Stability, Optical Flow and Stochastic Resonance in Cellular Wave Computing

by Levente Török

B.Eng. College of Mechanical Engineering and Automation (1995)
M.Eng. University of Miskolc (2000)

Submitted to the
Péter Pázmány Catholic University in a fullfillment of the requirements for the degree of
Doctor of Phylosophy

November 2005

Under the supervision of Ákos Zarándy, Ph.D.

Analogic and Neural Computing Laboratory,
Computer and Automation Research Institute
Hungarian Academy of Sciences

Dedicated to the memory of my father,

Miklós, Török,
1940-2003

Abstract

Just as many of us, I am also prone to handle engineering achievements as being a construction built from solid, reliable, brick-like components. Naturally, this not true in all senses. However, just like living species of mother Nature, electrical systems may not require perfect predictability in all aspects to operate correctly but only in the important ones. Hence, the central questions are "what is important?" and "how to design systems for it?".

In this aspect, the *stability* of the systems we use, is a *must*; the *applicability* is a *need*; the noise tolerance is a *desire*.

In the light of these findings, I concentrated on these questions in my work with special regard to a non-conventional computational paradigm, named Cellular Neural/Non-linear Networks Universal-Machine (CNN-UM) paradigm.

First, in response to the question *how we can use this constuction*, I dealt with questions of stability of a newly invented version of the CNN-UM. Then, answering the quesion *what it can be used it for*, I applied the CNN to calculate a computationally heavy task, by the application of which it was accellareted by several orders of magnitudes. Finally, I dealt with the question *whether it can be made noise robust*, and I found that the system can even produce the so called stochastic resonance phenomena in which it can be optimized against noise by the assistance of the noise itself.

Acknowledgements

First of all, I am greatful to my professor, Prof. Tamás Roska, for providing a constant support for my studies, that was first to collect all knowledge, that I felt to be important to make up a basis for a research-engineer, and also for the support of my research that was usually running in diverse directions simultaneously.

I would also thank my supervisor, Dr. Ákos Zarándy, giving me both feedback and suggestions, by asking me some ground-real questions on my work.

I also have to say thank a very close colleague, Dr. László Orzó, who's critical view fertilized me, I believe, in a positive way. I feel, that in many cases, discussions and heated debates ended with contribution to my knowledge. I also thank my close friend Kristóf Karacs, my teachers and professors in my studies such as Dr. Tihamér Ádám, Dr. József Bukszár, Dr. András Lőrincz, Dr. Pál Rózsa, Dr. Dénes Vadász, Dr. Ivánné Vágó, for supporting and educating me in many ways.

I will also need to emphasize the help and support of my family. Among them I have to pronounce the role of my father, Dr. Miklós Török, who made me enthusiastic about the main questions of the science and borrowed me his, sometimes over too critical, aspects on ethical questions. I am just hoping that I can be a good student of him; my mother, and my elder brother who never stopped supporting me, no matter what I did.

Finally, thanks *Office of Naval Research and the Army Research Office*, LOCUST project (IST-2001-38097), Hungarian Academy of Sciences Grant No. 00014-021-0884.

Contents

Chapter 1

Introduction

The field of non-linear science rises a lot questions that we are not ready to solve and we are left without any hope of solving them in the near future. There are certain equations that received much higher respect than others. In the past 15 years, many efforts has been spent on understanding and exploiting the capabilies of one of them called Cellular Non-linear/Neural Network (CNN). The main driving force of these works was the hope that the construction is simple enough to get implemented in analog very large scale integrated circuitry (aVLSI) but complex enough to be able solve general, parallel, computing problems and several computationally hard (e.g. NP) tasks quickly (in P) and efficiently. Here, I am not to answer the question whether it was successful or not but I hope that I contributed to the world's knowledge on this particular equation with my work both in application and in theoretical directions. According to this I have covered three topics.

The first topic covers stability analysis of the a multi layered version of CNN which aims to clarify that which parameter constellations can this particular version of CNN be used for any purpose. The main problem here is to ensure, with proof, that stable and reproducible output of the circuitry can be reached. This requirement is a natural expectation for any device, but analog circuitry sometimes have hard time to meet this criterion.

Opposed to the first topic, which is a rather theoretical issue, the second topic focuses on much more application oriented questions. Namely, I have discovered, an interesting formalism that lets us to calculate optical-flow estimation on aVLSI circuitry in which multi-scale information is incorporated. Interestingly enough, the algorithm I proposed provides sub-pixel accuracy that is currently unbeatable in the aVLSI community and is comparable to very complex, cca. 10 years old algorithms on digital platforms.

Thirdly, I introduce a phenomena called "smooth stochastic resonance". Stochastic resonance (SR) occurs when the signal-to-noise-ratio (SNR) at output of a non-linear device is maximized by additional noise at the input. Stochastic resonators (SRT) known before, were able to demonstrate this behavior only with abrupt jumps at the output. However, it resulted in the maximization of the SNR, it was inconvenient enough to apply for anything. Hence it was desirable to obtain an easy-to-use solution for this.

In the following sections I give very short summary of the findings that are detailed longer in the subsequent chapters and at the same time serves as a summary of theses.

1.1 First Thesis: Stability of MLCNN

Since the introduction of the Universal Machine (UM) on Cellular Neural/Non-linear Networks, many works has been devoted to the analysis of the principle and the circuitry. Inspired by biology, namely retina, the multi-layer architecture was also designed and fabricated in analog VLSI (aVLSI). Although, there were not much works published on the stability of these type of networks. Hence, there was a need to set up bounds on parameters in which engineers can search for useful constellations (or templates).

I was able to give a formalism that lets us to generalize most of theorems claimed on single-layer-CNN to multi-layer-CNN. Being able to use this tool, I presented several new theorems and made a strong conjecture in the circle of networks with two and arbitrary number of layers. Theorems were selected with special regard to the applicability.

For the sake of readability let us define single layer Full-Range CNN as

$$\tau\dot{\hat{x}} = -g(\hat{x}) + \hat{A} * \hat{x} + \hat{B} * \hat{u} + \hat{z} \tag{1.1}$$

where $*$ denotes convolution operation and its multi-layer counterpart (MLCNN) as

$$\tau_p\dot{\hat{x}}_p = -g(\hat{x}_p) + \sum_{o=1}^{L} \hat{A}_{op} * \hat{x}_o + \hat{B}_p * \hat{u}_p + \hat{z}_p \tag{1.2}$$

where p is the layer number, L is number of layers, τ_p is the time constant of layer p, $\hat{A}_{op}, \hat{B}_p$ are the feedback template from layer o to layer p and feed forward templates to layer p, respectively. The matrices $\hat{x}, \hat{u}, \hat{z}$, are the 'state', 'input' and 'bias' . Finally, function g is a non-analytic and defined as

$$g((\hat{x})_{ij}) = \begin{cases} 0 & , \text{ if } |(\hat{x})_{ij}| < 1 \\ lim_{m\to\infty} \quad (|(\hat{x})_{ij}| - 1) \cdot m \cdot \mathbf{sgn}((\hat{x})_{ij}) & , \text{else .} \end{cases}$$

Once we found the way to generalize equ. 1.2 in form of equ. 1.1 (see chapter 2), we can transfer theorems from single layer to two and multi-layer CNNs. (i.e. 2LCNN, MLCNN). Being armed with such a tool, I have generalized 7 theorems and 1 strong conjecture. The criteria of the theorems that maintain stability in case of a 2 layer CNN with 3x3 templates, can be sloppily below.

A two layer CNN is completely stable in the

- *Symmetric templates case*: if not only templates operating each layers (i.e. intra-layer template) are symmetric, but those that are operating among layers (i.e. inter-layer template) are symmetric to each other (i.e. cross-symmetric),
- *τ-cross symmetric case*: if intra-layer template elements are symmetric and inter-layer template elements are cross-symmetric up to a constant multiplicative factor. (i.e. so called layer time-constant τ)
- *Non-symmetric case*: if items of the templates follow a pattern of geometric series (i.e. the ratio of the neighboring elements are constant) in both horizontal and vertical directions, and similar mirroring rule as above applies to the inter-layer templates,
- *τ-non symmetric case*: the theorem above holds, for any ratio of intra layer time-constant
- *Sign symmetric case*: if the sign pattern of the intra-layer templates are strictly sign-symmetric and follow the same sign pattern, and the inter-layer templates are equal to that or are their inverse

- *"Sign patterns for larger templates" case*: if the sign alteration frequency of the inter-layer template elements in both, horizontal and vertical, directions is either 0 or 1, while the intra-layer sign patter is equal to that or is the inverse of that.

In case of a CNN with arbitrary number of layers in the

- *"Sign patterns of large templates for more than 2 layers" case*: if the sign alteration frequency of the inter-layer template elements in both directions is either 0 or 1, while the matrix describing the inversions compared to that (i.e. sign modifier matrix) is a so-called chess-board type.

Several numerical and symbolical computations support the conjecture that the CACE architecture (e.g. 2LCNN with inter-layer templates that have only a central non zero element) is stable in the staturated region in the

- *"Non-reciprocal 1D CACE" case*: if both inter-layer, intra-layer templates are sign anti-symmetric and the central element of the intra-layer feedbacks are greater than one.

This is detailed in chapter 2.

1.2 Second Thesis:Analog-VLSI, Array-Processor Based, Bayesian, Multi-Scale Optical Flow Estimation

Optical flow (OF) estimation aims to derive a motion-vector field that characterizes the motion in a video sequence of images. Since the motion cannot be deterministically derived, I turned the problem into a stochastic optimization framework, in which multiple-scale view of the subsequent images is used at once. This was done to reduce the effect of the error prone cases (also known as *blank wall* or *aperture* problem), which are main source of the artifact in OF estimation.

1.2.1 A CNN friendly stochastic multi-scale Optical Flow estimation model

A fundamentally a new multi-scale OF model was designed in which I showed two ways of formalizing the Bayesian incorporation of multiple scales, leading essentially to the same equations to compute in the model. The new algorithm was found to be optimally fitting to the architecture of the CNN-UM by the application of which it was accelerated by 3 orders of magnitudes compared to digital architectures.

1.2.2 On-chip tests

Due to the resolution (128x128) and reliability of the CNN-UM implementation (ACE16k), first in the history, the new general-purpose, aVLSI-based OF estimator was tested on a standard image sequence (Yosemite) that is used as the main performance indicator in the computer vision community. However the new solution performs with fair accuracy compared to the digital sequential competitors (i.e. sub-pixel absolute error and average angular error (AAE) $\approx 10\circ$), but it runs at 100 frames per second (fps) on-chip, and the on-chip tests relax well to simulations (AAE $\approx 3.5\circ$). On pixel-per-second basis, the new method outperforms any aVLSI solutions with at least one maginitude difference, while the accuracy is comparable to digital solutions, which is unique.

1.3 Third thesis: Smooth Stochastic Resonance

Stochastic Resonance (SR) is a phenomenon in which a system with input signal and noise shows a well defined peak in the signal-to-noise (SNR) characteristic at the output as the function of the noise (RMS) at the input.

I have invented a new class of stochastic resonator (SRT) and Stochastic Resonance phenomena. The new SRT consist of a classical SRT, one or more time derivative circuits and the same number of time integrators. First, the incoming signal with additive noise is time derivated, then it passes through a classical SRT, and, finally, it is time integrated. This is the Integro-Differential SRT (IDSRT). The resulting SR phenomena show a well defined SR. The main virtue of the new system is the ability to produce arbitrary smooth output which was unprecedented before. Interestingly, the signal transfer and SNR are the best at the high frequency end.

I have also showed that the same phenomena can be produced in 2D images where the terms of "derivation" and "integration" correspond to inverse-diffusion (deblur) and diffusion (blur) operation.

1.3.1 The analysis of IDSRT

First, I clarified the analysis of the stochastic resonator called Level Crossing Detector (LCD) and reproduced the results the corresponding experiment, then I gave the analysis of the IDSRT with symmetric and asymmetric LCD. Furthermore, I showed the model's ability of transferring the SR characteristics of the LCD to IDSRT but the later one can produce arbitrary smooth output.

1.3.2 The experimental results on IDSRT

Simulation results confirmed the calculations mentioned above.

Chapter 2

Stability of Multi-Layer Cellular Neural/Non-linear Networks

2.1 Brief

We have found a formalism that lets us present generalizations of several stability theorems (see [CR90, CC92, Gil93, For02]) on Multi-Layer Cellular Neural/Non-linear Networks (MLCNN) formerly claimed for Single-Layer Cellular Neural/Non-linear Networks (CNN). The theorems were selected with special regard to usefulness in engineering applications. Hence, in contrast to many works considering stability on recurrent neural networks, the criteria of the new theorems have clear indications that are easy to verify directly on the template values. Proofs of six new theorems on 2-Layer CNNs (2LCNN) related to symmetric, τ-symmetric, non-symmetric, τ-non-symmetric, and sign-symmetric cases are given. Furthermore, a theorem with a proof on a MLCNN with arbitrary template size and arbitrary layer number in relation to the sign-symmetric theorem is given, along with a conjecture for the one-dimensional, two-layer, non-reciprocal case.

