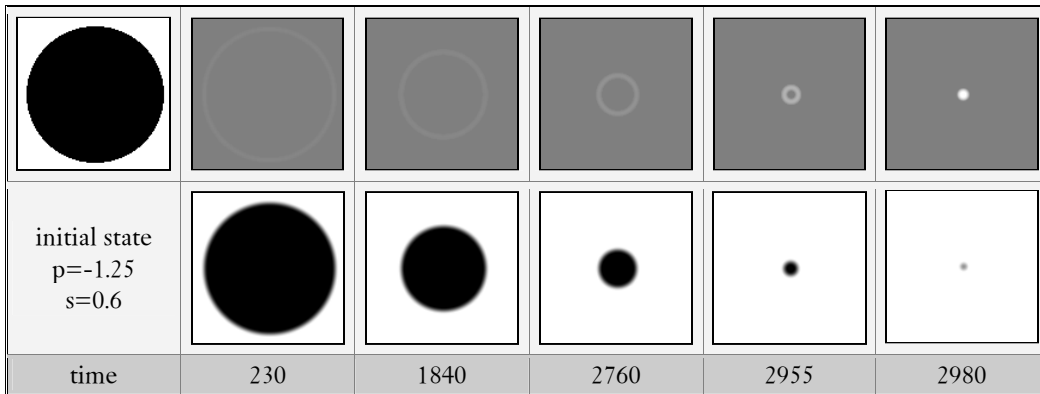


Figure 81. Measurement of the curvature. The first row contains the state derivative. The gray level encodes the curvature. White means negative curvature. ($p=-1.25, s=0.6, r=0$, image size: 128×128)

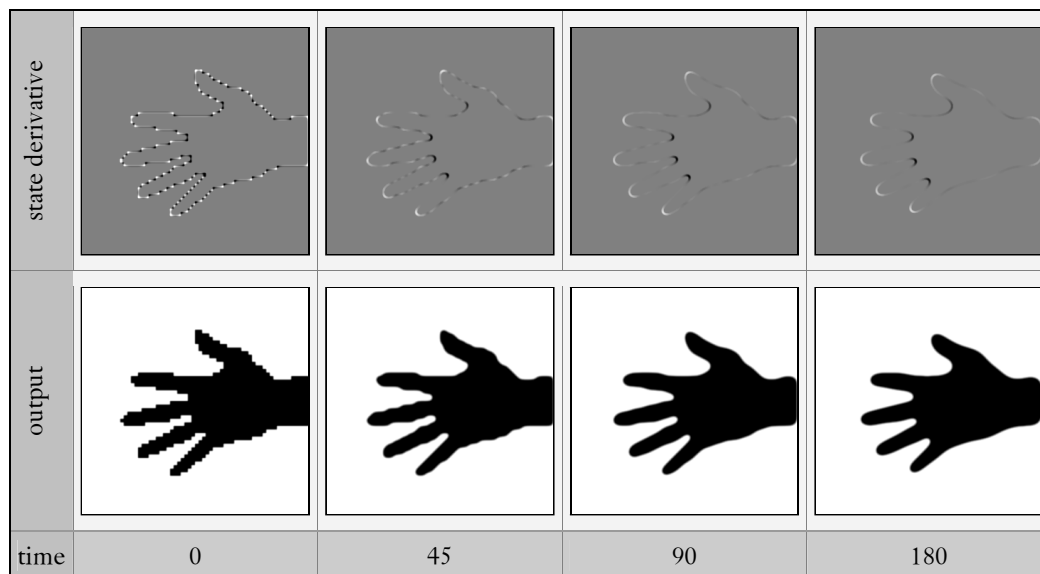


Figure 82. Curvature smoothing. Upper row shows the snapshots of the state derivative. The bottom row shows the output of the CNN. ($p=-1.1, s=0.6, r=0$, image size: 320×320 , original size: 64×64)

Complex shape deformation

In this experiment simulation shows the evolution of a human hand that is enlarged from a smaller image without smoothing. Thus the enlarged image is very rough in spite of high resolution. Snapshots in Fig. 82 show that the contour of the hand is smoothed. However, the shape is continuously distorted meanwhile. The same evolution is shown in a more complex scene in Fig. 83.

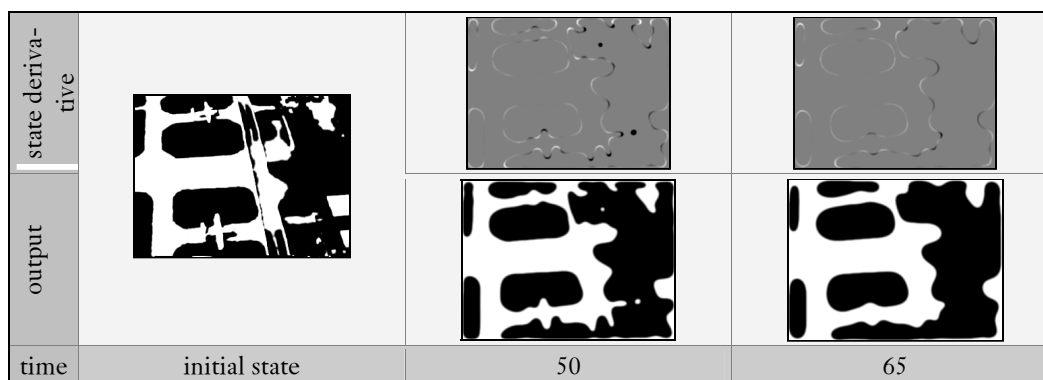


Figure 83. Curvature smoothing in complex scene. Upper row shows the snapshots of the state derivative. The bottom row shows the output of the CNN. ($p=-1.1, s=0.6, r=0$, image size: 320×240)

5.3.2 Connection to bipolar waves

In [85] the bipolar wave was introduced: two trigger waves are propagating in the same medium. The initial condition is special: it is zero except for few pixels. The zero initial state for the cell is the origin - an unstable equilibrium. This makes it possible to go to either stable equilibrium. While opposite-signed cells are separated by zero cells the two types of trigger waves are spreading with the same speed and without influencing each other. When the black and white trigger waves collide the evolution continues according to the local curvature as derived above. This co-evolution can be controlled by template parameter p . If $p > 1$ there is no other stable equilibrium only in the saturation regions of the nonlinear transfer function [46]. It means that the number of linear cells is zero and the wave front freezes after a transient. When $p < 0$ the wave front moves if the number of linear cells B is 'large enough' ≈ 3 . Experiments show that when B is equal or greater than three the evolution is smooth.

These can be used for computation, e.g. to find the contour that is equidistant from the black and white patches (see Fig. 84).

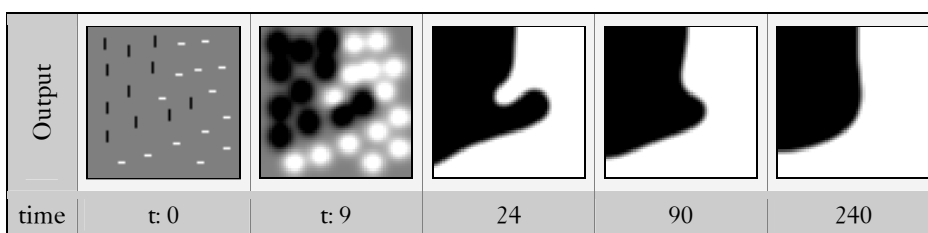


Figure 84. $r=0, s=0.6, p=-1$, The boundary condition is zeroflux

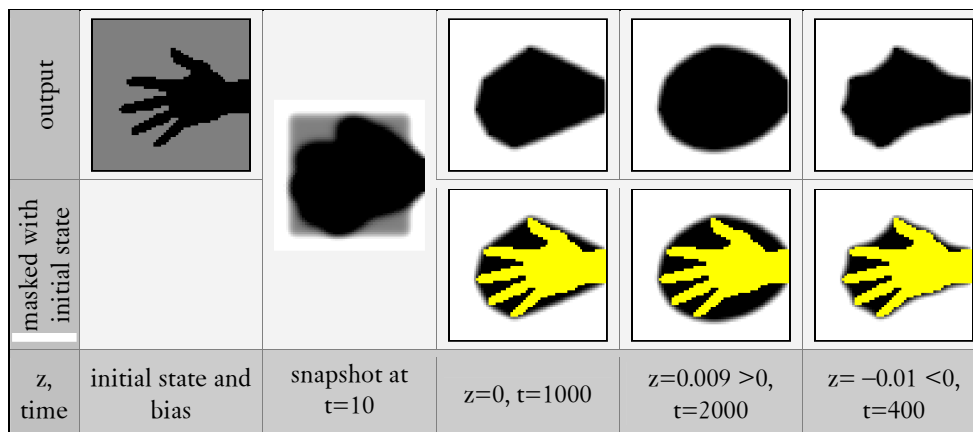


Figure 85. Effect of input on the curvature. ($p=-1, s=0.6, r=0, b_0=1$, image size: 64×64)

Example

Let us consider the following equation:

$$\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{k=-r}^r \sum_{l=-r}^r A_{kl} f(x_{i+k, j+l}(t)) + z_{ij}$$

Now we introduced a constant term: z_{ij} . Its effect is that the contour is not stationary when its curvature is zero but when it has either a certain negative or positive curvature (see Fig. 85).

If z_{ij} is greater than zero contours with positive curvature are stationary, if it is less than zero contours with negative curvature are stationary. The bias map z_{ij} is chosen so that the cells belonging to the original shape are also stable i.e. they are in the black saturation region.

6. *Practical wave computing and experiments*

6.1 Wave computing

In section 6.1.1 a binary propagating wave is introduced, which propagates along a predefined direction and the propagation stops when a certain kind of partial convex hull of the object is filled.

In section 6.1.2 a so called bipolar wave is described that propagates symmetrically, but *black* and *white waves* spread simultaneously. When two differently colored waves collide the propagation stops, a boundary between the black and white region is formed of which characteristics was analyzed in Chapter 5.

An algorithmic frame is described in 6.1.3. It lists the basic steps of an analogic algorithm for concavity based object decomposition. The steps are composed of template and logic operations. Through decomposition recognition is accomplished since the result of each decomposition block is an image which is empty (white) or not (contains black pixels). Based on this a binary decision tree can be built of which leaves contain the detected object classes.

In section 6.2 a mobile robotic environment is introduced in which the wave based computing forms an algorithmic block. Finally in section 6.3 active wave experiments are shown that were measured on a CNUM with second order core cell. Basic active wave types are shown such as trigger, traveling, target and spiral waves.

6.1.1 Direction constrained wave

Trigger waves, which have symmetric generator template matrix, were discussed in detail in [64]. If symmetry is broken a direction selective trigger wave generator yields. The propagation starts from those black pixels around which there is a properly oriented, L-shaped concave pixel configuration (see Fig. 86).

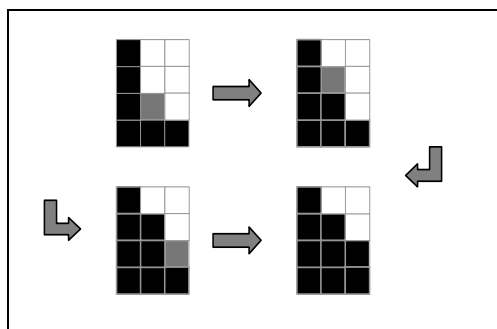


Figure 86. The result of the concavity filler template in a 3x4 sized window. The stable pixels are denoted by black. Gray color denotes those pixels that are to be turned into black in the next step. Propagation occurs if there is a properly oriented, L shaped pixel configuration around a white pixel.

Expressed as pixel level rule: those white pixels which have black neighbours to north-west, south-west and south will be black. Other directions are obtained with rotation and mirroring. The result of a concavity filler template can be seen in Fig. 86 and Fig. 87.

As a result, eight different templates can be produced. The templates are characterised by the angle (α) of the direction of the propagation. Possible α values are: $\alpha = \arctan(\pm 0.5) + k2\pi$, $\alpha = \arctan(\pm 2) + k2\pi$, $k=0..1$.

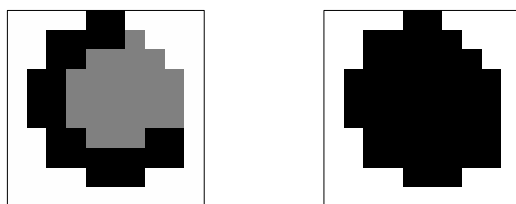


Figure 87. The result of the concavity filler template in a 10x 10 sized window, $\alpha=26.56^\circ$. Gray color denotes those pixels that are to be turned into black.

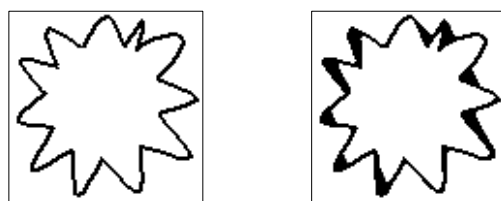


Figure 88. The Result of the concavity filler template., $\alpha=26.56^\circ$

The following template generates propagation for the $\alpha=26.56^\circ$ ($\arctan -0.5$) direction:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad z = 2 \quad (6.1)$$

This type of wave can fill concave segments of the object depending on the orientation of the concavity. The effect of the template can be well observed in Fig. 88. The concavity template (which can be found in the CNN Software Library [62]) produces similar, but direction independent result.