2.2 Notation and Symbols

2LCNN	2 Layer CNN
$\tilde{x}, \tilde{A}$	compact vector form
$\hat{x}, \hat{A}$	matrix form
x^T, A^T	transpose of matrix and vector
A	recurrent part of the template of CNN in concatenated form
B	feed forward part of the template of CNN
CNN	Cellular Neural/Non-linear Network
CNN-UM	CNN Universal Machine
CYCNN	Chua and Yang Model of CNN
diag	a diagonal matrix composed of vector in the argument
f	transfer function of the CNN
FR-CNN	Full-Range Model of CNN

J^L	sign modifier matrix
MLCNN	Multi-Layer CNN
PWL	Piece Wise Linear function
sgn	signum function
τ_p	layer time constant
f^{-1}	invese of function f
$(\tilde{A}_{ij})_{kl}$	element at position k, l of the sub-matrix at position ij of matrix A
$\dot{x}$	time derivative of x
$\otimes$	diadic multiplication
$\odot$	Kronecker matrix multiplication

2.3 Introduction

The analysis of the stability of Multi-Layer Cellular Neural/Non-linear Networks (MLCNN) [CY88a, CR90] is motivated by the appearance of a new 2-Layer CNN Universal Machine (CNN-UM) architecture called CACE (Complex cell Analog-and-logic non-linear Computing Engine) [RFR$^+$02], that has already been manufactured. Interestingly, the region of stable operation (i.e. template values) of Multi-Layer architetures has not been analyzed as intimately as its Single-Layer counterpart which is still far from being entirely clarified.

However they are not completely discovered yet, several important spatio-temporal phenomena have been experimentally tested and its relevance in modeling has been demonstrated ref. [CTTA00].

Our papers main virtue is a formalism that enables us to treat Multi-Layer architectures as simple as a Single-Layer CNN.

The main results refer mostly for a 2-Layer CNN. Namely, we show that a 2LCNN is stable

- if the inter-layer feedback templates are symmetric, and the intra-layer feedback template is cross-symmetric or τ-cross symmetric (i.e. weighted with the corresponding layer time constant),
- if in the class of non-symmetric templates, the so-called constant-ratio rule holds among the template entries,
- if all of the templates match to a single sign pattern in which sign alteration frequency is a fixed one or zero in both horizontal and vertical directions and the phases of the two inter-layer feedbacks are equivalent while the inter-layer feedbacks may vary upto a constant phase compared to them.

Similarly to that, MLCNN with arbitrary layer number is stable with arbitrary template size if a similar hypothesis holds with the extension that the matrix of the phase changes is in the class of chess board type matrices.

In the following, the CACE architecture is described step-by-step (Sec. 2.4) starting from the Chua Yang model of CNN, then its compact vector form is reviewed with the extension to multi-layer equivalent equations that made the generalization of the formerly known theorems possible (Sec. 2.5), and new theorems are presented in Sec. 2.6. The appendix recalls the known and the new proofs with our unified formalism to help the reader, however our contribution can be found in the main part.

2.4 Multi-Layer and CACE Architectures

2.4.1 Chua-Yang Model [CY88a]

A CNN operates on a lattice-structured set of *state variables* $(\hat{x})_{ij}$, $i \in \{1,...,N\}; j \in \{1,...,M\}$[1] *input* $(\hat{u}_{ij})$ and *bias map* $(\hat{z}_{ij})$ influenced by each other via a small set of local connection variables, called *template* ($\hat{A}$ and $\hat{B}$). Both the variables and the templates are arranged in matrix form. In hardware implementations the size of the templates usually does not extend beyond the neighborhood of radius $r = 1$, which mean a 3×3 *template* size (i.e. $r = 1 \Leftrightarrow \exists(\hat{A})_{kl} \neq 0 \ : \ k,l \in \{-1,0,1\}$).

The system is governed by the following differential equation,

$$\tau(\dot{\hat{x}}(t))_{ij} = -(\hat{x}(t))_{ij} + \sum_{kl \in S_r(ij)} (\hat{A})_{kl} \cdot (\hat{y}(t))_{i+k,j+l} + \sum_{kl \in S_r(ij)} (\hat{B})_{kl} \cdot (\hat{u}(t))_{i+k,j+l} + (\hat{z}(t))_{ij}$$

or in short

$$\tau\dot{\hat{x}} = -\hat{x} + \hat{A} * \hat{y} + \hat{B} * \hat{u} + \hat{z} \tag{2.1}$$

where $*$ symbolizes the convolution operator. Note that capitals are constants.

$$(\hat{y})_{ij} = f((\hat{x})_{ij}) = \frac{(|(\hat{x})_{ij} + c| - |(\hat{x})_{ij} - c|)}{2c} \tag{2.2}$$

where c is usually chosen to be 1.

This model was introduced by Chua and Yang in ref. [CY88a] therefore it is called Chua-Yang model (CYCNN).

2.4.2 Full-Range Model [ERVDVC94]

For practical reasons, different types of non-linearities were introduced. One of them is the so-called Full-Range model, which limits the operational range of state variables $(\hat{x})_{ij}$, rather than limiting the output as in Eq. (2.2),

$$\tau\dot{\hat{x}} = -g(\hat{x}) + \hat{A} * \hat{x} + \hat{B} * \hat{u} + \hat{z} \tag{2.3}$$

where

$$g((\hat{x})_{ij}) = \begin{cases} 0 & , \text{ if } |(\hat{x})_{ij}| < 1 \\ lim_{m\to\infty} \quad (|(\hat{x})_{ij}| - 1) \cdot m \cdot \mathbf{sgn}((\hat{x})_{ij}) & , \text{else .} \end{cases} \tag{2.4}$$

Since the majority of the stability works was made on CYCNN model and almost exclusively FRCNN model was implemented, one might asks the question whether the stability theorems claimed for CYCNN holds for FRCNN too? Some important aspects of the differences between the two models (i.e. CYCNN and FRCNN) were examined in ref. [CG03], where it was observed that even the trajectories of the two models might depart qualitatively, the convergence criteria of all the stability theorems that have been known so far are still intact.

The function $g()$ is a limit function because of its asymptotic definition. A definition like this, indeed, imposes difficulties in numerical tractability and in functional analysis but it is easier to implement and

[1] We index and use brackets of matrices as $(m_{ij}(t))_{kl}$, where (ij) refers to the submatrix in the ith row jth column of the hypermatrix (m) at time (t) and (kl) refers to the matrix element in kth row lth column.

it speeds up the operation (either convergence or instability) compared to the CY model. Furthermore, it reduces noise sensitivity in the analog implementations.

2.4.3 Multi-Layer Architectures

Even though Multi-Layer CNN (MLCNN) architectures have been known since 1991 [CS91], it did not generate as many publications as the Single-Layer architerures did since, which is probably due to its increased complexity in the analysis.

Multi-layer architectures consist of two or more sets of indexed state variables influenced via $(\hat{A})_{op}$ kernels where o and p refer to the so-called *from* and *to* layers, respectively. Disregarding the technical difficulties in VLSI implementations, MLCNNs could be defined as,

$$\tau_p \dot{\hat{x}}_p = -g(\hat{x}_p) + \sum_{o=1}^{L} \hat{A}_{op} * \hat{x}_o + \hat{B}_p * \hat{u}_p + \hat{z}_p$$

where L defines the number of layers ($p \in \{1, ..., L\}$).

One should note that the entries in vector τ might vary from layer to layer. For a simple case like $L = 2$ (i.e. 2LCNN), one can get,

$$\begin{aligned} \tau_1 \dot{\hat{x}}_1 &= -g(\hat{x}_1) + \hat{A}_{11} * \hat{x}_1 + \hat{A}_{21} * \hat{x}_2 + \hat{B}_1 * \hat{u}_1 + \hat{z}_1 \\ \tau_2 \dot{\hat{x}}_2 &= -g(\hat{x}_2) + \hat{A}_{12} * \hat{x}_1 + \hat{A}_{22} * \hat{x}_2 + \hat{B}_2 * \hat{u}_2 + \hat{z}_2. \end{aligned} \tag{2.5}$$

Defining the same architecture with respect to Chua-Yang model,

$$\begin{aligned} \tau_1 \dot{\hat{x}}_1 &= -\hat{x}_1 + \hat{A}_{11} * \hat{y}_1 + \hat{A}_{21} * \hat{y}_2 + \hat{B}_1 * \hat{u}_1 + \hat{z}_1 \\ \tau_2 \dot{\hat{x}}_2 &= -\hat{x}_2 + \hat{A}_{12} * \hat{y}_1 + \hat{A}_{22} * \hat{y}_2 + \hat{B}_2 * \hat{u}_2 + \hat{z}_2. \end{aligned}$$

2.4.4 CACE Architecture

In ref. [RFR+02] the CACE architecture with a complete CNN-UM core is defined as

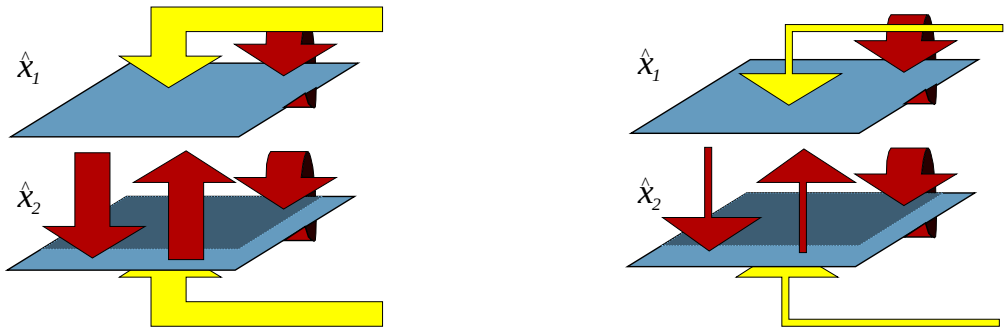
$$\begin{aligned} \tau_1 \dot{\hat{x}}_1 &= -g(\hat{x}_1) + \hat{A}_{11} * \hat{x}_1 + \hat{a}_{21} \cdot \hat{x}_2 + \hat{b}_1 \cdot \hat{u}_1 + \hat{z}_1 \\ \tau_2 \dot{\hat{x}}_2 &= -g(\hat{x}_2) + \hat{a}_{12} \cdot \hat{x}_1 + \hat{A}_{22} * \hat{x}_2 + \hat{b}_2 \cdot \hat{u}_2 + \hat{z}_2. \end{aligned} \tag{2.6}$$

One can define CACE architecture with respect to Chua-Yang model as

$$\begin{aligned} \tau_1 \dot{\hat{x}}_1 &= -\hat{x}_1 + \hat{A}_{11} * \hat{y}_1 + \hat{a}_{21} \cdot \hat{y}_2 + \hat{b}_1 \cdot \hat{u}_1 + \hat{z}_1 \\ \tau_2 \dot{\hat{x}}_2 &= -\hat{x}_2 + \hat{a}_{12} \cdot \hat{y}_1 + \hat{A}_{22} * \hat{y}_2 + \hat{b}_2 \cdot \hat{u}_2 + \hat{z}_2. \end{aligned}$$

In Eq. (2.6) lower case $\hat{a}_{21}, \hat{a}_{12}, \hat{b}_1, \hat{b}_2$ refer to scalar values which are multiplied by the state and the input vectors, rather than convolved with them. This simplification is due to the costs of implementation belonging to the fully connected inter-layer feedback templates. Nevertheless, interesting spatio-temporal dynamics can evolve in this structure.

In Fig. 2.1 one can compare differences between a 2-layer and a CACE architecture.



(a) *General 2-layer CNN (2LCNN) architecture* (b) *The CACE architecture*

Figure 2.1: *The fully inter-connected MLCNN and the sparsely interconnected CACE architectures are depicted on the left and right, respectively. Intra-layer feedback templates ($\hat{A}_{11}$ and $\hat{A}_{22}$) are symbolized by self-closing arrows on the layers ($\hat{x}_1$ and $\hat{x}_2$), intra-layer feedback templates are symbolized by arrows between the layers. The differences between CACE and 2-layer architectures are emphasized by the width of the arrows referring to the size of convolution kernel*

Note that in contrast to former CNN implementations τ_1 and τ_2 time constants are programmable.

2.5 Compact Vector Form

At the first sight, the parts most difficult to handle in the system of equations are the convolutions. To get rid of them, a standard rewriting method is used.