6.1.2 Bipolar waves

All cells of a trigger wave CNN - except the cells changing in the wave front - are in one of the stable states: they are either black or white. However, there is a third, though unstable state - the zero level or origin -, which can be applied in computation. Setting the initial state of a CNN to zero except for few pixels, two different waves can be initiated in the same structure: black waves starting from black pixels and white waves triggered by white pixels. Other empty areas are set to zero. When two identically coloured waves collide, they join. But when two different ones collide, the propagation stops and a boundary region is formed between them. See Fig 89.

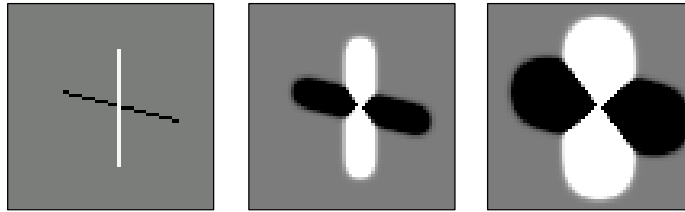


Figure 89. Transients of bipolar waves. When two different waves collide, annihilation occurs.

Using the theory developed in Chapter 5 and choosing the number of liner cells between the black and white regions to be zero, one possible template for this is:

$$\mathbf{A} = \begin{bmatrix} 0 & 0.6 & 0 \\ 0.6 & 0.8 & 0.6 \\ 0 & 0.6 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad z = 0 \quad (6.2)$$

Since the speed of propagation of the black and the white waves are the same, the stopping will occur half way between two different initiator patches. The boundary will be a line (Fig. 90).

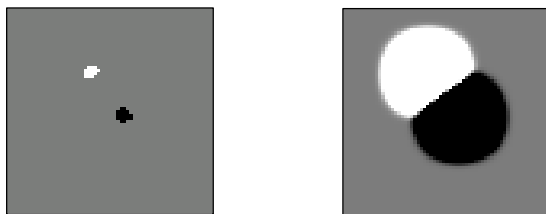


Figure 90. Waves of two patches. The annihilation zone divides the distance of the initial patches into two equal parts.

If more than two different patches are present, the boundary region will consist of line segments (Fig. 91). This may be exploited in clustering of point sets.

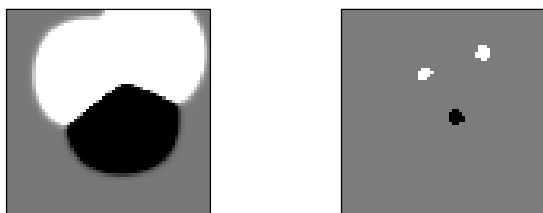


Figure 91. Bipolar waves when three points are present. The annihilation zone consists of two line segments.

6.1.3 Curvature and concavity based object decomposition

One of the characteristic features which human recognition seems to be based on is the concavities and arcs of objects. Several objects can be well described by the positions and relations of their concavity locations.

During his experiment, Fujita [126] found that in monkeys' inferotemporal cortex are neurons which are sensitive exclusively to specific complex shapes and patterns. This recognition is not a kind of template matching, but a much more robust process which can tolerate a wide range of changes of illumination and viewing angle of objects.

The CNN operation presented in section 6.1.1 is suitable for detection of differently oriented arc concavities. Since the concavity is a scale invariant property, it can be an effective descriptor of objects.

Here a CNN algorithm frame for object classification is presented, where feature-based decomposition is applied first and then the resulting images are filtered by logic operations.

1. Apply the direction selective concavity filler template to the initial black and white images. Do this for all desired directions of arcs by applying the appropriately transformed template.
2. Subtract the original image from all of the result images and remove small patches and single pixels.

3. Form logic combinations of the result images of the previous step, which contain at this point only patches at the locations of selected concavities. By logic combination, direction selectivity can be improved.
4. Classify the patches by distance. As a result, we get images on which patches are left that have specified distance and orientation compared to each other.
5. Make logic combination of the result images of the previous step.
6. Compose logic OR of each image. The result is a binary feature vector.

Steps 4 and 5 are not always necessary. The distance classification can be accomplished by applying the variants of shadow templates [62] for projecting shadows of prescribed length into appropriate directions. If two images are given that contain patches, let the transient of the shadow template run until it reaches the desired length in the first image. Then we take the logic AND of the two images. The result contains patches which fall into the selected direction and are not farther from the patches in the first image than the length of the projected shadow.

This algorithm skeleton is used for some applications which are presented in the following section.

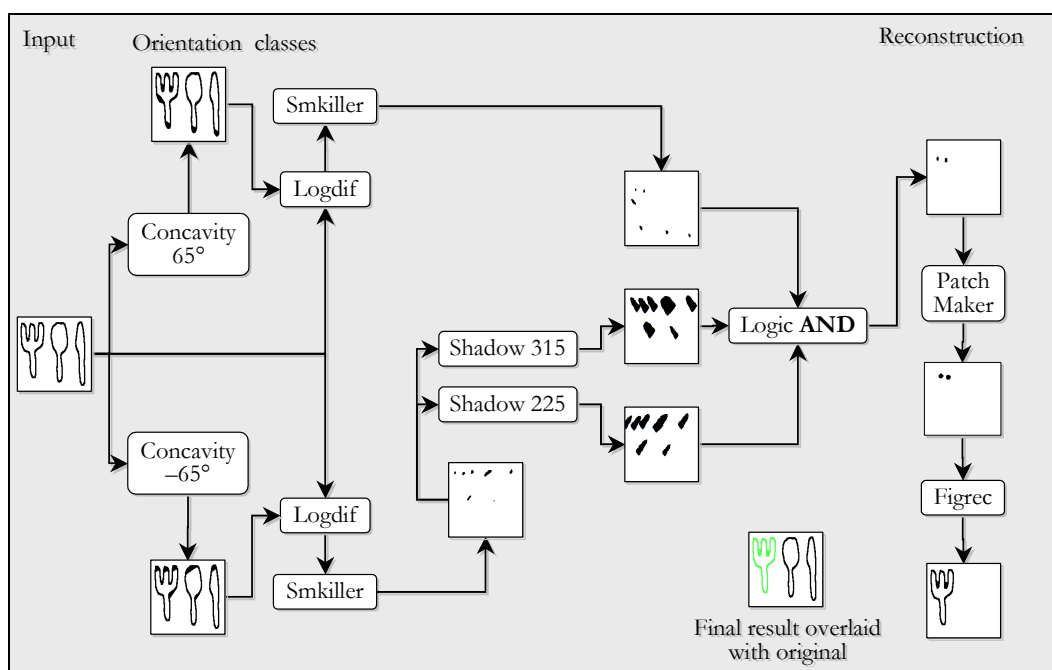


Figure 92. Fork detection using the algorithm frame described formerly.

6.1.4 Fork detection

Using the algorithm frame detailed above I present a simple example where the task is to detect the fork in the picture. We have to find characteristic features of the object then to

extract them. Such features are concavities or endings and their relative position. Some post- and intermediate processing is needed such as small object removing.

The solution starts with applying two direction-selective concavity filters (Step 1). Thus we can fill in the tip and the bifurcations of the fork (see Fig. 92). Next taking the difference of the result and the original we get the locations of the tip and the bifurcation points (step 2).

Following templates were used:

$$\text{Concavity 65: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, z = 2$$

$$\text{Concavity -65: } \mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, z = 2$$

$$\text{Shadow 315: } \mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, z = 2.5$$

$$\text{Shadow 225: } \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 2 & 1 \\ -1 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, z = 2.5$$

The *Figrec* template reconstructs the image starting from one point. The *Smkiller* template removes the small, few pixels wide objects. The *Figrec*, *Smkiller* and the *Patchmaker* templates are available in the CNN software library [62]. The *Shadow** templates result in directed shadows originating from concave locations.

6.1.5 Hand orientation detection

This algorithm shares the basic principle described in 6.1.3. Supposing a suitably preprocessed image the algorithm starts with a thresholding. Next the difference of the original and the intermediate results are taken and artifacts are removed. Finally, the nonempty image out of the three intermediate results denotes the detected orientation class.

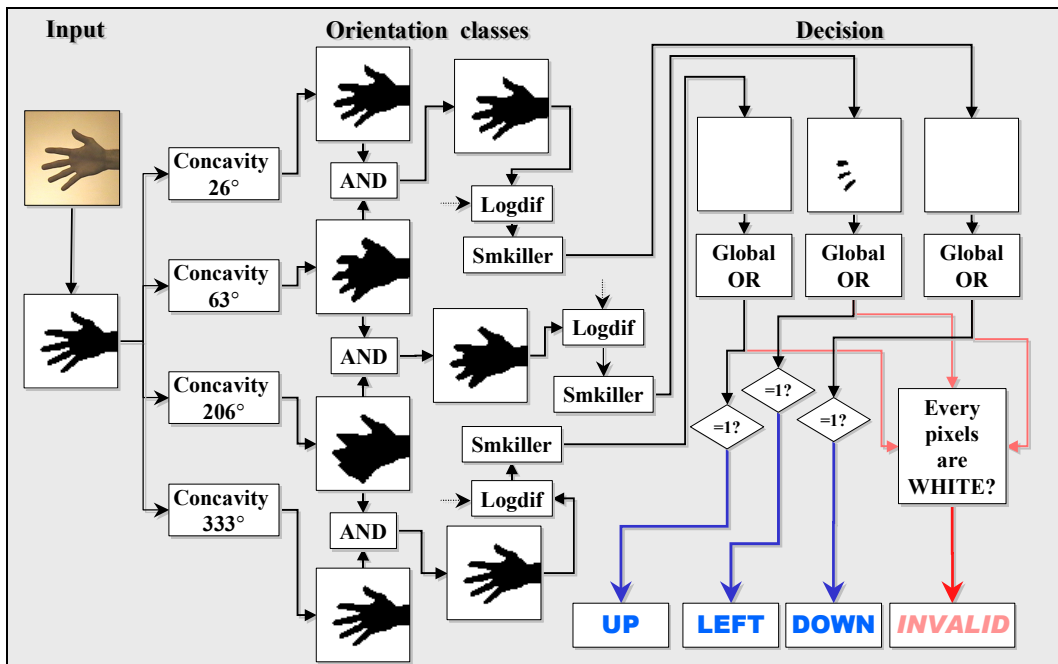


Figure 93. Hand orientation detection by the 64x64 CNNUM chip. The dotted input line of Logdif symbolises the signal source from the black and white input images. The text boxes contain the names of the applied templates and logic operations.

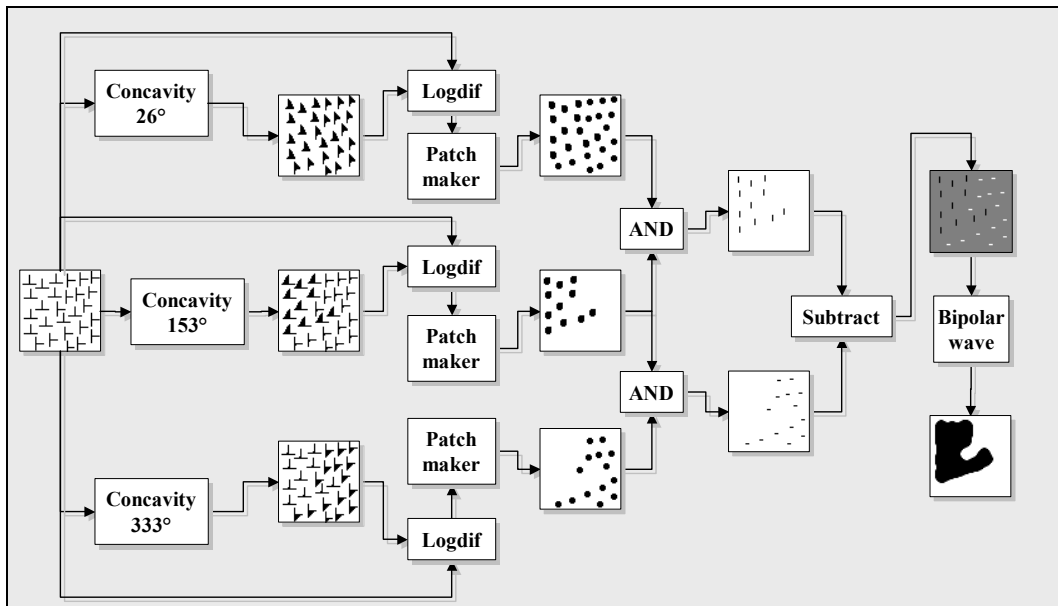


Figure 94. Texton segmentation. The text boxes contain the names of the applied templates and logic operations. The final image contains the regions.

6.1.6 Texton segmentation

This algorithm is also built according to the steps listed in 6.1.3. However, as a final step segmentation block is added. The segmentation applies bipolar waves described in 6.1.2. The result of computation is the steady state of the propagation.

6.2 Mobile navigation unit

The templates and subroutines described previously constitute a basic image processing tool-set. In the following I introduce a mobile surveillance robot (see Fig. 95) of which control algorithm integrates the aforementioned tool-set.

The aim was to design and produce a mobile platform-based system that is able to identify objects or events by simple features (shape, texture, motion, etc.) and to approach or navigate to them. Moreover, it is able to gather information and transmit information about them to the base station. The mobile unit communicates with the base station through a wireless channel. The base station monitors the mobile unit of which control can be taken over at any time.

The robotic system is equipped with several sensors, operates on battery and controlled by the Bi-i system introduced in chapter 3 on page 18. The robot itself is an off-the-shelf mobile platform fabricated by K-Team from Switzerland.