Consider a state equation of an $M \times N$ CNN. In the compact form, new vectors $(\tilde{x}, \tilde{y}, \tilde{u})$ are introduced by repacking variables by an invertible rearrangement

$$(\tilde{x})_{i+M \cdot j} = (\hat{x})_{ij}. \tag{2.7}$$

In order to be able to write the equivalent system of equations, the form of the kernels needs to be reformulated as well, resulting in a *Block Toeplitz* matrix shown in Fig. 2.2.

So the equivalent system of equations may be written as

$$\tau \dot{\tilde{x}} = -g(\tilde{x}) + \tilde{A}\tilde{x} + \tilde{B}\tilde{u} + \tilde{z}$$

Applying this to a 2LCNN, one can write Eq. (2.5) as

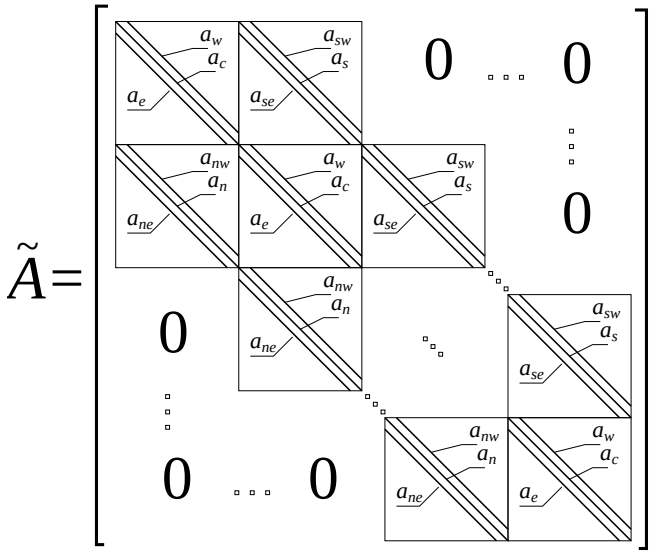
$$\begin{aligned} \tau_1 \dot{\tilde{x}}_1 &= -g(\tilde{x}_1) + \tilde{A}_{11}\tilde{x}_1 + \tilde{A}_{21}\tilde{x}_2 + \tilde{B}_1\tilde{u}_1 + \tilde{z}_1 \\ \tau_2 \dot{\tilde{x}}_2 &= -g(\tilde{x}_2) + \tilde{A}_{12}\tilde{x}_1 + \tilde{A}_{22}\tilde{x}_2 + \tilde{B}_2\tilde{u}_2 + \tilde{z}_2. \end{aligned} \tag{2.8}$$

By packing the state, the output, the input and the bias map vectors of the two layers in single vectors, as $x^T = (\tilde{x}_1^T, \tilde{x}_2^T), y^T = (\tilde{y}_1^T, \tilde{y}_2^T)$ etc., one can write

$$\tau \dot{x} = -g(x) + Ax + Bu + z \tag{2.9}$$

$$\hat{A} = \begin{array}{|c|c|c|} \hline a_{nw} & a_n & a_{ne} \\ \hline a_w & a_c & a_e \\ \hline a_{sw} & a_s & a_{se} \\ \hline \end{array}$$

(a) *Entries in the kernels are denoted by the capitals of compass points (i.e. n for north,e for east,etc.) plus c for central*



(b) *The same kernel in compact matrix representation form*

Figure 2.2: *To transform the convolution in Eq. (2.1) to vector-matrix multiplication, a matrix-to-vector repacking described by Eq. (2.7) was used therefore, the matrix of kernels needed to be converted to the so-called compact matrix form having Block Toeplitz structure as shown on the right-hand side.*

where hypermatrices τ, A, B were created by similar concatenation as

$$\tau = \begin{pmatrix} \tau_1 I & 0 \\ 0 & \tau_2 I \end{pmatrix}, \quad A = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{21} \\ \tilde{A}_{12} & \tilde{A}_{22} \end{pmatrix}, \quad B = \begin{pmatrix} \tilde{B}_1 & 0 \\ 0 & \tilde{B}_2 \end{pmatrix}, \tag{2.10}$$

where I denotes the identity matrix. Each hypermatix is of size $2MN \times 2MN$. Note that, the compact vector form of CNN is exactly the same as Eq. 2.10. It can easily be extended for arbitrary number of layers with arbitrary size of templates. The form still holds.

However it needs to be remarked that there is one order of magnitude difference in the degree of freedoms of the convolution kernels ($df(\hat{A}_{ij}) \rightarrow 4 \times 9$) and their counterpart matrices ($df(A_{ij}) \rightarrow 2MN \times 2MN$). Naturally, not all stable solutions for matrix A can be transformed to a kernel. The main constraint on the matrix structure is the principle of *spatial invariance*.

The obtained form provides a powerful tool, by which many of the known theorems claimed for Single-Layer CNNs can be generalized to MLCNN architectures. All we need to do is to carefully revise all steps that might touch the degree of freedom of matrix A and explore all the new possibilities inherent to its greater being.

In the next sections we give a far not complete list of theorems that can be extended in this manner.

Definition 1. If in network none of the cell's state depends explicitly on the cell's position (or site index) then the network is called *spatially invariant*.

In the definition above, implicit dependence might refer to dependence on index differences.

Disregarding boundary conditions the definition can be formulated for a Single-Layer CNNs as,

$$(A)_{ij} = function(i-j) \tag{2.11}$$

(read as: the entries of matrix A depend exclusively on a function of the difference of the entries' horizontal and vertical indexes), which is equivalent to the definition of Toeplitz matrices. Similar constraints apply to MLCNNs,

$$(A_{kl})_{ij} = function(kl, i-j). \tag{2.12}$$

Definition 2. The definition of the *locality* can be formulated as

$$(\hat{A})_{ij} = 0 : ||i,j|| > r \;\Rightarrow\; A_{\sigma(k,l;i,j)} = 0 : ||i,j|| > r$$

where σ is an invertible mapping of state variables from $\hat{A}$ to A.

Remark 1. The Bidirectional Associative Memory (BAM) neural network resembles a 2LCNN to a certain extent. This network was introduced by Kosko [Kos88] as,

$$\begin{aligned}(\dot{\hat{x}}_1)_i &= -(\kappa_1)_i(\hat{x}_1)_i + \sum_{j=1}^{p}(\hat{A}_1)_{ji}\cdot(\hat{y}_2)_j + (\hat{\phi}_1)_i \\ (\dot{\hat{x}}_2)_i &= -(\kappa_2)_i(\hat{x}_2)_i + \sum_{j=1}^{p}(\hat{A}_2)_{ji}\cdot(\hat{y}_1)_j + (\hat{\phi}_2)_i.\end{aligned}$$

which is equivalent to,

$$\begin{aligned}\frac{1}{(\kappa_1)_i}(\dot{\hat{x}}_1)_i &= -(\hat{x}_1)_i + \frac{1}{(\kappa_1)_i}\left(\sum_{j=1}^{p}(\hat{A}_1)_{ji}\cdot(\hat{y}_2)_j + (\hat{\phi}_1)_i\right) \\ \frac{1}{(\kappa_2)_i}(\dot{\hat{x}}_2)_i &= -(\hat{x}_2)_i + \frac{1}{(\kappa_2)_i}\left(\sum_{j=1}^{p}(\hat{A}_2)_{ji}\cdot(\hat{y}_1)_j + (\hat{\phi}_2)_i\right).\end{aligned} \tag{2.13}$$

This network can be regarded as a one dimensional, two-layer, fully connected, heteroassociative network. With the matrix concatenation formalism provided above the Eq. (2.13) can be rewritten as,

$$\tilde{\chi}\dot{\tilde{x}} = -\tilde{x} + \tilde{A}\tilde{y} + \tilde{\phi}$$

where $\tilde{\chi} = \begin{pmatrix}\chi_1 & 0\\ 0 & \chi_2\end{pmatrix}$ composed of $\chi_1 = \mathbf{diag}\{1/(\kappa_1)_1, 1/(\kappa_1)_2, ..., 1/(\kappa_1)_P\}$ and $\chi_2 = \mathbf{diag}\{1/(\kappa_2)_1, 1/(\kappa_2)_2, ..., 1/(\kappa_2)_P\}$) denotes the neuron charging time constant, $\tilde{x}^T = (x_1^T, x_2^T)$ and $\tilde{A} = \begin{pmatrix}0 & \chi_1\cdot A_1\\ \chi_2\cdot A_2 & 0\end{pmatrix}$ denotes the connection matrix. $\tilde{y}^T = (y_1^T, y_2^T)$ is given by $(\tilde{y})_i = s_i(x_i(t))$ where s_i represent the neural activation function with the well-known sigmoid characteristics and finally $\tilde{\phi}^T = ((\chi_1\tilde{\phi}_1)^T, (\chi_2\tilde{\phi}_2)^T)$ is a constant additive term.

It looks clear that the only difference between 2LCNN and BAM is manifested in spatial invariance and locality criteria of A. Apart from this, the two networks' dynamical behaviour is equivalent.

In sections (Sec. 2.6.1, 2.6.3) we start from the $\tau_1 = \tau_2 = 1$ case and develop toward the $\tau_1 \neq \tau_2$

case in (Sec. 2.6.2, 2.6.4), while theorem 5 is left unaffected by the modification of temporal parameters (layer time constants) but spectacular extension for larger template size is detailed in Sec. 2.6.5.

2.6 Main Results

Probably the most widely used approach for concluding systems' stability is done by means of Lyapunov functions. Sometimes it is particularly difficult to write an appropriate Lyapunov functional for a given system since its definition is not constructive. Fortunately, several Lyapunov functions have been introduced for CNN versions but other stability considerations coexist. In the following sections, several of them will be introduced.

2.6.1 Symmetric Case

The root of the following theorem can be found in ref. [CR02] for Single-Layer CNNs, where perfect central symmetry is assumed for the template elements. Our new theorem is a logical extension to that, and proposes relation for interconnecting template elements.

Definition 3. Cross symmetry is defined as centrally mirrored equivalence of template elements in $\hat{A}_{12}$ and $\hat{A}_{21}$. (i.e. $\hat{(A_{12})}_{kl} = (\hat{A}_{21})_{-k-l}$ or see Fig. 2.3(b)).

Definition 4. A dynamical system ($\dot{x} = f(x(t)), x(t_0) = x_0,\ x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}^n$) is said to be *completely stable* if for $\forall x_0 \in \mathbb{R}^n$:

$$\lim_{t\to\infty} x(t, x_0) = const$$

(ref. [Gil93])

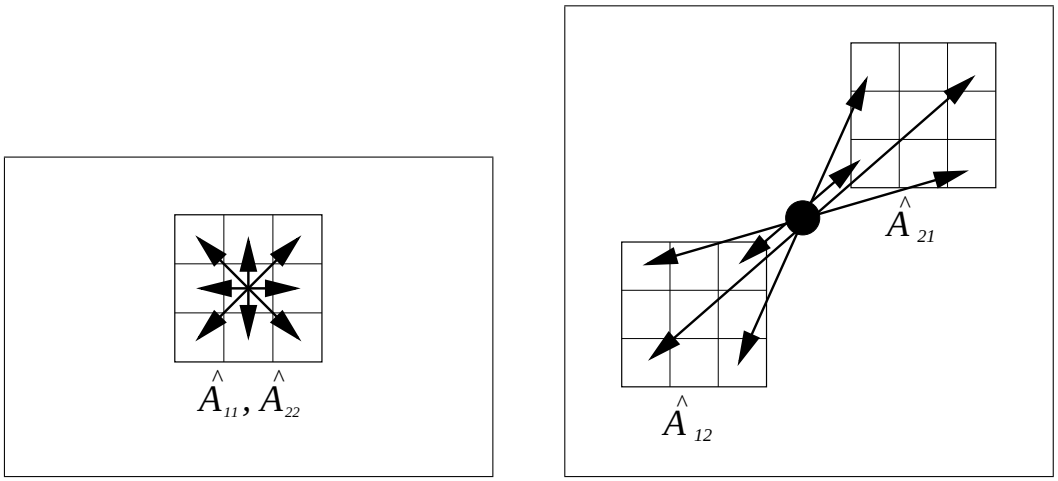
Theorem 1. *A 2-layer CNN is* ***completely stable***[2] *if*

- *The template values are central symmetric for intra-layer feedback sub-template elements ($\hat{A}_{11}$, $\hat{A}_{22}$),*
- *The template values are cross symmetric for inter-layer feedback sub-template elements ($\hat{A}_{12}$, $\hat{A}_{21}$),*
- *Layer time constants are unity ($\tau_1 = \tau_2 = 1$),*
- *$f()$ is continuously differentiable, bounded and $f'() > 0$,*
- *All equilibrium points are isolated.*

The interpretation of the criteria: The first criterion was probably the very first known by the CNN community known as central symmetry. In Fig. 2.3 arrows point to template elements are need to be equal to satisfy central symmetry (Fig. 2.3(a)) and cross symmetry (Fig. 2.3(b)).

The example below meets both criteria above.

[2]Here we restrict ourself to non autonomous systems hence constant $u(t)$ and $z(t)$ are assumed.



(a) *First criterion: Central-symmetry must hold for intra-layer feedback template elements*

(b) *Second criterion: Cross-symmetry must hold for inter-layer feedback template elements*

Figure 2.3: *Arrows point to kernel entries needed to be equal to satisfy the first two criteria of Theorem 1*

$$\hat{A}_{11} = \begin{array}{|c|c|c|}\hline .2 & .3 & .5 \\ \hline .4 & 2 & .4 \\ \hline .5 & .3 & .2 \\ \hline \end{array} \quad \hat{A}_{12} = \begin{array}{|c|c|c|}\hline .1 & .5 & .2 \\ \hline .2 & 1 & .2 \\ \hline .3 & .3 & .4 \\ \hline \end{array}$$

$$\hat{A}_{21} = \begin{array}{|c|c|c|}\hline .4 & .3 & .3 \\ \hline .2 & 1 & .2 \\ \hline .2 & .5 & .1 \\ \hline \end{array} \quad \hat{A}_{22} = \begin{array}{|c|c|c|}\hline .1 & .2 & .3 \\ \hline .1 & 2 & .1 \\ \hline .3 & .2 & .1 \\ \hline \end{array}$$

Later we will recall the trajectory limit length theory of Forti *et al* in ref. [For02] which enables us to drop the last two hypotheses of this theorem as well.

Proof: Assume a continuously differentiable and bounded bijective function $y(x)$, then the equilibrium points of Eq. (2.9) are isolated.

In the case of symmetric, spatial invariant templates, the compact matrix is symmetric ($A = A^T$), which lets us write a Lyapunov function of the form,

$$V(x) = -\frac{1}{2} y^T A y + \sum_{i=1}^{2MN} \left[\int_{\theta}^{y_i} f^{-1}(v) dv \right] - y^T B u - y^T z \tag{2.14}$$

which is the first Lyapunov function introduced for CNNs in [CY88a].

In ref. [CY88a] it is proved that $x(t)$ is bounded ($\exists X_l : |x(t)| < X_l, \forall t$) which means the boundedness of $V(x)$. Having a bounded $V()$ and being its time derivative along the trajectory negative (see appx. A.1), $V()$ is an appropriate Lyapunov function of the system.

By applying LaSalle's invariant principle [LaS67], one can conclude that all trajectories are convergent to the invariant set defined by $M = \{x : \dot{V}(x) = 0\} \quad \forall x \in \mathbb{R}^n$. In fact, complete stability derives from the connectedness of the ω-limit set of each trajectory.

Corollary: In the case of Single-Layer CNNs, the symmetric property of matrix ($A = A^T$) implies the symmetry of the $\hat{A}$ template ($(\hat{A})_{kl} = (\hat{A})_{-k\ -l}$). In the case of 2LCNN this property implies slightly different rules, as well.

Obviously, the symmetry is repeated for $\hat{A}_{11}$ and $\hat{A}_{22}$ intra-layer feedback templates ($(\hat{A}_{ii})_{kl} =$

$(\hat{A}_{ii})_{-k\ -l}\, i \in \{1,2\}$), since their spatial arrangement in hypermatrix A is identical to that of the original CNN from at least the symmetric point of view (i.e. diagonal matrix positions). The only difference arises in case of inter-layer feedback matrices for which the following rule can be concluded,

$$\tilde{A}_{21} = \tilde{A}_{12}^T \quad \Rightarrow \quad (\hat{A}_{12})_{kl} = (\hat{A}_{21})_{-k\,-l}.$$

This relation is called *cross-symmetry* between inter-layer feedback sub-template elements.

Remark 2. It must be noted that the case of perfectly symmetric intra-layer feedback and perfectly anti-symmetric inter-layer feedback case ($(\hat{A}_{12})_{kl} = -(\hat{A}_{21})_{-k\,-l}$) has already been studied in ref. [FLMM93].

2.6.2 τ-Symmetric Case

The case of non-unity τ was neglected in the former sections. Now, the model to be examined is extended and by relying on a theorem proposed in ref. [For02] a new theorem is given. The method overcomes the problem of non-invertible input/output activation function $f()$ in Eq. (2.14), which therefore can be piecewise linear (PWL) or piecewise analytic. Furthermore the theorem proofs the stability in case of *non-isolated* equilibrium points, too. (see appx. A.2)

Theorem 2. *A 2-layer CNN is* ***completely stable*** *if the template values are*

- *Central symmetric for intra-layer feedback sub-templates elements ($\hat{A}_{11}$, $\hat{A}_{22}$)*
- *τ-cross symmetric for inter-layer feedback sub-templates elements ($\hat{A}_{12}$, $\hat{A}_{21}$)*
- *$f()$ is continuously differentiable, bounded and $f'() > 0$*
- *All equilibrium points are isolated.*

Due to the methodology of the proof, the theorem can be extended to the *Absolutely Stable* class of global pattern formation, similarly to the theorem of *M. Forti*, which means complete stability for any choice of parameters defining $\hat{A}_{ij}, \hat{B}_i, \hat{u}_i, \hat{z}_i\ i,j \in \{1,2\}$ satisfying the symmetry criteria given above.

The interpretation of the new criterion: Compared to the previous theorem, a modification appears in the *τ-cross symmetric* criterion which is depicted in Fig. 2.4 by using the same symbols.

Proof: Let us rearrange the equations of a 2-layer CYCNN

$$\begin{aligned}\tau_1\dot{\tilde{x}}_1 &= -\tilde{x}_1 + \tilde{A}_{11}\tilde{y}_1 + \tilde{A}_{21}\tilde{y}_2 + \tilde{B}_1\tilde{u}_1 + \tilde{z}_1\\ \tau_2\dot{\tilde{x}}_2 &= -\tilde{x}_2 + \tilde{A}_{12}\tilde{y}_1 + \tilde{A}_{22}\tilde{y}_2 + \tilde{B}_2\tilde{u}_2 + \tilde{z}_2\end{aligned}$$

as

$$\begin{aligned}\dot{\tilde{x}}_1 &= -\tfrac{1}{\tau_1}\tilde{x}_1 + \tfrac{1}{\tau_1}\tilde{A}_{11}\tilde{y}_1 + \tfrac{1}{\tau_1}\tilde{A}_{21}\tilde{y}_2 + \tfrac{1}{\tau_1}\left(\tilde{B}_1\tilde{u}_1 + \tilde{z}_1\right)\\ \dot{\tilde{x}}_2 &= -\tfrac{1}{\tau_2}\tilde{x}_2 + \tfrac{1}{\tau_2}\tilde{A}_{12}\tilde{y}_1 + \tfrac{1}{\tau_2}\tilde{A}_{22}\tilde{y}_2 + \tfrac{1}{\tau_2}\left(\tilde{B}_2\tilde{u}_2 + \tilde{z}_2\right)\end{aligned}$$

Again, by the concatenation of equations (as in Eq. (2.8)), one can obtain,

$$\dot{x} = -\tau x + A^* y + B^* u + z^* \tag{2.15}$$

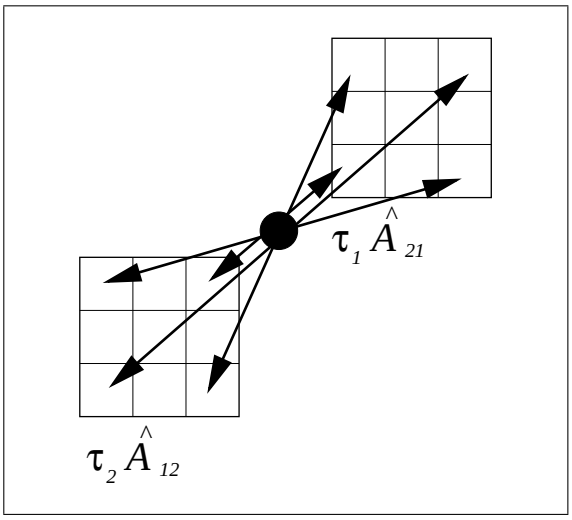


Figure 2.4: *τ-cross symmetric rule: The figure should be interpreted as fig. 2.3 but here templates are multiplied by the corresponding time constant (τ).*

where

$$A^* = \begin{pmatrix} \frac{1}{\tau_1}\tilde{A}_{11} & \frac{1}{\tau_1}\tilde{A}_{21} \\ \frac{1}{\tau_2}\tilde{A}_{12} & \frac{1}{\tau_2}\tilde{A}_{22} \end{pmatrix}, \quad B^* = \begin{pmatrix} \frac{1}{\tau_1}\tilde{B}_1 & 0 \\ 0 & \frac{1}{\tau_2}\tilde{B}_2 \end{pmatrix}$$

and

$$z^* = \begin{pmatrix} \frac{1}{\tau_1}\tilde{z}_1 \\ \frac{1}{\tau_2}\tilde{z}_2 \end{pmatrix}, \quad \tau = \begin{pmatrix} \tau_1 \cdot I & 0 \\ 0 & \tau_2 \cdot I \end{pmatrix}.$$

Consequently, we obtained original CYCNN equation again with the exception of matrix τ. The Lyapunov function introduced in appx. A.2 implies strict symmetry requirement on A^*. This requirement implies symmetric $\hat{A}_{11}$ and $\hat{A}_{22}$ which is the same as in the hypothesis of the previous theorem. Hence compared to the previous theorem, the only modification this matrix τ multiplication leads to can be written as

$$\left(\frac{1}{\tau_1}\tilde{A}_{21}\right)^T = \frac{1}{\tau_2}\tilde{A}_{12}$$

or

$$\tau_2(\hat{A}_{12})_{kl} = \tau_1(\hat{A}_{21})_{-k\,-l}, \tag{2.16}$$

which is the τ-cross symmetric rule. This rule is depicted in Fig. 2.4.

2.6.3 Non-Symmetric Case

Being armed with the new formalism, another Lyapunov function can be rewritten, which again implies a specific relation of intra- and inter-layer template values. This case can be regarded as an extension of the case given in ref. [Gil93] which considers single-layer networks only.

Theorem 3. *A 2-layer CNN is **completely stable** if*

- *The constant ratio rule holds for intra-layer feedback sub-template elements ($\hat{A}_{11}$, $\hat{A}_{22}$)*
- *The constant ratio rule holds between inter-layer feedback sub-template elements ($\hat{A}_{12}$, $\hat{A}_{21}$)*

- *Zero additive term ($\hat{B} * \hat{u} + \hat{z} = 0$) is present*
- *Layer time constants are unity ($\tau_1 = \tau_2 = 1$)*
- *$f()$ is continuously differentiable, bounded and $f'() > 0$*
- *All equilibrium points are isolated.*

The interpretation of the criteria: The so-called *constant ratio rule* is a term for the rule specified by *M. Gilli, et al.* It explains the stability criterion as a ratio of template elements. We say a template is conform to the rule, if the ratio of the northern and southern template elements multiplied by the ratio of the eastern and western elements is equal the ratio of north-southern and south-western elements (see Fig. 2.5 for explanation). Similarly, in a 2LCNN, the same rule must hold for intra-layer feedback sub-template elements, too.

With the same symbolism used in Fig. 2.5, the constant ratio rule *between the two layers* is displayed in Fig. 2.6(a) which is articulated in terms of the inter-layer feedback sub-template elements.