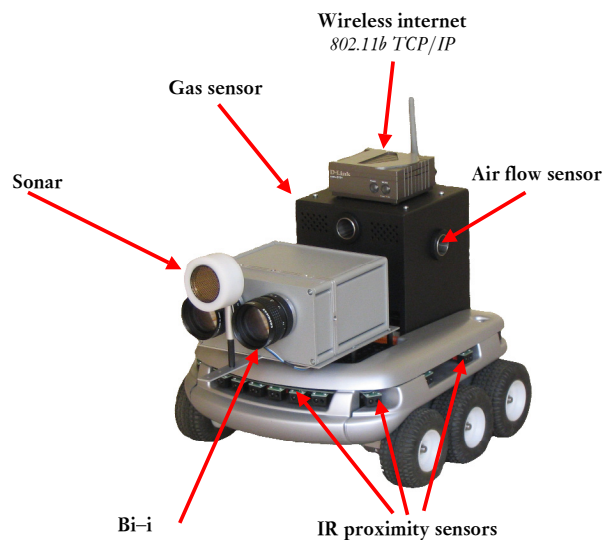


Figure 95. Mobile unit overview.

6.2.1 Hardware infrastructure

The hardware of the system is the result of collaboration of several partners of the project. The Bi-i system was designed and produced by the Analogic Computers Ltd [108]. The airflow and gas sensors were produced by MTA-KFKI-MFA research institute (see Fig.

95). The high-level algorithm frame for controlling the mobile unit was designed at MTA-SZTAKI, in which the author took part.

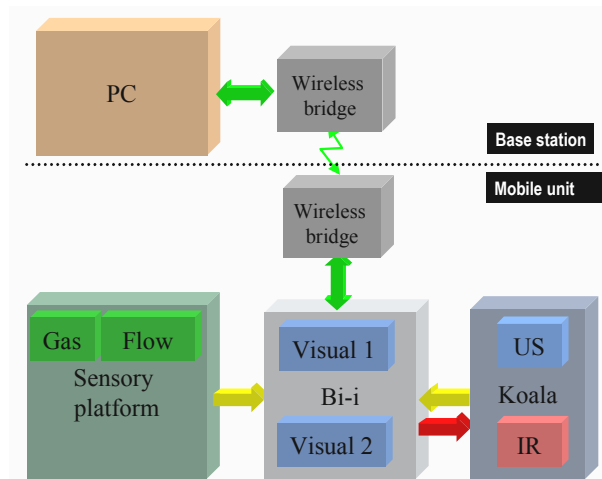


Figure 96. Block diagram of the mobile system.

The block diagram of the system is shown in Fig. 96. Main components are the base station that consists of a PC with a wireless bridge and the mobile unit. The mobile unit is built up of a mobile platform (Koala), the Bi-i system and the sensory platform. The Koala is a processor controlled mobile platform equipped with IR proximity sensors and sonar (US) that controls the low-level navigation of the car. The Sensory platform hosts the gas and airflow sensors.



Figure 97. Real - virtual environment



Figure 98. Robot in the test world

The mobile unit is tested in a closed test-world (Fig. 97 and 98) in which different objects or places should be recognized/found by the fused infrared, ultrasound, gas and visual sensory information. Future developments include the virtual representation of the model world and the car. We developed an integrated control environment that provides visual, literal and/or numerical feedback from the robot and allows the user to change parameters, strategy and operation modes of the mobile unit and to interact immediately if necessary (see Fig. 99).

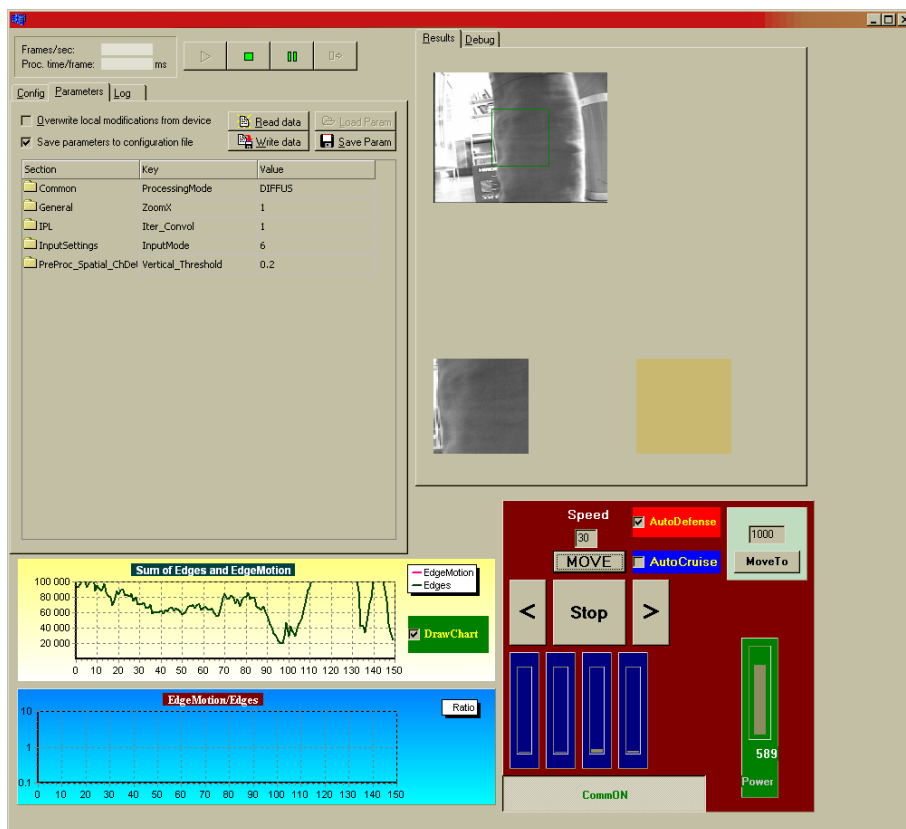


Figure 99. Integrated control environment of the mobile unit.

6.2.2 Software infrastructure

The Bi-i system is responsible for the high-level control. It contains three dedicated processors (CNN-UM, DSP -digital signal processor, Communication processor). The processor of the robot car is responsible for the control of the motors. It computes a low-level obstacle avoidance algorithm that is modulated by the Bi-i system. The car receives direction and speed commands from the Bi-i that computes the so-called hybrid strategy based mobile robot control algorithm. The main emphasis is put on the visual processing that is the most computationally intensive part of the control architecture (Fig. 100).

The computationally intensive parts of the image processing algorithms run on the CNN-UM. This includes the topographic preprocessing of images (feature and motion extraction and the fusion of them). The DSP controls the CNN-UM, gathers measurements both from the sensory platform and from the robot car. The car provides – if necessary – IR, sonar measurements and homeostatic sensory information. The DSP computes some part of motion analysis and performs post processing necessary for the categorization of information. The other modules that run on DSP update the *world model* – a global memory of the space discovered –, determine the goal to be reached (*planner module*) and perform the integration of the different sensory modalities so as to provide a suitable input to the *navigation module*, that selects the appropriate behaviour (movement) to reach the goal.

In the following I introduce some of the modules that make use of CNN technology.

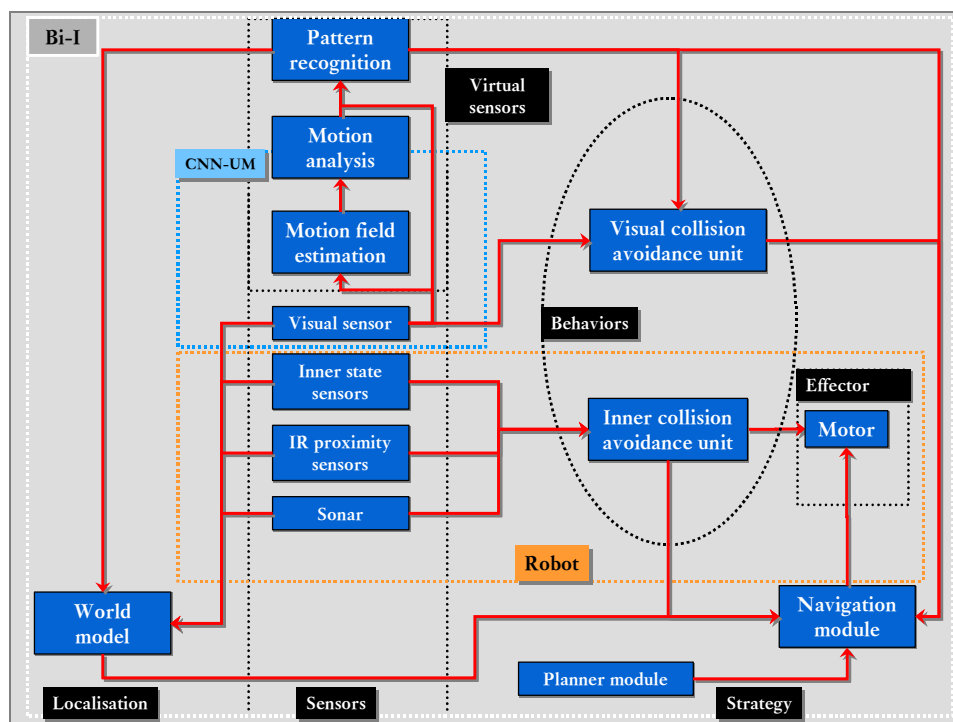


Figure 100. Mobile unit control architecture.

6.2.3 Optic flow estimation

Optic flow calculation is a rather computationally intensive step of image processing. Still, its use cannot be omitted because it is one of the key steps of spatio-temporal motion processing. Algorithms found in literature [127] usually include some kind of computationally intensive method e.g. block matching techniques, iterative minimization methods or matrix inversion. However, using simple parallel near-sensory image processing techniques a very fast approximate motion field can be extracted. The result of this method can be used as a first guess for more sophisticated methods or, if the precision is enough, it can be the starting point for segmentation and recognition tasks.

Let us consider a moving object. The key element of the algorithm is a fading memory that serves as storage for motion history. The movements of selected features are stored in it such a way that the feature map is iteratively added to the memory. Adding is followed by a thresholding to keep the values bounded. The detected feature is represented in the map with black pixels (maximum intensity level). Meanwhile, the whole memory content is leaking i.e. the intensity of each pixel is gradually decreasing. As an object moves it is added to the memory in a little shifted position depending on its speed. Due to the fading property of the memory a motion history is produced resembling a comet that draws a tail. The head is the actual position of the object feature having maximum intensity; the tail preserves the previous positions with decreasing intensity. The direction of the gradient on this tail gives the direction of the motion and its magnitude is proportional to the inverse of the magnitude of the speed of the object. As a result, a vector field of the moving features (objects) is produced. It is supposed that the feature map is not “too dense”. This supposition is necessary to have enough space for the motion history.

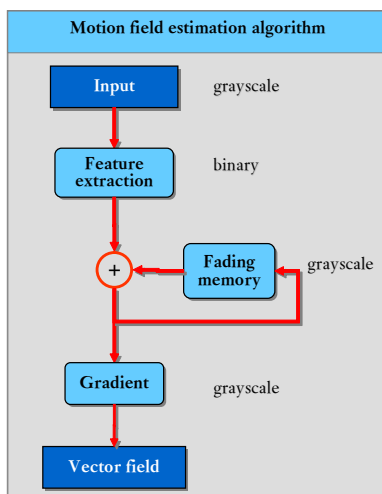


Figure 101. The block diagram of the motion field estimation algorithm.

The faster the object the longer the tail is. The slower the object the higher the gradient is on the tail.

Using several feature maps and fading memories effective motion analyzer algorithms can be built due to the inherent parallel nature of the method proposed.

Due to this it is a completely focal-plane array feasible solution. Electronically the image memory can be represented with (fading) capacitors that are charged to maximum value when the corresponding pixel is black. The feature map can be e.g. the time derivative of the input, of which computation is easy by a hardware. Additional morphological operations such as dilation, or erosion is applied to smooth the feature map.



Figure 102. The original snapshots

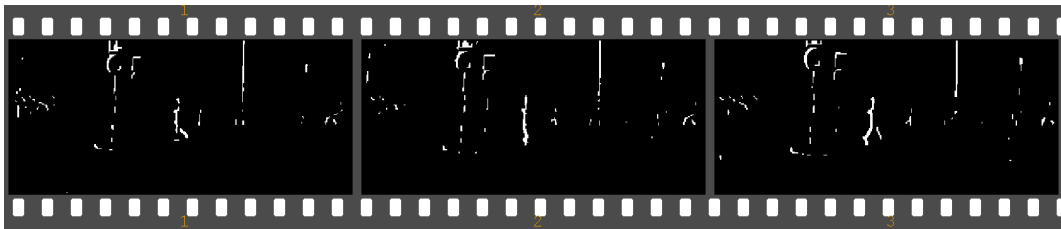


Figure 103. The feature map. The image feature is the thresholded temporal derivative of the image flow.

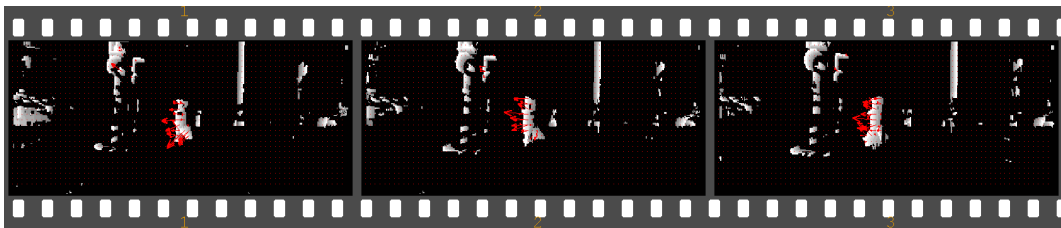


Figure 104. Fading memory: head (white patches) and tail (gray parts) of the motion history map.

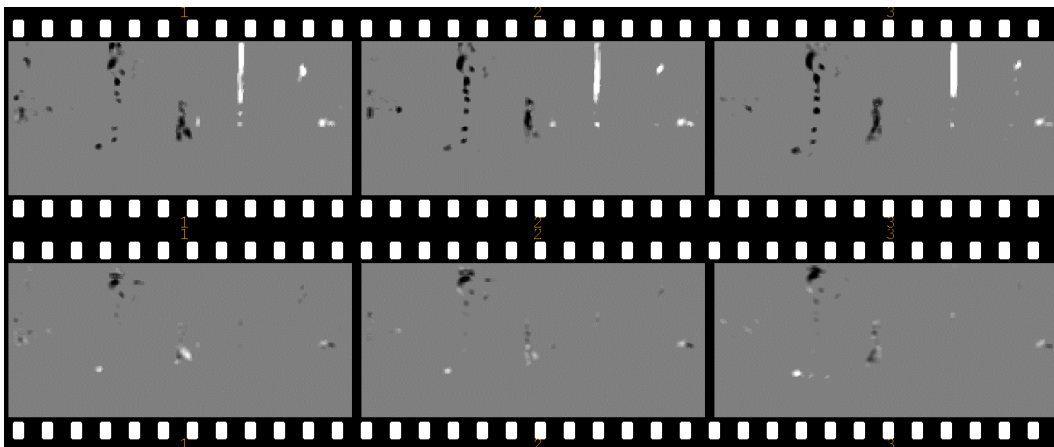


Figure 105. Gradient map. Upper film shows the gradient along x direction, the lower one shows along y direction. White color means positive gradient component, black denotes negative.



Figure 106. Final output. The vector field is overlaid onto the original image flow.

6.2.4 Visual collision avoidance

The algorithm is based on the fact that the obstacle parameters such as area of the object, size of the object, edge quantity, edge speed, etc. change faster as the observer gets more and more closer to the object provided that the observer has constant speed. If the speed of change reaches a prescribed limit we declare a threat of collision and send a signal to the navigation unit.

Figure 107 shows the block diagram of the algorithm. The input frames are grayscale images. Let us consider the last four frames of the image flow. Difference is computed between Frame_t and Frame_{t-1} , Frame_{t-1} and Frame_{t-2} , Frame_{t-2} and Frame_{t-3} . Then the absolute value of the frames are taken. Thus we get three images that are mostly zero and contain only black pixels where change occurred. On the first two frames a propagating template is computed the result of which can predict the possible change of the image.

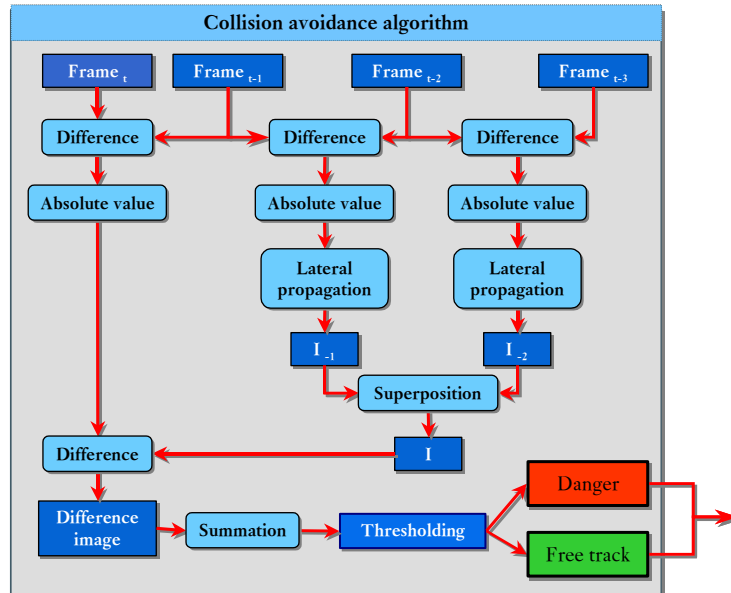


Figure 107. Visual TTC (time to collision) based analogic algorithm.

These two images are added topographically to filter the noise and fluctuations of the frames. The difference between predicted and the last absolute difference image gives a good measure of the speed of change of the edges. The sum of the pixels in the *Difference image* gives a number. Based on this number we can set a threshold above which the system declares *danger* in which case the car should immediately stop.

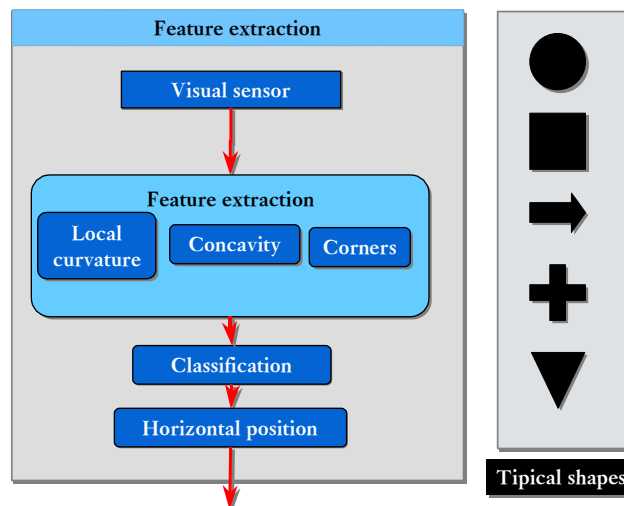


Figure 108. Pattern recognition module.

6.2.5 Feature detection

This algorithm makes use of the methods described in section 5.3.1 and 6.1. The consecutive image frames are filtered and thresholded. Then an image is produced for each local feature (curvature, concavity, corners, etc.) that is extracted and bound to the shape they belong to (Fig. 108).

The horizontal position of the detected shape gives the direction information to the navigation module. This information is also integrated to the world model.

6.3 Complex spatio-temporal wave experiments with CNN

The Complex Cell CNN programmable array computer [123,124] is an extension of the CNN Universal Machine [46–49]. The cells are organized into a 32×32 square grid. Each cell has second order dynamics and local interconnections to its neighbors. This structure is especially suitable for computing a certain set of ordinal differential equations. Simple PDEs can be transformed so that they can be easily programmed on the array computer. With the programmable Complex Cell CNN Universal Machine scientists are provided with a unique opportunity to study active-wave propagation and other reaction-diffusion differential equation based phenomena in a programmable manner. Exploiting the inherent programmability of the CNN architecture, complex wave-computing analogic algorithms can be designed.

In the following I describe the mathematical model of the core of the complex cell chip. Section 6.3.3 contains the real-time chip measurements. We should emphasize that no simulation results are presented here. This chip was designed also for computing biologically inspired retina modelling.

The architecture of the chip follows the design of the first order CNN-UM chip [98], but its first order cell core is replaced by a second order one [123]. With this change we are able to reproduce basic autowave phenomena. Fig. 109 shows the functional architecture of the second order, two-layer CNN.

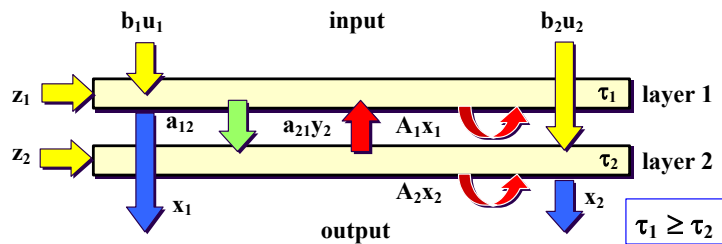


Figure 109. Functional diagram of the CACE1k chip

6.3.1 Mathematical model

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

$$\begin{aligned}
 \tau_1 \frac{dx_{1,ij}(t)}{dt} &= -g(x_{1,ij}(t)) + \sum_{k \in N_1} A_{1,k} x_{1,k}(t) + a_{12} x_{2,ij}(t) + b_1 u_{ij} + z_{1,ij} \\
 \tau_2 \frac{dx_{2,ij}(t)}{dt} &= -g(x_{2,ij}(t)) + \sum_{k \in N_2} A_{2,k} x_{2,k}(t) + a_{21} x_{1,ij}(t) + b_2 u_{ij} + z_{2,ij} \\
 1 \leq i \leq M, 1 \leq j \leq M, 0 \leq |u_{ij}| \leq 1, 0 \leq |x_{ij}| \leq 1
 \end{aligned} \tag{6.3}$$

Where M and N denote the vertical and horizontal dimensions of the array respectively.

$$g(x_{n,ij}) = \lim_{m \rightarrow \infty} \begin{cases} m(x_{n,ij} - 1) + 1 & \text{if } x_{n,ij} > 1 \\ x_{n,ij} & \text{if } |x_{n,ij}| \leq 1 \\ m(x_{n,ij} + 1) - 1 & \text{if } x_{n,ij} < -1 \end{cases} \tag{6.4}$$

Function $g(\cdot)$ regulates the state so that the state stays within the $+1..-1$ interval. Variables u_1 and u_2 are the independent inputs, b_1 and b_2 are their weight factors; $z_{1,ij}$ and $z_{2,ij}$ are space variant bias maps. Variables x_1 and x_2 denote the state variables of the layers. Each x_{ij} corresponds to one cell; it is one pixel of an M by N image if we consider the system's output as a picture. A_1, A_2 are the weights of the intra-layer couplings, a_1, a_2 are the inter-layer weights; τ_1 and τ_2 are the time constants of the two layers. Equation (6.3) utilizes the so-called *full-range model* (FSR)[125], where the voltage of the state variable is always the same as the output.

6.3.2 Stored programmability

The CACE1k chip can be programmed using the so-called AMC (ANALOGIC MACHINE CODE) language. This is similar to the assembler language but it contains additional high-level instructions, such as: image loading, frame grabber instructions, etc. and built in image processing functions. In the Appendix is a simple AMC program which computes the snapshots of the autowave example. The actual computation is performed in line 40. Before this line is the preparation for computation: setting up the parameters. After the computation is done, the results are read out and displayed from the local analog memories of the chip.

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

$$\begin{aligned}
 \mathbf{A}_1 &= \begin{bmatrix} a_{-1,-1}^1 & a_{-1,0}^1 & a_{-1,1}^1 \\ a_{0,-1}^1 & a_{0,0}^1 & a_{0,1}^1 \\ a_{1,-1}^1 & a_{1,0}^1 & a_{1,1}^1 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} a_{-1,-1}^2 & a_{-1,0}^2 & a_{-1,1}^2 \\ a_{0,-1}^2 & a_{0,0}^2 & a_{0,1}^2 \\ a_{1,-1}^2 & a_{1,0}^2 & a_{1,1}^2 \end{bmatrix} & a_{21} & a_{12} \\
 b_1 & b_2 & z_1 & z_2 & \tau_1 : \tau_2 = \tau : 1, & \tau = 1..16
 \end{aligned} \tag{6.5}$$

The operation of the array computer is completely determined by the 25 template values, the initial states and boundary conditions. In (6.5) \mathbf{A}_1 and \mathbf{A}_2 matrices denote the weights of the inter-layer connections of the slower and the faster layers. The strength of the influence of the second layer on the first is controlled by a_{21} ; a_{12} stands for the reverse case. Symbols b_1, b_2, z_1 and z_2 are the weights of the independent inputs and the space variant bias maps. The ratio of the time constants of the two CNN layers are controlled by $\tau_1; \tau_2$ is fixed. An analogic algorithm is made up of combinations of template executions,

logic instructions and spatial arithmetic operations. The example shown in the Appendix is a very simple one, it contains only template execution operation.

6.3.3 Wave phenomenon

This subsection contains the on-chip measurements of the active wave phenomena. I present chip measurement of the different nonlinear wave phenomena: trigger wave, traveling wave, autowave, and spatio-temporal edge detection.

Trigger wave

This example shows a very simple effect: an active (nonlinear) wave in the faster layer initiates a second wave in the slower layer through the positive inter-layer coupling. The input is fed to the faster layer. Fig. 110 shows snapshots of the wave evolution.

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} \quad a_{21} = 3.3 \quad a_{12} = 0 \quad (6.6)$$

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

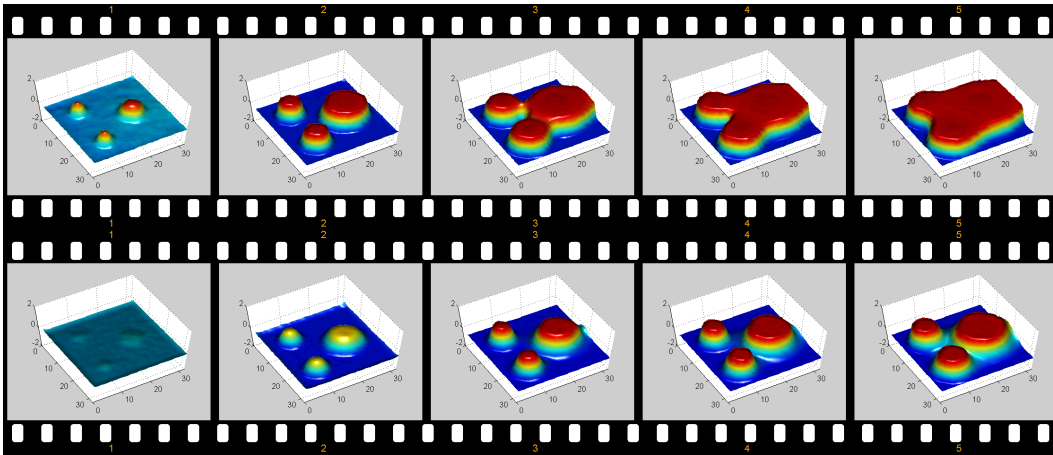


Figure 110. Trigger wave propagation in the faster and in the slower layer. The first film shows the output of the faster layer.