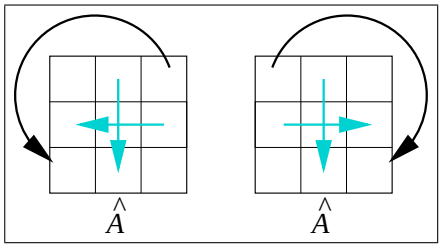
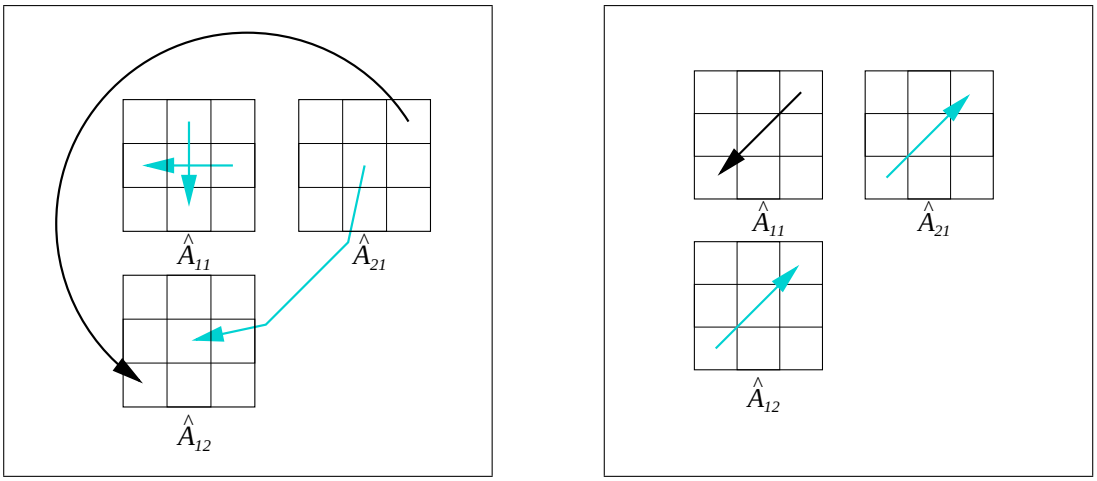


Figure 2.5: *The constant ratio rule of template elements holds if the ratio of the northern and southern template elements (vertical gray arrows) multiplied by the ratio of the eastern and western elements (horizontal gray arrows) is equal to the ratio of north-southern and south-western elements (black arrow). Or in short* $\frac{a_{ne}}{a_{sw}} = \frac{a_n}{a_s}\frac{a_e}{a_w}$ *and* $\frac{a_{nw}}{a_{se}} = \frac{a_n}{a_s}\frac{a_w}{a_e}$. *In case of a 2-Layer CNN (2LCNN), intra-layer feedbacks (i.e.* $\hat{A}_{11}$ *and* $\hat{A}_{22}$*) must possess this property.*



(a) *Eq. (2.20)* (b) *Eq. (2.21)*

Figure 2.6: *The required constant ratio rule between inter-layer feedback template elements*

The example below satisfies the criteria of the theorem in question:

$$\hat{A}_{11} = \begin{array}{|c|c|c|}\hline .4 & .4 & .8 \\ \hline .4 & 0 & .8 \\ \hline .1 & .1 & .2 \\ \hline \end{array} \quad \hat{A}_{12} = \begin{array}{|c|c|c|}\hline 1.2 & 1.2 & 2.4 \\ \hline 1.2 & .3 & 2.4 \\ \hline .1 & .3 & .6 \\ \hline \end{array}$$

$$\hat{A}_{21} = \begin{array}{|c|c|c|}\hline .4 & .4 & .8 \\ \hline .4 & .1 & .8 \\ \hline .1 & .1 & .2 \\ \hline \end{array} \quad \hat{A}_{22} = \begin{array}{|c|c|c|}\hline 1.6 & 1.6 & 3.2 \\ \hline 1.6 & 0 & 3.2 \\ \hline .4 & .4 & .8 \\ \hline \end{array}$$

Proof: Assume a Lyapunov function in the following quadratic form,

$$V(x) = -\frac{1}{2} y^T P y, \tag{2.17}$$

where $P = D(A - I)$ is symmetric and consists of non-negative entries only. If there exists a $D = \mathbf{diag}\{d_1, d_2, ..., d_{MN}\}$ diagonal matrix which satisfies the symmetry criterion for P, the network is completely stable. (for the proof see appx. A.3)

Let us introduce a formalism,

$$a_{ij} = \begin{cases} (A)_{ij} - 1 & \text{, if} \quad i = j \\ (A)_{ij} & \text{, else .} \end{cases} \tag{2.18}$$

From the diagonality assumption on D, one can write $P_{ik} = d_i a_{ik}$, and from the symmetricity assumption on P, $P_{ik} = P_{ki} \Leftrightarrow d_i a_{ik} = d_k a_{ki}$.

From the principle of spatial invariance we have a Toeplitz type restriction on matrix A, (see Fig. 2.2(b)) The neighboring elements of D must obey the following relation,

$$\frac{d_i}{d_{i+1}} = \frac{a_{i+1,i}}{a_{i,i+1}} = \frac{a_e}{a_w}. \tag{2.19}$$

Consequently, every Nth neighboring elements are in relation

$$\frac{d_i}{d_{i+N}} = \frac{a_{i+N,i}}{a_{i,i+N}} = \frac{a_n}{a_s}.$$

Since the above holds for all i, the d_i/d_{i+1} constitute a geometrical series along the diagonal of D. (i.e. $d_i/d_{i+N} = (d_i/d_{i+1})^N \triangleq q^N$). As a consequence,

$$\frac{d_i}{d_{i+N+1}} = \frac{d_i}{d_{i+N}} \frac{d_i}{d_{i+1}} = \frac{a_{i+N+1,i}}{a_{i,i+N+1}} = \frac{\mathbf{a_{ne}}}{\mathbf{a_{sw}}} = \frac{\mathbf{a_n}}{\mathbf{a_s}} \frac{\mathbf{a_e}}{\mathbf{a_w}}$$

$$\frac{d_i}{d_{i+N-1}} = \frac{d_i}{d_{i+N}} \frac{d_i}{d_{i-1}} = \frac{a_{i+N-1,i}}{a_{i,i+N-1}} = \frac{\mathbf{a_{nw}}}{\mathbf{a_{se}}} = \frac{\mathbf{a_n}}{\mathbf{a_s}} \frac{\mathbf{a_w}}{\mathbf{a_e}}.$$

These relations describe the *constant ratio rule*. By writing A instead of $\tilde{A}$ (i.e. extension from one single layer to double layer), the consequences for 2LCNN can be examined.

The same rule applies for A_{11} independently of the other submatrix element in the hypermatrix. Having a geometrical series in $\tilde{D} = \{d1, ..., d_{MN}, d_{MN+1}, ..., d_{2MN}\}$ it leads to the following similar, but much trickier rules,

$$\frac{d_i}{d_{i+MN}} = \frac{a_{i+MN,i}}{a_{i,i+MN}} = \frac{a_c^{21}}{a_c^{12}}$$

Consequently,

$$\frac{d_i}{d_{i+N+MN}} = \frac{d_i}{d_{i+N}}\frac{d_i}{d_{i+MN}} = \frac{a_{i+N+MN,i}}{a_{i,i+N+MN}} = \frac{\mathbf{a_n^{21}}}{\mathbf{a_s^{12}}} = \frac{\mathbf{a_n^{11}}}{\mathbf{a_s^{11}}}\frac{\mathbf{a_c^{21}}}{\mathbf{a_c^{12}}}$$
$$\frac{d_i}{d_{i+1+MN}} = \frac{d_i}{d_{i+1}}\frac{d_i}{d_{i+MN}} = \frac{a_{i+1+MN,i}}{a_{i,i+1+MN}} = \frac{\mathbf{a_e^{21}}}{\mathbf{a_w^{12}}} = \frac{\mathbf{a_e^{11}}}{\mathbf{a_w^{11}}}\frac{\mathbf{a_c^{21}}}{\mathbf{a_c^{12}}}.$$

For diagonal compass points (i.e. south-western, south-eastern, north-western, north-eastern),

$$\frac{d_i}{d_{i+N+1+MN}} = \frac{d_i}{d_{i+1}}\frac{d_i}{d_{i+N}}\frac{d_i}{d_{i+MN}} = \frac{a_{i+N+1+MN,i}}{a_{i,i+N+1+MN}} =$$
$$= \frac{\mathbf{a_{ne}^{21}}}{\mathbf{a_{sw}^{12}}} = \frac{a_n^{11}}{a_s^{11}}\frac{a_e^{11}}{a_w^{11}}\frac{a_c^{21}}{a_c^{12}} = \frac{\mathbf{a_{ne}^{11}}}{\mathbf{a_{sw}^{11}}}\frac{\mathbf{a_c^{21}}}{\mathbf{a_c^{12}}} \tag{2.20}$$
$$\frac{d_i}{d_{i+N-1+MN}} = \frac{d_i}{d_{i-1}}\frac{d_i}{d_{i+N}}\frac{d_i}{d_{i+MN}} = \frac{a_{i+N-1+MN,i}}{a_{i,i+N-1+MN}} =$$
$$= \frac{\mathbf{a_{nw}^{21}}}{\mathbf{a_{se}^{12}}} = \frac{a_n^{11}}{a_s^{11}}\frac{a_w^{11}}{a_e^{11}}\frac{a_c^{21}}{a_c^{12}} = \frac{\mathbf{a_{nw}^{11}}}{\mathbf{a_{se}^{11}}}\frac{\mathbf{a_c^{21}}}{\mathbf{a_c^{12}}}.$$

The relation of Eq. (2.20) is depicted in Fig. 2.6(a)

By similar considerations one can get,

$$\frac{a_{se}^{21}}{a_{nw}^{12}} = \frac{a_{se}^{11}}{a_{nw}^{11}}\frac{a_c^{21}}{a_c^{12}} \qquad \frac{a_{sw}^{21}}{a_{ne}^{12}} = \frac{a_{sw}^{11}}{a_{ne}^{11}}\frac{a_c^{21}}{a_c^{12}}.$$

And from similar rearrangements, it is easy to show that

$$\begin{aligned}\frac{a_{se}^{21}a_{se}^{12}}{a_{nw}^{21}a_{nw}^{12}} &= \frac{\mathbf{a_n^{11}}}{\mathbf{a_s^{11}}}\frac{\mathbf{a_w^{11}}}{\mathbf{a_e^{11}}} = \frac{\mathbf{a_{nw}^{11}}}{\mathbf{a_{se}^{11}}} \\ \frac{a_{sw}^{21}a_{sw}^{12}}{a_{ne}^{21}a_{ne}^{12}} &= \frac{\mathbf{a_n^{11}}}{\mathbf{a_s^{11}}}\frac{\mathbf{a_e^{11}}}{\mathbf{a_w^{11}}} = \frac{\mathbf{a_{ne}^{11}}}{\mathbf{a_{sw}^{11}}}\end{aligned} \tag{2.21}$$

which can be depicted as in Fig. 2.6(b).

By examining the Block Toeplitz structure of A, one can conclude that the *spatial invariance* criterion that lead to the conclusions above, do not imply any relation between d_{MN} and d_{MN+1}. This would be particularly interesting so as to conclude whether D can be divided into two parts in which two independent geometrical series are formed or if it must form a single one.

To answer this controversial question, let us write what we used in the deduction of the rules above a bit more formally. The first step (Eq. (2.19)) can be written as

$$\frac{d_i}{d_{i+c}} = \frac{a_{i+c,i}}{a_{i,i+c}} = q^c, \tag{2.22}$$

which must hold regardless of the following substitution ($j = i + c$),

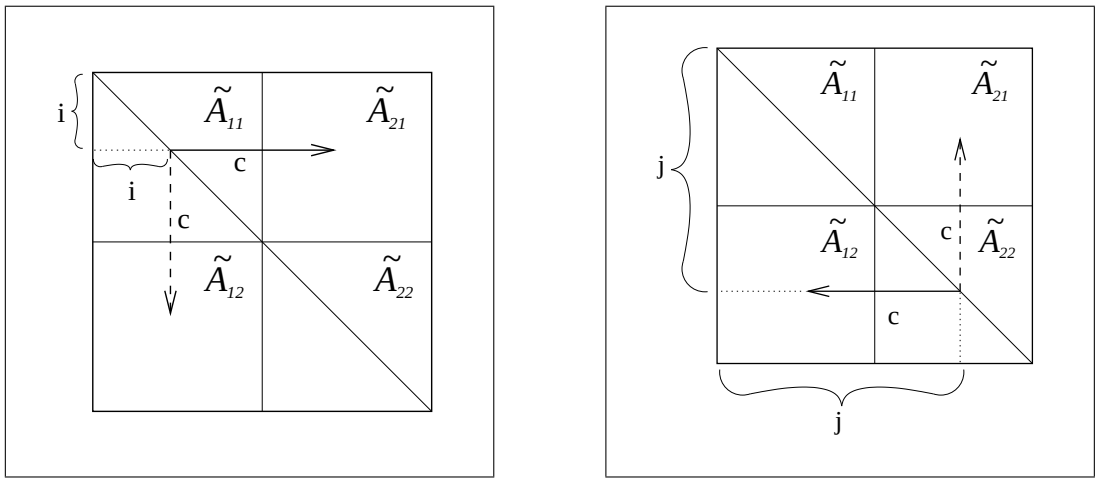
$$\frac{d_{j-c}}{d_j} = \frac{a_{j,j-c}}{a_{j-c,j}} = q^c \tag{2.23}$$

Roughly, if $c \geq MN$ then Eq. (2.22) (see Fig. 2.7(a)) leads to Eq. (2.21) and Eq. (2.20), while Eq.

(2.23) (see Fig. 2.7(b)) implies symmetric rules

$$\frac{a_{ne}^{21}}{a_{sw}^{12}} = \frac{a_{ne}^{22}}{a_{sw}^{22}} \frac{a_c^{21}}{a_c^{12}},$$

which completes the *contant ratio* rule in general.



(a) *Eq. (2.22)* (b) *Eq. (2.23)*

Figure 2.7: *The ratio of the elements pointed to by arrows in matrix ($\tilde{A}$) are in relations that are described by the corresponding equations*

2.6.4 τ-Non-Symmetric Case

Similarly to the theorem in Sec. 2.6.2, here we examine the effect of using non-unity layer time constants (τ), summarized in the following theorem as,

Theorem 4. *Theorem 3 holds for any ratio of layer time constants.*

Theorems which define a larger than zero set measure of region of parameters in which stability holds could only have importance in engineering practise. From this aspect, Theorem 4 significantly contributes to Theorem 3.

Proof: A Lyapunov function similar to Eq. (2.17) is used, but here $H = (A^* - \tau)$ which modifies Eq. (2.18) as,

$$a_{ij} = \begin{cases} (A^*)_{ij} - (\tau)_{ii} & ,\text{if} \quad i = j \\ (A^*)_{ij} & ,\text{else} \ . \end{cases} \tag{2.24}$$

Let's map matrix elements as $a^{ij*} \rightarrow a^{ij}$ as

$$\begin{aligned} a^{11*} &= \frac{1}{\tau_1} a^{11} \qquad a^{21*} = \frac{1}{\tau_1} a^{21} \\ a^{12*} &= \frac{1}{\tau_2} a^{12} \qquad a^{22*} = \frac{1}{\tau_2} a^{22} \end{aligned} \tag{2.25}$$

Hence, the rule above transforms Eq. (2.20) to

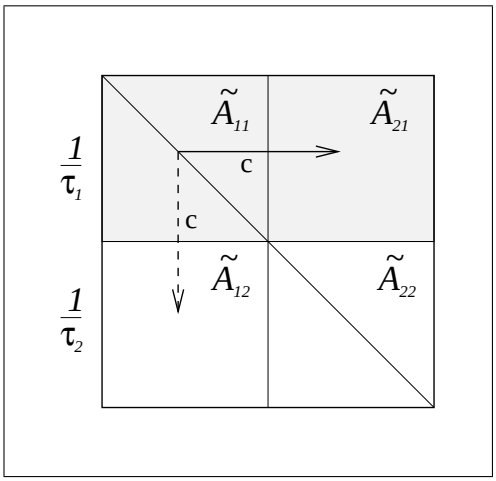


Figure 2.8: *The matrix A^* first used in Eq. (2.15) and later in Eq. (2.24) has this structure. Its submatrices are multiplied by $1/\tau_1$ and $1/\tau_2$ where they are gray and white, respectively. In the proof of Theorem 4 this matrix must be symmetrized by multiplying it by D in order to satisfy the corresponding hypothesis.*

$$\frac{a_{ne}^{21*}}{a_{sw}^{12*}} = \frac{a_{ne}^{11*}}{a_{sw}^{11*}} \frac{a_c^{21*}}{a_c^{12*}} =$$

$$\frac{\frac{1}{\tau_1}a_{ne}^{21}}{\frac{1}{\tau_2}a_{sw}^{12}} = \frac{\frac{1}{\tau_1}a_{ne}^{11}}{\frac{1}{\tau_1}a_{sw}^{11}} \frac{\frac{1}{\tau_1}a_c^{21}}{\frac{1}{\tau_2}a_c^{12}} = \frac{\mathbf{a_{ne}^{21}}}{\mathbf{a_{sw}^{12}}} = \frac{\mathbf{a_{ne}^{11}}}{\mathbf{a_{sw}^{11}}} \frac{\mathbf{a_c^{21}}}{\mathbf{a_c^{12}}} \tag{2.26}$$

which means that the mapping does not change the original statements.

The reason is simply as follows: during the mapping, $\tilde{A}$ is modified as in Fig. 2.8 and depending on the magnitude of c, this leads to two distinct cases in the equations which can be derived from the symmetry criterion on matrix P^*:

1. The $1 < c < MN$ case leads to rules, where the matrix elements referred to as $a_{i,i+c}$ and $a_{i+c,i}$ are in the same hypermatrix partition, so their ratio will not change by multiplying the elements of the partition by an arbitrary non-zero constant (i.e. $a^*_{i,i+c}/a^*_{i+c,i} = a_{i,i+c}/a_{i+c,i}$).

2. The $MN < c < 2MN$ case leads to rules which simplify the ratio exactly the same way as in Eq. (2.26).

Hence, one can see that τ values always simplify, which validates the theorem.

2.6.5 Sign Symmetric Case

The theorem introduced in ref. [CR90] and later used in ref. [CC92], prescribes the sign relation of template elements rather than the perfect relation of template values, which defines a much larger class of stable templates. Now, these findings are generalized to multi-layers.

Theorem 5. *A 2LCNN is **stable almost everywhere** if,*

- *Intra-layer feedbacks ($\hat{A}_{11}$,$\hat{A}_{22}$) are strictly sign-symmetric (see Fig. 2.10)*
  $\mathbf{sgn}(\hat{A}_{kk})_{ij} = (-1)^{i+j}$ *or*

$\mathbf{sgn}(\hat{A}_{kk})_{ij} = (-1)^i$ *or*
$\mathbf{sgn}(\hat{A}_{kk})_{ij} = (-1)^j$ *or*
$\mathbf{sgn}(\hat{A}_{kk})_{ij} = (-1)^0 \quad \forall k \in \{1, 2\}$

- *The sign of both of the inter-layer feedbacks are equal to that of the intra-layer feedbacks or its inverse (see Fig. 2.11)*
  $\mathbf{sgn}(\hat{A}_{k,3-k}) = -\mathbf{sgn}(\hat{A}_{kk})$ *or*
  $\mathbf{sgn}(\hat{A}_{k,3-k}) = \mathbf{sgn}(\hat{A}_{kk})$

The following template is an arbitrary example for the first template group in the second row of Fig. 2.11.

$$\hat{A}_{11} = \begin{array}{|c|c|c|}\hline -1 & .1 & -1 \\ \hline .1 & 1 & .1 \\ \hline -1 & .1 & -1 \\ \hline \end{array} \qquad \hat{A}_{12} = \begin{array}{|c|c|c|}\hline .2 & -2 & .2 \\ \hline -2 & 2 & -2 \\ \hline .2 & -2 & .2 \\ \hline \end{array}$$

$$\hat{A}_{21} = \begin{array}{|c|c|c|}\hline .3 & -3 & .3 \\ \hline -3 & 3 & -3 \\ \hline .3 & -3 & .3 \\ \hline \end{array} \qquad \hat{A}_{22} = \begin{array}{|c|c|c|}\hline -4 & .4 & -4 \\ \hline .4 & 4 & .4 \\ \hline -4 & .4 & -4 \\ \hline \end{array}$$

Convergence almost everywhere is a weaker condition than the *complete stability* because it can refer to unstable cases like equilibrium points, limit cycles and chaotic trajectories, though as a pathological only. The consequences define a much wider class of solutions than the previous theorems (say, a class of set of *non-isolated solutions*), which makes these theorems suitable for engineering purposes.

Definition 5. A dynamical system is stable *almost everywhere* if the set that does not converge to a constant equilibrium point has a Lebesque measure zero[3] (see ref. [CC92]).

In the proof of *Theorem 5* we will generalize the stability criterion given in ref. [CC92], following Hirsch's theorem in ref. [Hir85], claimed for cooperative/competitive systems.

Definition 6. A dynamical system $\dot{x} = F(x), F \in C^1$ is *cooperative* if

$$\frac{\partial F_i(x)}{\partial x_j} \geq 0 \quad \forall i \neq j$$

which means that the Jacobian matrix possesses the *non-negative off-diagonals only* property at every location.

Definition 7. A dynamical system $\dot{x} = F(x), F \in C^1$ is *irreducible* if a sequence of indexes from i to j exists ($\exists\{n_1 = i, n_2, n_3, ..., n_{m-1}, n_m = j\}$) such that

$$\Delta F(x)_{n_l, n_{l-1}} \neq 0, \, \forall i, j$$

where $l \in \{2, ..., m\}$ or equivalently, $\exists P$ permutation matrix such that, $P\Delta F(x)P^T = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}$).

[3]The definition of stable almost everywhere is used in Hirsch sense. [Hir85].

Let us consider it in case of a 2LCNN,

$$\frac{\partial F(x)_i}{\partial x_j} = (A)_{ij} \frac{\partial f_i(x)}{\partial (x)_i}$$

where

$$\dot{x} = F(x) = -x + Ay + Bu + z.$$

By finding a suitable transformation, our dynamical system can be transformed into a co-operative system. A solution is given by changing variables $x \to Jx'$, where J is chosen to be $\mathbf{diag}\{(-1)^{n_1}, (-1)^{n_2}, ..., (-1)^{n_{MN}}\}$ and $n_i \in (0, 1)$. Consequently, we get

$$J\dot{x}' = -Jx' + Af(Jx') + Bu + z \qquad | \quad J^{-1}. \tag{2.27}$$

$$\dot{x}' = F'(x') = -x' + J^{-1}Af(Jx') + J^{-1}Bu + J^{-1}z$$

As a consequence, the Jacobian matrix of F' is

$$\Delta F'(x') = J^{-1}A \; \Delta f(Jx')J - I$$

where $\Delta f()_{ij} = \delta_{ij} f_i'()$ and $I = \delta_{ij}$. Since they are diagonal, they commute and by a $A' = J^{-1}AJ$ substitution one can get,

$$\Delta F'(x') = J^{-1}AJ \; \Delta f(Jx') - I = A' \; \Delta f(Jx') - I.$$

Note that $J = J^{-1}$ which will ease the considerations later. The Jacobian matrix at every off-diagonal is,

$$\frac{\partial F_i'(x')}{\partial x_j'} = (A')_{ij} \frac{df(v)}{dv}\bigg|_{(Jx')_j}.$$

If A' is irreducible, non-negative at every off-diagonal position, the system is stable almost everywhere. Hence, we are looking for all appropriate J matrices, which might lead to an irreducible A' with non-negative off-diagonals. The task is now to find a methodology that generates all the possible solutions.

Note that irreducibility is intact under a JAJ affine transformation.

In general, rewriting an ODE equation into a compact form as it was done in case of the CNN (in Eq. (2.9), resulting in a matrix A), will not code all the constraints that the original system (i.e. CNN) had. These constraints include the principle of *spatial invariance* or diverse type of *boundary conditions*. As long as the principle of *spatial invariance* considerably decreases the degree of freedom of the matrix, the later constraint may alter only the structure of the *Block Toeplitz* matrix.

As a result of this investigation, we present all the possible solutions where Dirichlet boundary condition is assumed.

Every $J_{ii} = (-1)$ will result in a sign inversion of the corresponding (i.e. ith) column and (i.e. ith) row as depicted in Fig. 2.9.

Matrix A is not only Toeplitz but also sparse, which means that non-zero entries are located exclusively at diagonal and co-diagonal positions of the diagonal and co-diagonal blocks of the matrix. In case of a 2LCNN, the same applies to the intra-layer feedback and the inter-layer feedback blocks in the

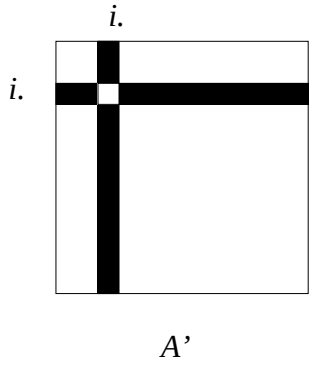


Figure 2.9: *The sign inversion of A under a JAJ transformation caused by a single negative entry (i.e.* (-1)*) in the diagonal of matrix J at the ith position, where black regions refer to the sign-inverted part of the matrix.*

hypermatrix. For a template size larger than 3×3, co-diagonals are filled up to the rth order.

First, for the sake of simplicity, we collected the solutions for a CNN of size MxN. In order to get spatial invariant solutions, realize that invariant co-diagonals must appear in the matrix A. In order to get this structure in A, we must have a periodic sign series in J (i.e. periodic n_i, too). The periodicity may be zero, unity, N or mixture of them (call it a *composite* later) which results in diverse but invariant, sign inverted blocks in matrix A.

One can conclude that only a few possibilities remain for J:

- *Identity* cases: $n_1 = 0, n_2 = 0, \ldots, n_{MN} = 0 \;\Rightarrow\; J = \{-1, -1, \ldots, -1\}$ which is equivalent to $n_1 = 1, n_2 = 1, \ldots, n_{MN} = 1 \;\Rightarrow\; J = \{1, 1, \ldots, 1\}$ since both projects A onto A' without any effect
- *Alternating* cases: $J = \{1, -1, 1, ..., (-1)^{MN}\}$ or $J = \mathbf{diag}\{-1, 1, -1, ..., (-1)^{MN}\}$ leads to chess table sign arrangement in A' which correspond to the vertically striped structures in the template
- *N block* cases: $J = \mathbf{diag}\{\underbrace{1, 1, ..., 1}, \underbrace{-1, -1, ..., -1}, \underbrace{1, 1, ..., 1}, ..., \quad (-1)^{M-1}\}$ and its inverse $J = \mathbf{diag}\{\underbrace{-1, -1, ..., -1}, \underbrace{1, 1, ..., 1}, \underbrace{-1, -1, ..., -1}, ..., \quad (-1)^{M}\}$ where multiple series of ones and minus ones follow each other in blocks of Ns. These Js produce chess table arrangements in the A' as well but with square sizes of N. These cases correspond to the horizonally striped structures in the template
- *Composite* of block N and alternating cases: Any multiplication of the *alternating* cases and the *block cases* leads to new solutions. These cases form the class of stable templates with chess table sign patterns.