Erasure effect

This example is an “enhancement” of the trigger wave example: a negative coupling is introduced from the slower layer back to the faster one. As a result, after a while, the trigger wave on the faster layer is erased by the trigger wave from the slower layer. As in case of the trigger wave the input is given to the faster layer. Fig. 111 shows the snapshots of the wave evolution on the faster layer and Fig. 112 shows the evolution on the slower layer.

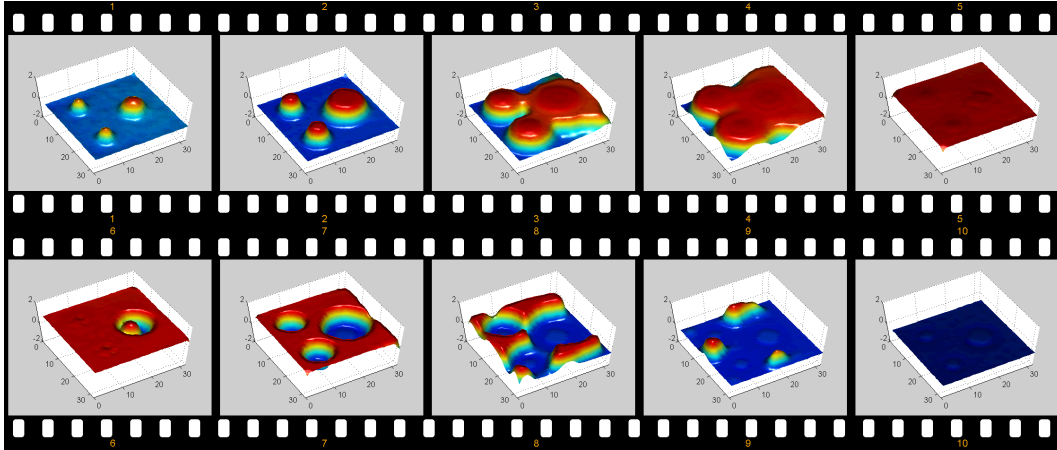


Figure 111. Retina-like wide-field erasure effect, faster layer.

$$\begin{aligned}
 \mathbf{A}_1 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 3.3 & a_{12} &= -5.4 & (6.7) \\
 b_1 &= 0 & b_2 &= 3 & z_1 &= 0.6 & z_2 &= -2.7 & \tau_1 : \tau_2 &= 16 : 1
 \end{aligned}$$

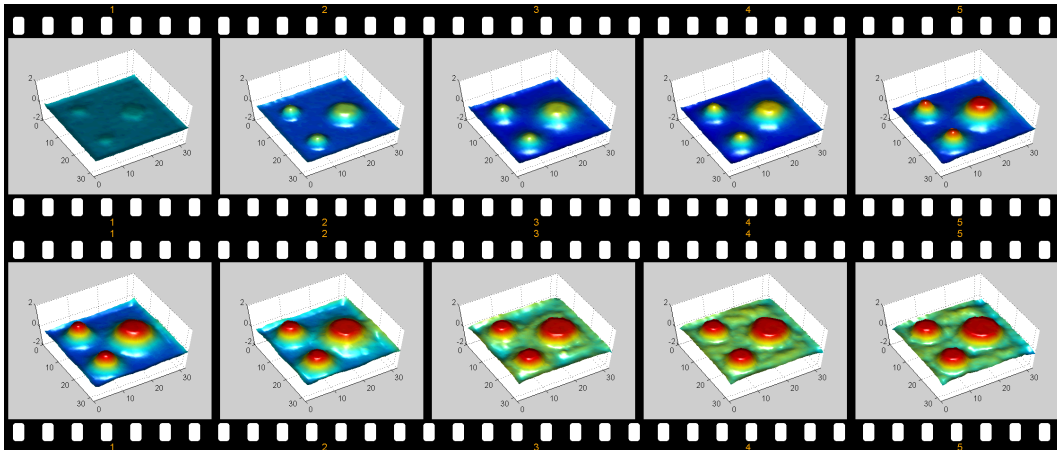


Figure 112. Retina-like wide-field erasure effect, slower layer.

Traveling wave

The characteristic of this wave is that a single wave front travels across the active medium. The traveling wave is simply formed from the trigger wave template by adding a negative intra-layer coupling from the slower to the faster layer. Observe the annihilation property of this wave type in Fig. 113.

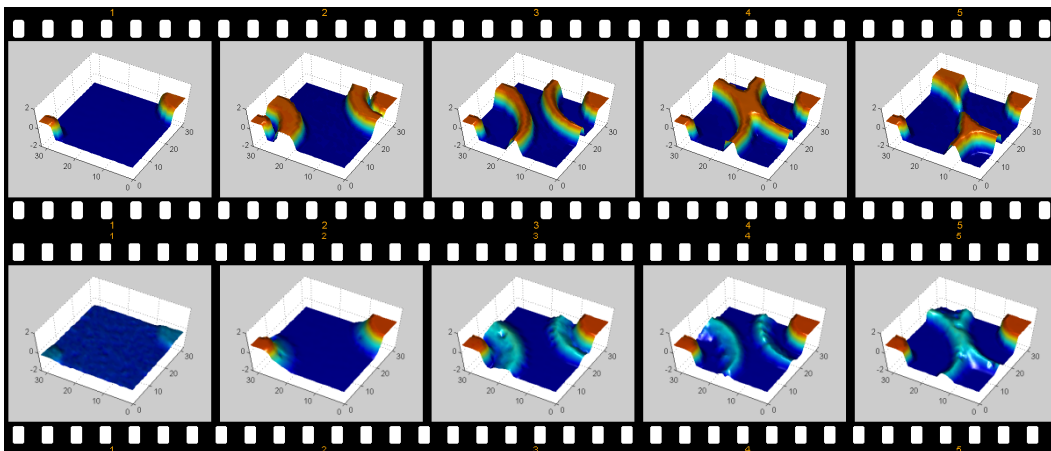


Figure 113. Traveling wave generation.

$$\begin{aligned}
 \mathbf{A}_1 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 4.5 & a_{12} &= -4.2 \\
 b_1 &= 0 & b_2 &= 5 & z_1 &= 0.6 & z_2 &= -2.7 & \tau_1 : \tau_2 &= 16 : 1
 \end{aligned} \tag{6.8}$$

Target wave

The target waves possess all of the properties of the traveling waves; the difference is that in this case multiple wave fronts are generated. See Fig. 114.

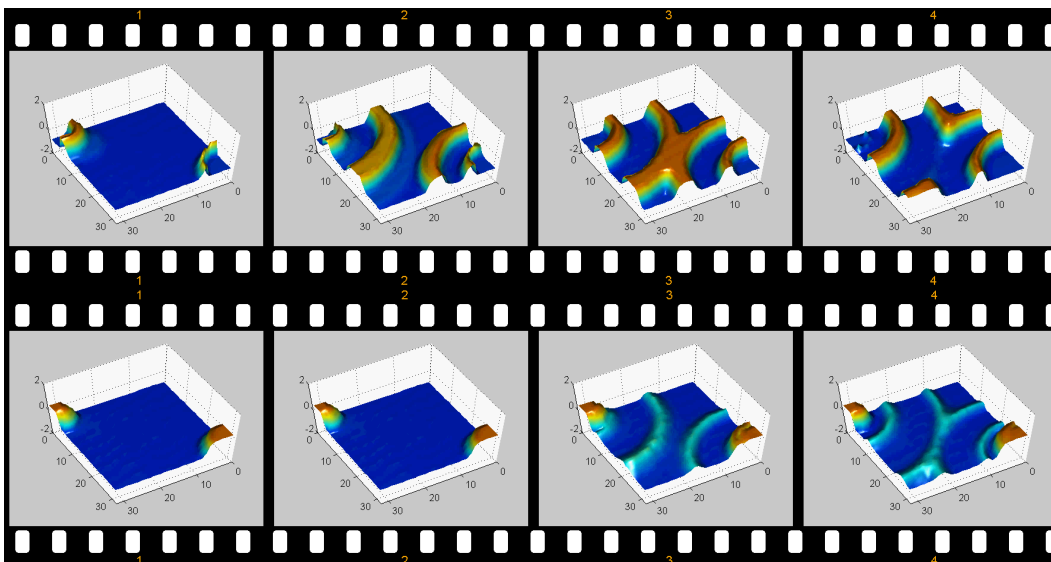


Figure 114. Target wave propagation CACE1k chip.

$$\begin{aligned}
 \mathbf{A}_1 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.9 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 5.7 & a_{12} &= -2.4 \\
 b_1 &= 0 & b_2 &= 3 & z_1 &= -1.5 & z_2 &= -3.9 & \tau_1 : \tau_2 &= 13 : 1
 \end{aligned} \tag{6.9}$$

Spiral wave

This type of wave received its name from the spiral-like autowave structure. This phenomenon was observed e.g. in biology. Examples are described in the Introduction of this dissertation. Fig. 115 shows measurement results on the CACE1k chip. The annihilation of two wave fronts occurs in this case as well.

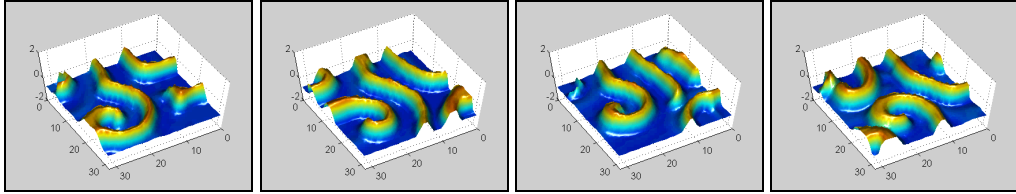


Figure 115. Spiral wave formation in the CACE1k chip.

$$\begin{aligned}
 \mathbf{A}_1 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.3 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0.6 & 0.8 & 0.6 \\ 0.8 & -0.9 & 0.8 \\ 0.6 & 0.8 & 0.6 \end{bmatrix} & a_{21} &= 6 & a_{12} &= -3 \\
 b_1 &= 0 & b_2 &= 3.9 & z_1 &= -4.5 & z_2 &= -3.6 & \tau_1 : \tau_2 &= 8 : 1
 \end{aligned} \tag{6.10}$$

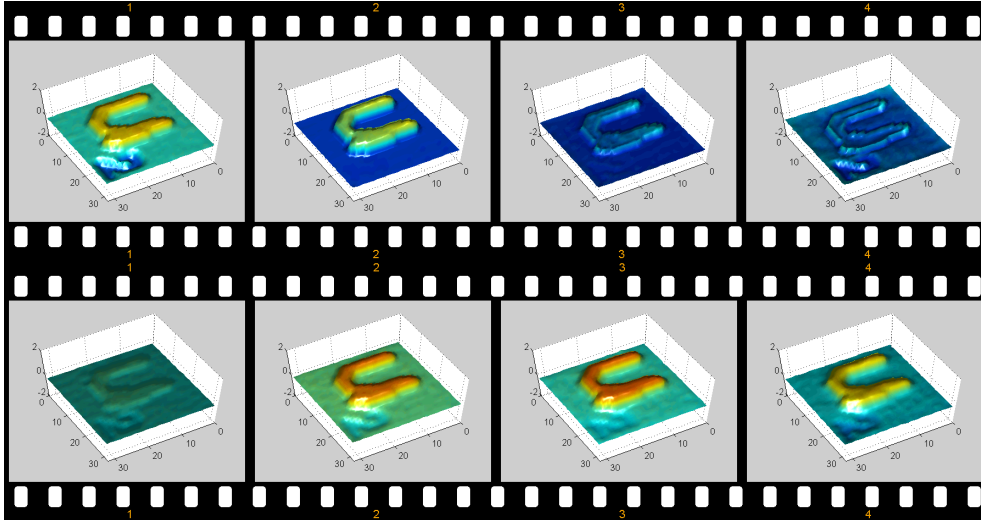


Figure 116. Edge detection with second order dynamics on the CACE1k chip. Upper row shows the output of the faster layer, lower row shows the output of the slower one.