The systematic sign-inversions in A correspond to sign inversions at the template sign patterns. We can visualize the rules above. If the sign pattern of a template matches to one of the sign patterns depicted in Fig. 2.10, the template belongs to the class of *stable almost everywhere* templates.

The transformation of the found configurations of Js to the case of 2LCNN leads to a larger set of stable template sign patterns which are given as follows.

The use of two independent layers ($\hat{A}_{12} = 0$ and $\hat{A}_{21} = 0$) recalls the Single-Layer CNNs' cases and implies the stability criteria specified above. Interesting questions arise whenever the two layers inter-connect via non-zero inter-layer feedback templates.

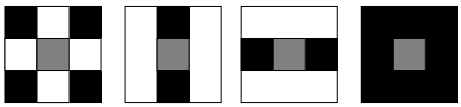


Figure 2.10: *These are the template sign symmetries of the spatially invariant solutions of* $(JAJ)_{ij} \geq 0$, *where* $J_{ij} = (-1)^{n_k}\delta_{ij}$. *The black regions refer to the sign-inverted part of the template. Non-negative, non-positive and arbitrary values are at black positions, white positions and gray positions respectively.*

While seeking for all possible configurations of Js of 2LCNNs, we will point out that one more case exists which multiplies the number of solutions of those found in the case of the Single-Layer CNNs' class.

- *MN block* case: $n_1 = 0, n_2 = 0, ...n_{MN} = 0, n_{MN+1} = 1, n_{MN+2} = 1, ...n_{2MN} = 1$ or its inverse. Both have the same influence. They result in an inversion on the off-diagonal matrices in hypermatrix A which corresponds to a sign inversion in the inter-layer feedbacks.

Next, for the generalization of the solutions given above, the parity of the CNNs size (MN) has to be fixed, since it affects the sign inversion patterns for inter-layer feedbacks (A_{ij} $i \neq j$).

As long as the even parity case prescribes the same patterns for both inter-layer feedbacks ($\hat{A}_{21}, \hat{A}_{12}$) and for the intra-layer feedback ($\hat{A}_{11}, \hat{A}_{22}$) patterns, the odd case inverts the sign patterns of the inter-layer feedbacks.

It may be easier to describe the sign inversions in J as a composition of waves. (e.g. $\sum_j \mathbf{sgn}(\mathbf{sin}(\omega_j i))$).

Considering the question above in this frame, we can describe it as a phase shift between the two halves of Js or an addition of a new wave the sum in the odd case. The wavelength of this wave should be 2MN.

Remark 3. The mixture of inversions where the first half of matrix J follows one pattern (or periodicity ω_j), while the other half follows another, does not lead to spatially invariant patterns in $\tilde{A}_{12}$ and in $\tilde{A}_{21}$, consequently they cannot result in a realizable inter-layer feedback template. For this reason these types of patterns must be excluded.

Remark 4. The class of stable sign patterns which can be generated from the considerations above results in templates that follow the same sign periodicity in both intra-layer and inter-layer feedback templates. Note that, they can be inverses of each other.

However, the odd case of MN inverts our claim for the inter-layer feedback, the parity of the size itself has no effect on this stability criterion at all. (see Fig. 2.11).

Remark 5. So far we have not considered the role of the layer time constants (τ) in this theorem with respect to its unity or non-unity. In fact, there is no need for any consideration since time constants can be only positive, hence no modification can have an effect on this theorem.

Invariant Topologies

In ref. [CR02], we can find two more solutions (see Fig. 2.12) that are hard to prove with the given formalism, however, can be generated by topologically invariant transformations.

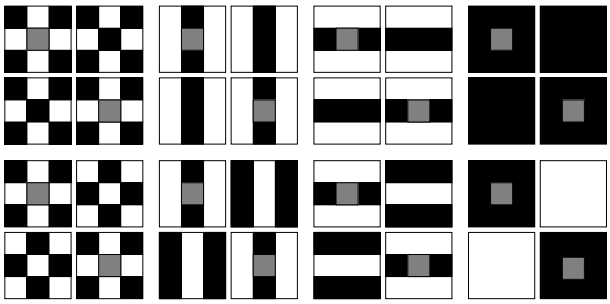


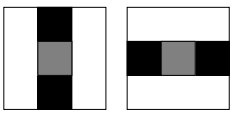
Figure 2.11: *The set of stable sign patterns. 2 by 2 groups refer to the intra-and inter-layer feedback templates' signs. Within each group, the templates are arranged in the following order: (A_{11} A_{21} ; A_{12} A_{22}).*

-	0	+
0	A	0
+	0	-

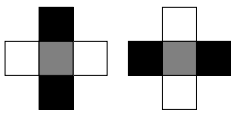
+	0	-
0	A	0
-	0	+

Figure 2.12: *Two sign patterns from the class of stable templates*

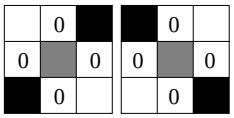
Considering the topological structure of the network of what these patterns seemingly assume, equivalence may be found between the displayed patterns and the sparse but sign symmetric CNNs. (see Fig. 2.13).



(a) *A stable CNN sign-pattern from Fig. 2.10*



(b) *An equivalent but sparse network realization of same sign pattern*



(c) *A topologically equivalent network obtained by rotation of (b)*

Figure 2.13: *The steps of transformation needed to get the patterns shown in Fig. 2.12.*

Without mentioning all the possible combinations of topologically invariant transformations applied on CNN and alike networks (see ref. [Rad00]), the rotated, but sparse version of stable patterns that can be obtained by the trick presented above are displayed in Fig. 2.14.

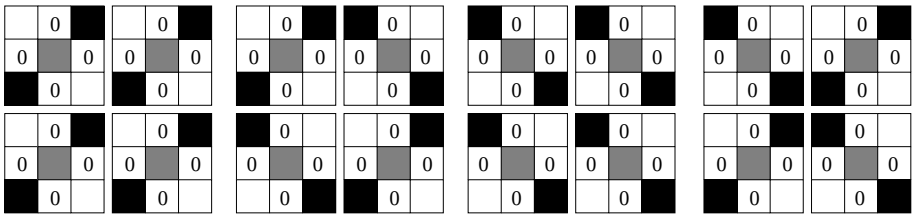


Figure 2.14: *All sign patterns generated by rules depicted in Fig. 2.13.*

Sign Patterns for Larger Templates

In the former sections, we restricted ourselves merely to double-layer architectures (2LCNN) with 3×3 template size. By assuming larger template sizes, the matrix responsible for describing dynamics of the

CNN in the first role will gradually lose its relative sparseness, hence it would complicate the deduction of most of the rules summarized in the theorems above. Due to the simplicity of the application of Hirsch's theorems and the wide range usability of the matrix concatenation technique in Eq. (2.10), theorems may be extended in the direction of even more layers and larger template sizes, resulting in rules which are still not too complicated to check against the templates.

The logic we follow will be roughly that

$$JAJ = funcion(i-j) \quad \Rightarrow \quad J = \mathbf{diag}\{(-1)^{n_1}, (-1)^{n_2}, (-1)^{n_3}, ..., (-1)^{n_N}\}. \tag{2.28}$$

In other words, we seek all solutions of Js composed of any series of power of (-1)s which is resulted by the Toeplitz, or more precisely *Toeplitz like*, conditions on JAJ.

Due to the Einstein formalism,

$$\mathbf{sgn}(JAJ) = \mathbf{sgn}(J_{ij}A_{jk}J_{kl}) = \mathbf{sgn}(\delta_{ij}J_i\,A_{jk}\,\delta_{kl}J_l) = J_iJ_l = \zeta \otimes \zeta^T$$

where vector $\zeta^T = [J_1, J_2, ..., J_n]$ and $\otimes$ symbolizes the diadic multiplication operator.

Since $J_iJ_l = J_lJ_i$, $\zeta \otimes \zeta^T$ is always symmetric, hence Eq. (2.28) may be written as,

$$\zeta \otimes \zeta^T = \mathbf{sgn}\left(function(|i-j|)\right) \tag{2.29}$$

Definition 8. A matrix with a *chess board type sign pattern* is defined as a matrix composed of rectangles of the same sign (zero included) from which none of the these rectangles are in contact with another of the same sign.

Lemma 1. *There exists a constructive way to determine ζ from $\zeta \otimes \zeta^T$, if and only if $\zeta \otimes \zeta^T$ is sign symmetric, has positive diagonal and has chess board type signs exclusively.*

No proof is required.

Only symmetric chess board type sign patterns in $\zeta \otimes \zeta^T$ satisfy the left-hand side of the Eq. (2.29) since any sign inversion in J causes sign inversion in the corresponding row and column at one time. Only diagonal sign patterns satisfy the right-hand side of the same equation. The two patterns that meet both of the statements above are the homogeneous one and the chess board signed one which are produced by constant (e.g. $J = \mathbf{diag}\{1, 1, 1, ..., 1\}$) or alternating series (e.g. $J = \mathbf{diag}\{-1, 1, -1, ..., 1\}$), respectively, or their inverse (already shown in Sec. 2.6.5).

Note that these special chess board types of matrices resemble to chess boards since the used rectangle size is unity.

To be more precise, one can realize that the constraint (i.e. right-hand side of the Eq. (2.29) above applies for a single-layer one-dimensional network exclusively where the matrix A is Toeplitz.

Concluding remarks on non one-dimensional structures involves Block Toeplitz matrices which extend the degree of freedom of ζ and of J as

$$\zeta \otimes \zeta^T = \mathbf{sgn}\left(function(\overbrace{|i-j|}^{\text{horizontal dependency}}, \overbrace{|[i/N]-[j/N]|}^{\text{vertical dependency}}) \right)$$

where $[.]$ denotes the integer truncation. The terms $[i/N], [j/N]$ refer to partitions in A that are needed to have homogeneous sign field.

Indeed, the searched unknown $function()$ is depending on $[i/N]-[j/N]$, which explains Toeplitz like sign alterations just like it was required for matrix elements in the one-dimensional case. This leads to the case that has already been shown in Sec. 2.6.5.

Continuing this methodology toward two-layers, one might write the MN block case as

$$\zeta \otimes \zeta^T = \mathbf{sgn}\left(function(|i-j|, |[i/N]-[j/N]|, |[i/MN]-[j/MN]|)\right) \tag{2.30}$$

which points much further in a constructive way.

For example the *alternating* and the *N block alternating* cases correspond to the alternating signs in templates in horizontal or in vertical directions that can be expanded for larger than 3×3 templates, like 5×5.

Later we will see Eq. (2.30) is only a special case which holds for two-layers only but not for larger networks.

Due to the concatenation technique introduced in Sec. 2.5, as the degree of freedom of $function()$ gradually increases, the number of solutions for J multiplies.

Now, we can claim our next theorem,

Theorem 6. *A 2LCNN with an arbitrary template size is* ***stable almost everywhere*** *if,*

- *The horizontal sign alteration frequency in all the templates are equal to one or zero up to the exception of the intra-layer feedback's central element,*

- *The vertical sign alteration frequency in all templates are equal to one or zero up to the exception of the intra-layer feedback's central element,*

- *The phase of the alteration are equal in groups* $\mathbf{sgn}(\hat{A}_{11}) = \mathbf{sgn}(\hat{A}_{22})$ *and* $\mathbf{sgn}(\hat{A}_{12}) = \mathbf{sgn}(\hat{A}_{21})$ *but may vary between groups.*

For better understanding the expression *alteration frequency* see Fig. 2.15.