Spatio-temporal edge detection

The basic idea of edge detection in the image processing is to somehow extract or enhance the high frequency components of the image. Here this is done with two layers. The slower layer, which gets its input from the faster one, computes the diffusion. The result is subtracted from the faster layer. As a result, we get the edge image of the original image. The input was fed to the faster layer. See snapshots of the transient in Fig 116.

$$\mathbf{A}_1 = \begin{bmatrix} 0.5 & 0.8 & 0.5 \\ 0.8 & -3.6 & 0.8 \\ 0.5 & 0.8 & 0.5 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad a_{21} = 4.5 \quad a_{12} = -5.7 \quad (6.11)$$
$$b_1 = 0 \quad b_2 = 3 \quad z_1 = -9 \quad z_2 = 0 \quad \tau_1 : \tau_2 = 6 : 1$$



References

- [1] V. I. Krinsky, V. N. Biktashev, and I. R. Efimov, "Autowave Principles for Image Processing", *Physica D* 49, pp. 247-253, 1991.
- [2] J. S. Russel, "Report of the Committee on Waves," Report of the 7th Meeting of the British Association for the Advancement of Science (John Murray, London, 1838), pp. 417-496.
- [3] V. I. Krinsky and A. M. Zhabotinsky, "Autowave Process in Systems with Diffusion", Ed. M. T. Grekhova, Gorki IPF Acad. Sci USSR, p. 6, 1981
- [4] A. N. Zaikin, A. M. Zhabotinsky, "Concentration waves propagation in two-dimensional liquidphase self-oscillating system," *Nature* 225, 535-537, 1970
- [5] A.C. Scott, "The electrophysics of a nerve fiber", *Rev. mod. Phys.* Vol. 47, pp 487-533, 1975
- [6] P. C. Newell, "Microbial Interactions, Receptors and Recognition", Series B, J. L. Reissig, ed., pp. 1-57, Chapman & Hall, 1977
- [7] B. Goodwin, "How the leopard changed its spots", Charles Scribner's Sons, New York, 1994
- [8] A. M. Turing, "The chemical basis of morphogenesis", *Phil. Trans. R. Soc. London*, B327, 37-72 (1952).
- [9] I. Prigogine, "From being to becoming", W. H. Freeman and Company, New York, 1980
- [10] Ser. Ed: L. O. Chua, Ed: V. Pérez-Villar V. Pérez-Muñuzuri, Pérez García and V. I. Krinsky, "New Trends in Pattern Formation in Active Nonlinear Media," World Scientific Series in Nonlinear Science, Series B, Vol. 3, 1995
- [11] L. O. Chua, "CNN: A Paradigm for complexity", World Scientific Pub Co; ISBN: 981023483X, 1998.
- [12] S. Wolfram, *A New Kind of Science*, Wolfram Media, Inc., Champaign IL, 2002.
- [13] S. Wolfram, *Cellular Automata and Complexity*, Westview press, Member of the Perseus Books Group, 1994.

-
- [14] A. Pérez-Muñuzuri, V. Pérez-Muñuzuri, M. Gómez-Gesteira, L.O. Chua, V. Pérez-Villar, "Spatiotemporal structures in discretely-coupled arrays of nonlinear circuits: A review", *Int. J. Bif. Chaos*, Vol. 5, No. 1, pp. 17-50, 1995.
- [15] Krinsky, V. I. (ed.) "Self Organization. Autowaves and Structures Far from Equilibrium" (Springer, Berlin, Singapore), pp. 9-19. 1984
- [16] E. Schrödinger, "What is Life? The Physical Aspect of the Living Cell & Mind and Matter" University Press, Cambridge, 1967
- [17] H. Haken, "Synergetics: From Pattern Formation to Pattern Analysis and Recognition", *International Journal on Bifurcation and Chaos*, Vol. 4, No. 5, pp. 1069-1083, 1994.
- [18] H. Haken, *Synergetics: An Introduction*, Springer Verlag, Berlin, 1978.
- [19] H. Haken, *Synergetic Computers and Cognition*, Springer Verlag, Berlin, 1991.
- [20] G. Nicolis, I. Prigogine, "Exploring Complexity," W. H. Freeman, New York, 1989
- [21] C. G. Langton, "Computation at the edge of chaos: Phase transitions and emergent computation," *Physica D*, Vol. 42, pp. 12-37, 1990
- [22] L. O. Chua, "CNN: a Vision of Complexity", *International Journal of Bifurcation and Chaos*, Vol. 7, No. 10, pp. 2219-2425, 1997.
- [23] R. Dogaru, L. O. Chua, "Edge of Chaos and Local Activity Domain of the Gierer Meinhardt CNN", *International Journal of Bifurcation and Chaos (IJBC)*, Vol. 8, No. 12, pp. 2321-2340, 1998
- [24] R. Dogaru, L. O. Chua, "Edge of chaos and local activity domain of FitzHugh-Nagumo Equation," *Int. J. Bifurcation and Chaos*, Vol. 8, No. 2, pp. 211-257, 1998
- [25] R. Dogaru, L. O. Chua, "Edge of chaos and local activity domain of the Brusselator CNN," *Int. J. Bifurcation and Chaos*, Vol. 8, No. 6, 1107-1130. 1998
- [26] L. O. Chua, "Passivity and Complexity", *IEEE Trans. on Circuits and Systems*, Vol. 46, No. 1, 1999.
- [27] J. P. Keener, "Propagation and its Failure in Coupled Systems of Discrete Excitable Cells", *SIAM J. Appl. Math.*, Vol. 47, 556-572, 1987.
- [28] H. Meinhardt, *The Algorithmic Beauty of Sea Shells (The Virtual Laboratory Series)*, Springer-Verlag New York, ISBN: 3540440100, 2003
- [29] M. Allesie, F. I. M. Bonke, J.G. Schopman, , "Circus movement in rabbit atrial muscle as a mechanism of tachycardia," *Circ. Res.* 33, 54-62.
- [30] L. O. Chua, S Yoon, R. Dogaru, "A nonlinear dynamics perspective of Wolfram's New Kind of Science. Part I: Threshold of Complexity", *International Journal of Bifurcation and Chaos*, Vol. 12, No. 12, 2655-2766, 2002
- [31] L. O. Chua, Ed., "Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays," *IEEE Transactions on Circuits and Systems: I*, vol. 42, no. 10, Oct. 1995.
- [32] J. Weickert, "A review of nonlinear diffusion filtering", (Invited Paper), *Scale-Space Theory in Computer Vision, Lecture Notes in Comp. Science* (Springer, Berlin), 1252:3-28, 1997.

-
- [33] B.M.H. Romeny (ed.), "Geometry-driven Diffusion in Computer Vision," Kluwer Academic Publishers, 1994.
- [34] A.P. Witkin, "Scale-space filtering," Proc. Eighth Int. Joint Conf. on Artificial Intelligence (IJCAI '83), vol.2, pp.1019-1022, 1983.
- [35] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, pp.629-639, July 1990.
- [36] N. Nordström, "Biased Anisotropic Diffusion – A Unified Regularization and Diffusion Approach to Edge Detection", Image Vision Computers, Vol.8, No 4, pp.318-327, July 1990.
- [37] L. Alvarez, P. L. Lions, and J. M. Morel, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. II", *SIAM Journ. on Num. Analysis*, Vol. 29, No. 3, pp. 845-866, 1992.
- [38] F. Catté, P.L. Lions, J. M. Morel and T. Coll, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion", *SIAM Journal on Numerical Analysis*, Vol.29, No.1, pp.182-193, 1992.
- [39] D. Terzopoulos, A. Witkin and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion," *Artificial Intelligence*, Vol. 36, Issue 1, pp. 91-123, August 1988,
- [40] M. Kass, A. Witkin, and D Terzopolous, "Snakes: active contour models," *Int. Journal of Computer Vision*, 1:321–331, 1988.
- [41] L. C. Evans, J. Spruck, "Motion of level sets by mean curvature I," *J. Differential Geometry*, 33 (1991), pp. 635-681
- [42] S. Osher, and J. A. Sethian, "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation", *Journal of Computational Physics*, Vol. 79, pp. 12-49, 1988.
- [43] G. Sapiro, "From Active Contours to Anisotropic Diffusion: Connections between Basic PDE's in Image Processing", *Proceedings 3rd IEEE International Conference on Image Processing (ICIP'96)*, Lausanne, Suisse, September 1996.
- [44] B. B. Kimia & K. Siddiqi, "Geometric Heat Equation and Nonlinear Diffusion of Shapes and Images", *Computer Vision and Image Understanding*, 64(3), 305-322, 1996.
- [45] J. Weickert, "Applications of Nonlinear Diffusion in Image Processing and Computer Vision," *Acta Math. Univ. Comenianae*, LXX(1):33--50, 2001.
- [46] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory", *IEEE Trans. on Circuits and Systems*, Vol. 35, pp. 1257-1272, Oct. 1988.
- [47] L. O. Chua and L. Yang, "Cellular Neural Networks: Applications", *IEEE Trans. on Circuits and Systems*, Vol. 35, pp. 1273-1290, Oct. 1988.
- [48] L. O. Chua, and T. Roska, "The CNN Paradigm", *IEEE Trans. on Circuits and Systems*, Vol. 40, pp.147-156, March 1993.
- [49] T. Roska and L. O. Chua, "The CNN Universal Machine: an Analogic Array Computer", *IEEE Trans. on Circuits and Systems*, Vol. 40, pp. 163-173, March 1993.
- [50] H. Harrer, J.A.Nossek, T. Roska, L.O. Chua, "A Current-mode DTCNN Universal Chip", *Proc. of IEEE Intl. Symposium on Circuits and Systems*, pp135-138, 1994.

-
- [51] J.M. Cruz, L.O. Chua, and T. Roska, "A Fast, Complex and Efficient Test Implementation of the CNN Universal Machine", Proc. of the third IEEE Int. Workshop on Cellular Neural Networks and their Application (CNNA-94), pp. 61-66, Rome Dec. 1994.
- [52] R. Domínguez-Castro, Servando Espejo, A. Rodríguez-Vázquez, R. Carmona, P. Földesy, Á. Zarándy, P. Szolgay, T. Szirányi and T. Roska "A 0.8 μm CMOS 2-D programmable mixed-signal focal-plane array-processor with on-chip binary imaging and instructions storage" Vision Chip with Local Logic and Image Memory, IEEE J. of Solid State Circuits 1997
- [53] A. Passio, A. Kananen and V. Porra, "A 176 x 144 processor binary I/O CNN-UM chip design", European Conference on Circuit Theory and Design - ECCTD'99, Design Automation Day proceedings, (ECCTD'99-DAD), Stresa, Italy, 1999
- [54] S. Espejo, R. Domínguez-Castro, G. Liñán, and Á. Rodríguez-Vázquez, "A 64x64 CNN universal chip with analog and digital I/O" Proc. 5 th Int. Conf. on Electronics, Circuits and Systems (ICECS-98), Lisbon, Portugal, pp. 203-206 1998.
- [55] G. Liñán, R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, "ACE16k: A Programmable Focal Plane Vision Processor with 128 x 128 Resolution". ECCTD '01 - European Conference on Circuit Theory and Design, pp.:345-348, August 28-31,2001, Espoo, Finland
- [56] V.M. Brea, A. Passio, D.L. Vilarino and D. Cabello, "A DTCNN CMOS Implementation of a Pixel-Level Snake Algorithm," ECCTD'01 - European Conference on Circuit Theory and Design, Vol. II, pp. 269 - 272. August, 2001, Espoo, Finland.
- [57] Á. Zarándy, R. Domínguez-Castro, S. Espejo "Ultra-high frame rate focal plane image sensor and processor", IEEE Sensor Journal, December 2002
- [58] R. Carmona, F. Jiménez-Garrido, R. Domínguez-Castro, S. Espejo, T. Roska, Cs. Rekeczky, I. Petrás and A. Rodríguez-Vázquez, "A Bio-Inspired two-Layer Mixed-Signal Flexible Programmable Chip for Early Vision". *IEEE Transactions on Neural Networks*, vol. 14, issue 5, pp. 1313-1336, ISSN 1045-9227, 2003
- [59] I. Petrás, C. Rekeczky, T. Roska, R. Carmona, F. Jimenez-Garrido, and A. Rodriguez-Vazquez, "Exploration of Spatial-Temporal Dynamic Phenomena in a 32x32-Cells Stored Program 2-Layer CNN Universal Machine Chip Prototype", *Journal of Circuits, Systems, and Computers (JCSC)* Vol. 12, No. 6, 2003
- [60] Á. Zarándy, T. Roska, P. Szolgay, S. Zöld, P. Földesy and I. Petrás, "CNN Chip Prototyping and Development Systems", European Conference on Circuit Theory and Design - ECCTD'99, Stresa, Italy, 1999.
- [61] T. Roska, Á. Zarándy, S. Zöld, P. Földesy and P. Szolgay, "The Computational Infrastructure of Analogic CNN Computing - Part I: The CNN-UM Chip Prototyping System", *IEEE Trans. on Circuits and Systems I: Vol. 46, No.2*, pp. 261-268, 1999.
- [62] "CNN Software Library (Templates, subroutines, and Algorithms) Version 8.1", Edited by T. Roska, L. Kék, L. Nemes, Á. Zarándy, Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, 1999
- [63] I. Szatmári, P. Földesy, Cs. Rekeczky, Á. Zarándy "Image Processing Library for the Aladdin Visual Computer," Proceedings of the CNNA-2002 Frankfurt, Germany.

-
- [64] C. Rekeczky and L. O. Chua, "Computing with Front Propagation: Active Contour and Skeleton Models in Continuous-Time CNN," *Journal of VLSI Signal Processing Special Issue: Spatio-temporal Signal Processing with Analogic CNN Visual Microprocessors*, vol. 23, 1999, pp. 373-402.
- [65] I. Szatmári, A. Schultz, C. Rekeczky, T. Kozek, T. Roska and L. O. Chua, "Bubble-Debris Classification via Binary Morphology and Autowave Metric on CNN," *IEEE Trans. on Neural Networks*, vol. 11, 2000, pp. 1385-1393.
- [66] I. Szatmári, C. Rekeczky and T. Roska, "A nonlinear wave metric and its CNN implementation for object classification," *Journal of VLSI Signal Processing Special Issue: Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors*, vol. 23, 1999, pp. 437-448.
- [67] Cs. Rekeczky, T. Roska and A. Ushida, "CNN-based difference-controlled adaptive nonlinear image filters," *Int. J. Circuit Theory and Applicat.*, vol. 26, pp. 375-423, 1998.
- [68] Cs. Rekeczky, "CNN architectures for constrained diffusion based locally adaptive image processing," *Int. J. Circuit Theory and Applicat.*, vol. 30, pp. 313-348, 2002.
- [69] P. Arena, L. Fortuna, M. Branciforte, "Reaction-Diffusion CNN Algorithms to Generate and Control Artificial Locomotion," *IEEE Transaction on Circuits and Systems, I: Fundam. Theory and Applic.*, Vol. 46 No. 2, pp. 259-266, Feb. 1999
- [70] D. Bálya, B. Roska, T. Roska, F. S. Werblin: "A CNN framework for modeling parallel processing in a mammalian retina", *Intl. Journal on Circuit Theory and Applications (CTA)*, Vol. 30, 2002
- [71] T. Roska and Á. Rodríguez-Vázquez: "Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors", *Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology, (JVSP)*, Vol. 23. No.2/3. Kluwer, ISSN 0922-5773, 1999.
- [72] K. R. Crouse and L. O. Chua, "Methods for Image Processing in Cellular Neural Networks: A Tutorial", *IEEE Trans. on Circuits and Systems*, Vol. 42, No. 10, pp. 583-601, October 1995.
- [73] K. R. Crouse and L. O. Chua, "A Synergetics Approach to Image Processing in Cellular Neural Networks", in *Proceedings 1996 International Symposium on Circuits and Systems IS-CAS'96*, Vol. 3, pp. 134-137, May 1996.
- [74] P. Arena, A. Basile, L. Mattia, "CNN Wave Based Computation for Robot Navigation Planning," *Proc. of ISCAS 2004*, 2004
- [75] Cs. Rekeczky, T. Roska, "Calculating Local and Global PDEs by Analogic Diffusion and Wave Algorithms", *Proc. of ECCTD01*, Vol. 2, pp. 17- 20, Espoo, Finland, 2001
- [76] F. Corinto, M. Biey, M. Gilli, "Nonlinear PDE Based Cellular Neural Network Models", *Proc. of ECCTD03*, Vol. II, pp. 440- 443, Cracow, Poland, 2003
- [77] V. Perez-Muñuzuri, V. Perez-Villar, and L. O. Chua, "Autowaves for image processing on a two-dimensional CNN array of excitable nonlinear circuits: Flat and wrinkled labyrinth," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 174-181, Mar. 1993.

-
- [78] A. Perez-Muñuzuri, V. Perez-Muñuzuri, V. Perez-Villar, and L. O. Chua, "Spiral waves on a 2-D array of nonlinear circuits," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 872-877, Nov. 1993.
- [79] V. Pérez-Muñuzuri, A.P. Muñuzuri, M. Gómez-Gesteira, V. Pérez-Villar, L. Pivka and L.O. Chua, "Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Cellular Neural Networks," *Philosophical Transactions of the Royal Society of London, Series A*. 353, pp. 101-113, 1995
- [80] L. O. Chua, M. Hasler, G. S. Moschytz and J. Neiryneck, "Autonomous Cellular Neural Networks: A Unified Paradigm for Pattern Formation and Active Wave Propagation," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, pp. 559-577, 1995.
- [81] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, "Simulating Nonlinear Waves and Partial Differential Equations via CNN - Part I: Basic Techniques", *IEEE Trans. on Circuits and Systems*, Vol. 42, No. 10, pp. 807-815, October 1995.
- [82] 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. on Circuits and Systems*, Vol. 42, No. 10, pp. 816-821, October 1995.
- [83] Cs. Rekeczky, I. Szatmári, and P. Földesy, "Computing on Silicon with Trigger-waves: Experiments on CNN-UM Chips", *IEEE International Symposium on Circuits and Systems ISCAS 2001*, Sydney, 2001
- [84] Cs. Rekeczky, I. Petrás, I. Szatmári and P. Földesy "Active Wave Computing on Silicon: Chip Experiments", *IEEE European Conference on Circuit Theory and Design - ECCTD 2003*, Krakow, pp. II-430-433, Sept., 2003
- [85] I. Petrás, T. Roska, "Application of Direction Constrained and Bipolar Waves for Pattern Recognition" *Int. Workshop on Cellular Neural Networks and Their Applications (CNNA2000)*, Proc., pp. 3-8, Catania, Italy, 2000
- [86] M. Gage & R. S. Hamilton "The heat equation shrinking convex plane curves," *J. Diff. Geom.*, vol. 23, pp. 69-96, 1986.
- [87] M. A. Grayson "The heat equation shrinks embedded plane curves to round points," *J. Diff. Geom.*, vol. 29, pp. 285-314, 1987.
- [88] F. Mokhtarian, A. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves," *IEEE PAMI*, Vol. 14, No. 8, pp. 789-805, 1992
- [89] J. J. Koenderink and A. J. van Doorn, "Dynamic shape. Biological Cybernetics," Vol. 53, pp. 383-396, 1986.
- [90] B. Kimia, A. Tannenbaum, and S. Zucker, "Shape, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space," *International Journal of Computer Vision*, Vol. 15, pp. 189-224, 1995.
- [91] P. Thiran, K. R. Crouse, L. O. Chua, and M. Hasler, "Pattern formation properties of autonomous cellular neural networks", *IEEE Trans. Circuits Syst. I*, vol. 42, pp. 757-774, Oct. 1995.

-
- [92] C. Serpico, G. Setti, P. Thiran, S. Callegari, "Comparing Dynamic Behaviour in 1-D CNNs and Reaction-Diffusion-Convection PDEs: a Review", ECCTD01, ESPOO Finland, 2001, pp. II-29-32
- [93] M. Biey, P. Checco, and M. Gilli, "Bifurcation and chaos in CNNs", *Journal of Circuits Systems and Computers*, vol. 12, no. 4, pp. 417-433, August 2003.
- [94] F. A. Savaci and J. Vandewalle, "On the Stability Analysis of Cellular Neural Networks," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, 1993
- [95] D. Ziou, S. Tabbone, "Edge Detection Techniques - An Overview," *International Journal of Pattern Recognition and Image Analysis*, Vol. 8, No. 4, pp. 537-559, 1998.
- [96] B.M.H. Romeny (ed.), *Geometry-driven Diffusion in Computer Vision*, Kluwer Academic Publishers, pp. 49, 1994.
- [97] A. Rodríguez-Vázquez, S. Espejo, R. Dominguez-Castro, J.L. Huertas, and E.Sánchez-Sinencio, "Current-Mode Techniques for the Implementation of Continuous- and Discrete-Time Cellular Neural Networks", *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol.40. No.3. pp. 132-146, 1993
- [98] G. Liñán, S. Espejo, R. Dominguez-Castro and A. Rodríguez-Vázquez, "ACE4k: an Analog I/O 64x64 Visual Microprocessor Chip with 7-bit Analog Accuracy", *International Journal of Circuit Theory and Applications*, Vol.30, pp.89-116, 2002
- [99] L. Chua and T. Roska, "Cellular neural networks and visual computing - Foundations and applications", Cambridge University Press, ISBN: 0521652472, 2001.
- [100] F. Zou and J. A. Nossek, "Stability of cellular neural networks with opposite-sign templates," *IEEE Trans. Circuits Syst. I*, vol. 38, pp. 675-677, 1991.
- [101] P. Thiran, G. Setti, and M. Hasler, "An Approach to Information Propagation in 1-D Cellular Neural Networks - Part I: Local Diffusion" *IEEE Trans. on Circuits and Systems*, Vol. 45, No. 8, pp. 777-789, 1998.
- [102] G. Setti, P. Thiran, and M. Hasler, "An Approach to Information Propagation in 1-D Cellular Neural Networks - Part II: Global Propagation" *IEEE Trans. on Circuits and Systems*, Vol. 45, No. 8, pp. 790-811, 1998.
- [103] F. Zou and J. A. Nossek, "Bifurcation and chaos in cellular neural networks," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 166-173, Mar. 1993.
- [104] M. Biey, M. Gilli, P. Checco, "Complex dynamic phenomena in space-invariant cellular neural networks", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Volume: 49 Issue: 3 , March 2002 pp. 340 -345.
- [105] G. Manganaro, P. Arena, L. Fortuna "Cellular Neural Networks: Chaos, Complexity and VLSI Processing", Springer Verlag, New York; ISBN: 3540652027, 1999
- [106] L. O. Chua and T. Roska, "Stability of a Class of Nonreciprocal Cellular Neural Networks", *IEEE Trans. on Circuits and Systems*, Vol. 37, No. 12, pp. 1520-1527, 1990.
- [107] T. Roska and A. Rodríguez-Vázquez (editor), "Towards the Visual Microprocessor: VLSI Design and the Use of Cellular Network Universal Machines", John Wiley & Sons; ISBN: 0471956066, 2000.

-
- [108] The ALADDIN System, <http://www.analogic-computers.com/>
- [109] C. W. Wu and L. O. Chua, "A More Rigorous Proof of Complete Stability of Cellular Neural Networks", *IEEE Trans. on Circuits and Systems*, Vol. 44, No. 4, pp. 370-371, 1997.
- [110] M. Gilli, "Stability of Cellular Neural Networks and Delayed Cellular Neural Networks with Nonpositive Templates and Nonmonotonic Output Functions," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, pp. 518-528, vol. 41, 1994
- [111] G. De Sandre, "Stability of 1-D-CNN's with Dirichlet Boundary Conditions and Global Propagation Dynamics," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, pp. 785-792, 2000.
- [112] P. P. Civalleri and M. Gilli, "Global dynamic behaviour of a three cell connected component detector CNN," *International Journal of Circuit Theory and Applications*, vol. 23, no. 2, pp. 117-135, March 1995.
- [113] N. Takahashi and T. Nishi, "On the Global Stability of Two-Cell Cellular Neural Networks with Opposite-Sign Connections," Proceedings of the 15th European Conference on Circuit Theory and Design, vol.3, pp.93-96, August 2001.
- [114] P. P. Civalleri and M. Gilli, "A Spectral Approach to the Study of Propagation Phenomena in CNNs", *Int. J. Circuit Theory and Applications*, vol. 24, 1996, pp. 37-48.
- [115] M. Gilli, "Analysis of periodic oscillations in finite-dimensional CNNs through a spatio-temporal harmonic balance technique," *Int. J. Circuit Theory and Applicat.*, vol. 25, pp. 279-288, July-August 1997.
- [116] Zou, F. & Nossek, J. A. "A chaotic attractor with cellular neural networks" *IEEE Trans. Circuits Syst. I*, vol. 38, 811-812, 1991.
- [117] M. Gilli, "Strange attractors in delayed cellular neural networks," *IEEE, Trans. Circuits and Syst.-I*, vol. 40, 1993, pp. 849-853.
- [118] I. Petráš, Tamás Roska, Leon O. Chua "New Spatial-Temporal Patterns and The First Programmable On-Chip Bifurcation Test-Bed" *IEEE Trans. on Circuits and Systems I, (TCAS I)*, pp. 619-633, May 2003
- [119] P. Thiran, "Influence of Boundary Conditions on the Behaviour of Cellular Neural Networks," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, 1993
- [120] I. Petráš, P. Checco, M. Gilli, and T. Roska, "On the Effect of Boundary Condition on CNN Dynamics: Stability and Instability; Bifurcation Processes and Chaotic Phenomena", *Proc. of ISCAS 2003*, pp. III 590-593, 2003
- [121] J.P. Eckmann and D. Ruelle, "Ergodic Theory of Chaos and Strange Attractors", *Review of Modern Physics*, vol. 57, pp. 617-659, 1985.
- [122] N.Takahashi and L.O.Chua, "On the Complete Stability of Nonsymmetric Cellular Neural Networks", *IEEE Trans. Circuits and Systems-I*, vol.45, no.7, pp.754-758, July 1998.
- [123] Cs. Rekeczky, T. Serrano-Gotarredona, T. Roska and A. Rodríguez-Vázquez, "A Stored Program 2nd Order/3-Layer Complex Cell CNN-UM", *Proc. Sixth IEEE Intern. Workshop on Cellular Neural Networks and their Applications*, Catania, Italy, May 2000, pp.219-224.

- [124] R. Carmona, F. Jiménez-Garrido, R. Domínguez-Castro, S. Espejo and A. Rodríguez-Vázquez, "Bio-inspired Analog VLSI Design Realizes Programmable Complex Spatio-Temporal Dynamics on a Single Chip". *Proc. Design, Automation and Test in Europe Conference (DATE'02)*, pp. 362-366. Paris, France, March 2002. Published by the IEEE Computer Society, Los Alamitos, CA, ISBN 0-7695-1471-5.
- [125] S. Espejo, R. Carmona, R. Domínguez-Castro and A. Rodríguez-Vázquez, "A VLSI Oriented Continuous-Time CNN Model". *Intern. J. Circuit Theory and Applications*, John Wiley & Sons, 24, No.3, May-June 1996, pp.341-356.
- [126] Fujita, K. Tanaka, M. Ito and K. Cheng, "Columns for visual features of objects in monkey inferotemporal cortex", *Nature* 360, pp. 343-346, 1992.
- [127] J.L. Barron, D.J. Fleet, S.S. Beauchemin, "Performance Of Optical Flow Techniques", *Int J of Comp Vis*, vol. 12, no. 1, pp. 43-77, 1994.
- [128] Zarándy, A., Domínguez-Castro, R., Espejo, S.: "Ultra-high Frame Rate Focal Plane Image Sensor and Processor". *IEEE- Sensors Journal*, IEEE. Vol. 2 . Num. 6 - Dec. 2002. pp. 559-565
- [129] Á. Zarándy, Cs. Rekeczky, I. Szatmári, "Vision Systems Based on the 128x128 Focal Plane Cellular Visual Microprocessor Chips", *Proc. of ISCAS 2003*, pp. III 518-521, 2003
- [130] V. Gál, J. Hátori, T. Roska, D. Bálya, Zs. Borostyánkői, M. Brendel, K. Lotz, L. Négyessy, L. Orzó, I. Petrás, Cs. Rekeczky, J. Takács, P. Venetiáner, Z. Vidnyánszky and Á. Zarándy "Receptive Field Atlas and Related CNN Models", *International Journal of Bifurcation and Chaos (IJBC)*, Vol. 14(2), pp. 551-584, February 2004
- [131] D. Bálya, I. Petrás, T. Roska, R. Carmona, A. Rodríguez-Vázquez "Implementing the Multi-Layer CNN Retina Model on the Complex-Cell Stored Program 2-Layer CNN Universal Machine Chip Prototype", *Int. Journal of Bifurcation and Chaos, (IJBC)*, Vol. 14(2), pp. 551-584, 2004,
- [132] I. Petrás, T. Roska, "New spatial-temporal patterns and the first programmable on-chip bifurcation test-bed", *Proceedings of 7th IEEE International Workshop on Cellular Neural Networks and their Applications*, Frankfurt, Germany, 2002
- [133] P. Szolgay, K. László, L. Kék, T. Kozek, L. Nemes, I. Petrás, Cs. Rekeczky, I. Szatmári, Á. Zarándy, S. Zöld and T. Roska, "The CADETWin Application Software Design System - A Tutorial", *European Conference on Circuit Theory and Design - ECCTD'99, Design Automation Day proceedings, (ECCTD'99-DAD)*, Stresa, Italy, 1999
- [134] P. Szolgay, Á. Zarándy, S. Zöld, T. Roska, P. Földesy, L. Kék, T. Kozek, K. László, I. Petrás, Cs. Rekeczky, I. Szatmári and D. Bálya, "The Computational Infrastructure for Cellular Visual Microprocessors", *Proceedings of Seventh Int. Conf. on Microelectronics for Neural, Fuzzy, and Bio-Inspired Systems, (MicroNeuro'99)*, pp. 54-60, Granada, IEEE Computer Society, 0-7695-0043- 9, 1999
- [135] M. Gilli, T. Roska, L. O. Chua and P. P. Civalleri, "CNN Dynamics Represents a Broader Class than PDEs", *International Journal of Bifurcation and Chaos*, Vol.12, pp.2051-2068, 2002

-
- [136] F. Cortino, M. Gilli, "Comparison between the dynamic behaviour of Chua-Yang and Full-Range cellular neural networks", (CAS,in press).
- [137] M. Gilli, F. Corinto, M. Biey, P. P. Civalleri, "On the dynamic behaviour of cellular neural networks," Proc. of IJCNN 2002, pp. 1936–1941, 2002
- [138] B. E. Shi, "Order Statistic Filtering with Cellular Neural Networks," *Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, vol. , 1994, pp. 441-444.
- [139] P. Arena, S. Baglio, L. Fortuna, G. Manganaro, "Dynamics of state controlled CNNs", Proc of ISCAS'96, Vol. 3, pp. 56–59, 1996
- [140] I. Prigogine, "Nonlinear Science and the Laws of Nature," *International Journal of Bifurcation and Chaos*, Vol. 7, No. 9, 1917-1926, 1997
- [141] H. Haken, "Vision of Synergetics," *International Journal of Bifurcation and Chaos*, Vol. 7, No. 9, 1927-1951, 1997
- [142] H. Meinhardt, "Biological Pattern Formation as A Complex Dynamic Phenomenon," *International Journal of Bifurcation and Chaos*, Vol. 7, No. 1, 1-26, 1997
- [143] A. V. Holden, M. J. Poole and J. V. Tucker, "An algorithmic model of the mammalian heart: propagation, vulnerability, re-entry and fibrillation," *International Journal of Bifurcation and Chaos*, Vol. 6, No. 9–10, 1623-1636, 1996.
- [144] V. I. Sbitnev, A. O. Dudkin, "Checkerboard Spiral Waves in a 2d Coupled Map Lattice. Scaling Evidence", *International Journal of Bifurcation and Chaos*, Vol. 9, No. 5, , pp. 919–928, 1999
- [145] S. Paul, K. Hüper, J. A. Nossek and L. O. Chua, "Mapping Nonlinear Lattice Equations onto Cellular Neural Networks," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, pp. 196-203, 1993
- [146] St. Jankowski, C. Mazur, R. Wanczuk, "A CNN model of lattice gas", *Proceedings of 11 European Conference on Circuit Theory and Design, (ECCTD'93)*, pp.699-702, Davos, 1993,
- [147] P. C. Newell, "Attraction and adhesion in the slime mold *Dyctyostelium*," in *Fungal Differentiation: A Contemporary Synthesis* ed. Smith, J. E., *Micology Series 43* (Marcel Dekker, NY), p. 43, 1983
- [148] K. Kaneko, Ed., *Theory and applications of coupled map lattices*, John Wiley & Sons, Chichester, 1993.
- [149] K. Kaneko, I. Tsuda, *Chaos and Beyond*, ISBN 3540672028, Springer Verlag, New York, 2001, pp 57-106.
- [150] E.R. Kandel, J.H. Schwartz, "Principles of Neural Science", second edition, Elsevier, New York, Amsterdam, Oxford, 1985
- [151] A. Sanz-Anchelergues, A. P. Munuzuri, "Experimental and Quantitative Modeling Studies of Turing Pattern Formation Under Stochastic Spatial Fluctuations", *International Journal of Bifurcation and Chaos*, Vol. 11, No. 11 , 2739–2749, 2001
- [152] L. Goras, L. O. Chua and D. M. W. Leenaerts, "Turing patterns in CNNs—Part I, once over lightly", *IEEE Trans. Circuits and Systems—Part I*, 42, 602–611 (1995).

-
- [153] L. Goras and L. O. Chua, "Turing patterns in CNNs—part II: equations and behaviours" *IEEE Trans. Circuits and Systems—Part I*, 42, 612–626 (1995).
- [154] L. Goras, L. O. Chua and L. Pivka, "Turing patterns in CNNs—part III: computer simulation results", *IEEE Trans. circuits Systems - Part I*, 42, 627–637 (1995).
- [155] R. N. Madan, Ed., "Chua's Circuit: A Paradigm for Chaos," Singapore, World Scientific, 1993
- [156] Szabolcs Tóké, László Orzó, Ahmed Ayoub and Tamás Roska, "Flexibly Programmable Opto-Electronic Analogic CNN Computer (POAC) Implementation Applying an Efficient, Unconventional Optical Correlator Architecture", Special Issue on "CNN Technology and Visual Microprocessors", *Journal of Circuits, Systems and Computers*, vol. 12, no. 6, pp. 739–767, December 2003
- [157] S. Tóké, L.R. Orzó, G. Váró, A. Dér, P. Ormos and T. Roska, "Programmable Analogic Cellular Optical Computer using Bacteriorhodopsine as Analog Rewritable Image Memory", NATO Book: "Bioelectronic Applications of Photochromic Pigments" (Eds.: L. Keszthelyi, J. Stuart and A. Der) IOS Press, Amsterdam, Netherlands, pp 54–73. 2000
- [158] M. Di Marco, A. Tesi and M. Forti, "Bifurcations and oscillatory behaviour in a class of competitive Cellular Neural Networks," *Int. Journal of Bifurcation and Chaos*, Vol. 10, pp. 1267–1293, 2000
- [159] M. Di Marco, M. Forti and A. Tesi, "Existence and Characterization of Limit Cycles in Nearly Symmetric Neural Networks", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol.49, No.8, pp.979–992, 2002
- [160] M. Di Marco, M. Forti and A. Tesi, "Complex Dynamics in Nearly Symmetric Three-Cell Cellular Neural Networks", *Int. Journal of Bifurcation and Chaos*, Vol. 12, No. 6, pp. 1357–1362, 2002
- [161] I. Petrás, T. Roska, "CNN as Curve Shortening Flow Computer", *Proceedings of 8th IEEE International Workshop on Cellular Neural Networks and their Applications*, Budapest, Hungary, 2004
- [162] J. E. Hopcroft, R. Motwani, J. D. Ullman, "Introduction to Automata Theory, Languages and Computation", Addison Wesley, 2000.
- [163] B. Roska, F.S. Werblin, "Vertical Interactions Across Ten Parallel Stacked Representations in the Mammalian Retina," *Nature*, Vol. 410, pp. 583–587, 2001
- [164] <http://ct.radiology.uiowa.edu/~jiangm/courses/mm-cv-ip/node69.html>
- [165] <http://ct.radiology.uiowa.edu/~jiangm/courses/mm-cv-ip/node75.html>
- [166] http://math.berkeley.edu/~sethian/level_set.html
- [167] http://www.lems.brown.edu/vision/publications/Kimia's_Publication/Journals/shape_shocks_deformation-IJCV.pdf
- [168] <http://www.cim.mcgill.ca/~shape/publications/cvpr94.pdf>

Appendix A

Typical AMC program example.

```

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

```

APPENDIX A

```
51      sc.add.gamxf 50 gamf1 gamf1
52      sc.rel.gt 3500.0 gamf1 glm1
53      jumpc glm1 Loop
54      end
```


Appendix B

Curvature of a plane curve defined by a implicit equation $\Gamma: u(x,y)=0$. See online [165,168].

We assume that $(X(t),Y(t))$ is a parameterization of Γ . As shown in Fig. 117. Moreover we assume near point \mathbf{P} , $\dot{X} < 0$, $\dot{Y} < 0$, $u_y > 0$ and $u_x < 0$.

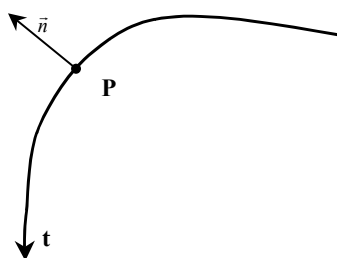


Figure 117. Plane curve defined by $u(x,y)=0$. $\vec{n} = \frac{\nabla u}{\|\nabla u\|}$

Then

$$u_x \dot{X} + u_y \dot{Y} = 0$$

$$u_{xx} \dot{X}^2 + 2u_{xy} \dot{X}\dot{Y} + u_{yy} \dot{Y}^2 + u_x \ddot{X} + u_y \ddot{Y} = 0.$$

Therefore

$$\dot{X}^2 + \dot{Y}^2 = \dot{X}^2 + \left(\frac{u_x}{u_y}\right)^2 \dot{X}^2 = \left(1 + \left(\frac{u_x}{u_y}\right)^2\right) \dot{X}^2$$

and

$$\begin{aligned} \dot{X}\ddot{Y} - \dot{Y}\ddot{X} &= -\dot{X} \frac{1}{u_y} \left[u_{xx} \dot{X}^2 + 2u_{xy} \dot{X}\dot{Y} + u_{yy} \dot{Y}^2 + u_x \ddot{X} \right] + \dot{X} \frac{u_x}{u_y} \ddot{X} = \\ &= -\frac{1}{u_y} \left[u_{xx} \dot{X}^2 + 2u_{xy} \dot{X}\dot{Y} + u_{yy} \dot{Y}^2 \right] \dot{X}. \end{aligned}$$

Hence, the curvature is

$$\begin{aligned}
\kappa &= \frac{\dot{X}\ddot{Y} - \dot{Y}\ddot{X}}{(\dot{X}^2 + \dot{Y}^2)^{3/2}} = \frac{-\frac{1}{u_y} [u_{xx}\dot{X}^2 + 2u_{xy}\dot{X}\dot{Y} + u_{yy}\dot{Y}^2]\dot{X}}{\left(\left[1 + \left(\frac{u_x}{u_y} \right)^2 \right] \dot{X}^2 \right)^{3/2}} = \frac{\frac{1}{u_y} \left[u_{xx} + 2u_{xy} \frac{\dot{Y}}{\dot{X}} + u_{yy} \left(\frac{\dot{Y}}{\dot{X}} \right)^2 \right]}{\left(1 + \left(\frac{u_x}{u_y} \right)^2 \right)^{3/2}} = \\
&= \frac{\frac{1}{u_y} \left[u_{xx} + 2u_{xy} \frac{u_y}{u_x} + u_{yy} \left(\frac{u_y}{u_x} \right)^2 \right]}{\left(1 + \left(\frac{u_x}{u_y} \right)^2 \right)^{3/2}} = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}.
\end{aligned}$$