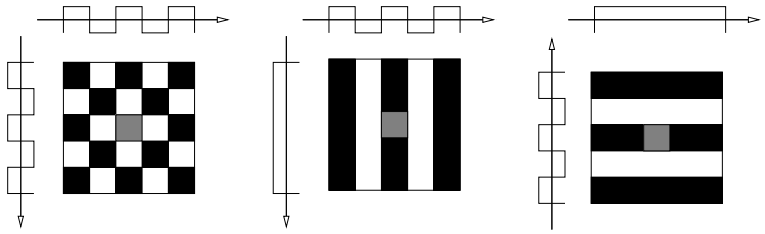


Figure 2.15: *Sign alteration frequencies of* 5×5 *templates are depicted. On the left-hand side, both the horizontal and the vertical sign alteration frequency are equal to one. In the center, horizontal sign alteration frequency is equal to one while vertical sign alteration frequency is zero. The one on the right-hand side is just opposite of the previous one.*

Below an example is given for the theorem above,

$$\hat{A}_{11} = \begin{array}{|c|c|c|c|c|}\hline -1 & 1 & -1 & 1 & -1 \\ \hline -1 & 1 & -1 & 2 & -1 \\ \hline -1 & 1 & .1 & 3 & -1 \\ \hline -1 & 1 & -1 & 4 & -1 \\ \hline -1 & 2 & -3 & 4 & -5 \\ \hline \end{array} \quad \hat{A}_{12} = \begin{array}{|c|c|c|c|c|}\hline 1 & -1 & 1 & -1 & 1 \\ \hline 1 & -1 & 1 & -2 & 1 \\ \hline 1 & -1 & 1 & -3 & 1 \\ \hline 1 & -1 & 1 & -4 & 1 \\ \hline 1 & -2 & 3 & -4 & 5 \\ \hline \end{array}$$

$$\hat{A}_{21} = \begin{array}{|c|c|c|c|c|}\hline 1 & -1 & 1 & -1 & 1 \\ \hline 1 & -1 & 1 & -2 & 1 \\ \hline 1 & -1 & 1 & -3 & 1 \\ \hline 1 & -1 & 1 & -4 & 1 \\ \hline 1 & -2 & 3 & -4 & 5 \\ \hline \end{array} \quad \hat{A}_{22} = \begin{array}{|c|c|c|c|c|}\hline -1 & 1 & -1 & 1 & -1 \\ \hline -1 & 1 & -1 & 2 & -1 \\ \hline -1 & 1 & -.1 & 3 & -1 \\ \hline -1 & 1 & -1 & 4 & -1 \\ \hline -1 & 2 & -3 & 4 & -5 \\ \hline \end{array}$$

Sign Patterns for More Layers

Considering it for more than two layers, the Toeplitz constraint does not hold as in Eq. (2.30) but we can write as,

$$\zeta \otimes \zeta^T = \mathbf{sgn}(function(|i-j|, |[i/N] - [j/N]|, |[i/MN]|, |[j/MN]|)) \tag{2.31}$$

The reason is that, the interaction of cells i and j must be fixed and be a function of the difference of the indexes (i.e. spatially invariant) as long as i and j are located in the same layer, the interaction of cells from the layer $[i/MN]$ and $[j/MN]$ may vary from layer-to-layer. For example, the interaction of the 1st layer to the 2nd not needed to be the same as the 2nd layer to the 3rd one. According to the given equation, they not even needed to be in a kind of sign-symmetric relation.

In relation to the identified multi-layer criteria, the left-hand side of Eq. (2.31) may be rewritten as

$$J^L \odot (\zeta_1 \otimes \zeta_1^T) = \zeta \otimes \zeta^T$$

where $\odot$ denotes the Kronecker matrix multiplication of a "small" sign modifier matrix J^L and the single-layer interaction sign-matrix $(\zeta_1 \otimes \zeta_1^T)$.

For example in a three-layer case, the *sign modifier matrix* may be

$$J^L = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

which explains that the interaction between layers $1, 2$ and $1, 3$ are in a kind of competitive (negative feedback) relation, while all the rests are in a kind of cooperative (positive feedback) relation (for details see [Hir85]).

Finally, summarizing this finding in a theorem:

Theorem 7. *A CNN with arbitrary number of layers and with arbitrary but fixed template size is* ***stable almost everywhere*** *if,*

- *The horizontal sign alteration frequency in all the templates are equal to one or zero up to the exception of the intra-layer feedback's central element,*
- *The vertical sign alteration frequency in all templates are equal to one or zero up to the exception of the intra-layer feedback's central element,*

- *Sign modifier matrix is chess board type, symmetric and positive diagonal (see the Lemma 1).*

2.6.6 Non-reciprocal Case

Definition 9. When the template values of a CNN are sign-anti-symmetric, the CNN is called *non-reciprocal*.

In ref. [CR90] the complete stability is conjectured and supported non-reciprocal CNNs. The studied dynamics of opposite sign valued template system cover a column-wise (or row-wise) coupling of cells exclusively, which means dynamics are row/column-wise independent of each other, thus it is called one-dimensional CNN or 1DCNN. For the summary, see appx. A.5.

Based upon very similar considerations on the structure of the Jacobian matrix, a new conjecture is given, which extends the former stability theorem (complete stability conjecture) to a column-wise (or row-wise) coupled 1D CACE architecture.

Definition 10. A one-dimensional CACE is composed of two inter-connected one-dimensional CNNs with intra-layer feedback of neighborhood radius one, and with scalar inter-layer feedback.

Conjecture 1. *A 1D CACE, will always settle in a saturated region and converge to stable equilibrium point if*

- *Inter-layer feedbacks are sign anti-symmetric,*
- *Intra-layer feedbacks are sign anti-symmetric,*
- *The central element of the intra-layer feedbacks is greater than one.* $(A_{kk})_c > 1$.

which means the stability of a non-reciprocal CACE architecture.

The interpretation of the criteria: The criteria above defines the following template structure:

$$\hat{A}_{11} = \begin{array}{|c|c|c|}\hline a & b & -c \\ \hline \end{array} \qquad \hat{A}_{12} = \begin{array}{|c|}\hline e \\ \hline \end{array}$$

$$\hat{A}_{21} = \begin{array}{|c|}\hline h \\ \hline \end{array} \qquad \hat{A}_{22} = \begin{array}{|c|c|c|}\hline j & k & -l \\ \hline \end{array}$$

where $a, b, c, j, k, l > 0$ and $e \cdot h < 0$.

Similarly to the Single-Layer 1D CNNs case, if any cell (k) reaches the saturated region (β), it will generate a -1 at kth (or at $(k+ML)$-th if it is on the second layer) diagonal position in the concatenated Jacobian matrix, and zeros above and below. A concatenated Jacobian matrix is constructed in a way similar to that of matrix A and is constructed from $\hat{A}$ in Eq. (2.10). Having zeros in a column with the exception of the diagonal, the corresponding eigenvalue can be explicitly given ($= -1$) and the system can be reduced by a factor of one by neglecting the corresponding row and column. *Chua, et al* in ref. [CR90] found a recursive method to calculate the determinant of the remaining band matrix ($\tilde{A}$), consequently they could point out that only positive eigenvalues exist in the reduced system. This implies that at least one cell will eventually reach the saturated region again from which it never returns.

A similar consideration applies to the CACE case, but no generative solution like this has been found so far for determinant calculation which is inevitable for a system of arbitrary size. Instead, we confirmed all possibilities by a symbolic software package and we claim:

For having positive principal minors of all possible configurations with an assumption on $\hat{A}_{11}$ and $\hat{A}_{22}$ as given above, a non-contradictory solution may exist if and only if $\hat{A}_{21}$ and $\hat{A}_{12}$ are of opposite sign scalars.

The example below generates the trajectory in Fig. 2.16.

$$\hat{A}_{11} = \begin{array}{|c|c|c|}\hline 1 & 2.5 & -1 \\ \hline \end{array} \qquad \hat{A}_{12} = \begin{array}{|c|}\hline -1 \\ \hline \end{array}$$

$$\hat{A}_{21} = \begin{array}{|c|}\hline 1 \\ \hline \end{array} \qquad \hat{A}_{22} = \begin{array}{|c|c|c|}\hline 1 & 2.5 & -1 \\ \hline \end{array}$$

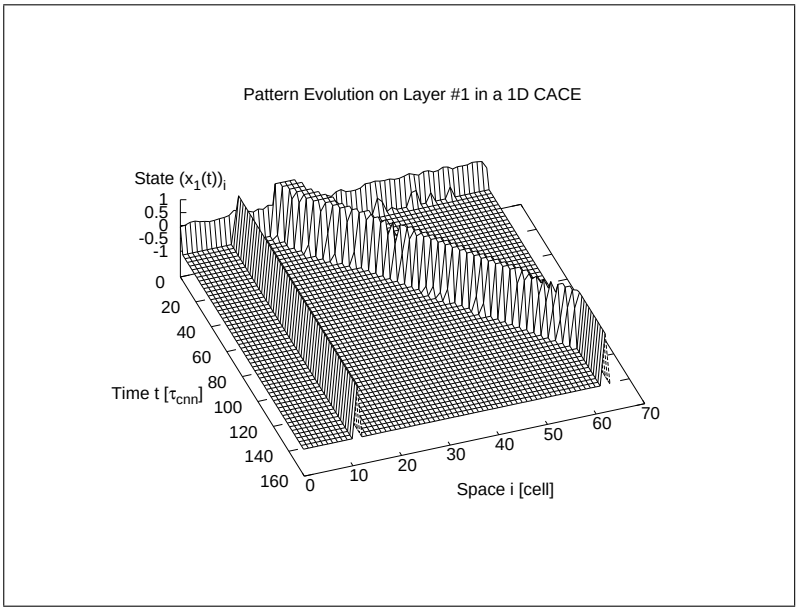


Figure 2.16: *The time evolution of the one-dimensional CACE structure with a non-reciprocal template.*

2.7 Conclusion

A description of a complex cell CNN-UM architecture (CACE) along with the MLCNN is given. This is motivated a survey on the possibilities of generalization of stability theorems from Single-Layer architectures to Multi-Layer architectures.

The power of a new formalism was demonstrated by claiming several theorems and one conjecture on multi-layer architectures inherently including the CACE architecture.

By means of constructed new Lyapunov functions the layer time constants dependence of the theorems was made possible.

The inference of the formalism we developed is proved to be useful in gerenalization of many theorems and Lyapunov functions. Its consequences are not completely explored, therefore, it is worth carrying out further investigations in the class of multi-layer nerual architectures.

Chapter 3

Analog-VLSI, Array-Processor Based, Bayesian, Multi-Scale Optical Flow Estimation

3.1 Brief

Optical flow (OF) estimation aims at derive a motion-vector field that characterizes motions on a video sequence of images. In this paper we propose a new multi-scale (or scale-space) algorithm that generates OF on Cellular Neural/Non-linear Network Universal Machine (CNN-UM), a general purpose analog-VLSI hardware, at resolution of 128x128 with fair accuracy and working over a speed of 100 frames per second. The performance of the hardware implementation of the proposed algorithm is measured on a standard image sequence. As far as we are concerned, this is the first time when an OF estimator hardware is tested on a practical-size standard image sequence.

3.2 Notation and Symbols

AGC	Automatic Gain Control
aVLSI	Analog Very Large Scale Integrated devices
AAE	Average Angular Error (degree)
AVE	Average Vector norm Error (pixels)
BC	Boundary Condition (of CNN)
BM	Block Matching (criteria)
$\Delta\bar{p}$	displacement vector
Δt	time difference between two subsequent frames
DFD	Displaced Frame Difference
DFD_q	Displaced Frame Difference at scale q
DFD_Gauss	Displaced Frame Difference with Gaussian windowing function
EMD	Elementary Motion Detector
FPGA	Field Programmable Array

fps	frames per second (measure)
I	Image
$I^{(i)}$	the ith Image in a sequence
OF	Optical Flow
OFC	Optical Flow Constraint
$\bar{p}$	Position in an image (measured in pixels)
PDE	Partial Differential Equation
PDF	Probability Distribution Function
r	Radius
$R^q_{\bar{p}}$	Region at $\bar{p}$ of scale q
SSD	Summed Squared Difference
STDD	STandarD Deviation
(x, y)	position in an image
$\bar{v}$	Velocity
VSS	Visual Sensory System
W	Windowing function
$*$	convolution operator

3.3 Introduction

In general, a visual sensory system (VSS) can be interpreted as a system that records light in a 2 dimensional (image) plane reflected by illuminated, 3 dimensional, real-world objects. The objects of the environment that are subject to motion relative to the observer induce a motion-vector field in VSSs. The objective of Optical Flow (OF) calculations is to estimate this vector field, which describes the transformation of one image to the consecutive one.

Far from being exhaustive, a review on existing OF methodologies will be given along with relation to the proposed method in sec. 3.4

In a simplistic approach one can classify techniques used in OF deduction depending on the basis as optical flow constraint (OFC), gray value matching (e.g. block matching (BM)), correlation (ref. [Ana89]), phase (cross correlation), higher order statistics (ref. [BJ96, SL01]), or energy [Hee88, SRC98].

In spite of the fact that BM approaches possess considerable disadvantages such as relatively high computational needs (i.e. quadratic in block size and quadratic in maximal vector size), BM has several advantages.

Firstly, in case of large displacements (larger than the support used for calculating the OFC) the validity of OFC is questionable since the higher order terms (H.O.T.) of the Taylor expansion becomes a significant factor that is neglected in OFC. Secondly, it is very simple to be calculated and formulated. Thirdly, it overcomes the OF ambiguities (only) within the applied block size (sec. 3.6) at the expense of the vector field bandwidth. Forthly, it is more robust against local intensity flicker than differential or variational techniques. Fifthly, it is essentially parallel, which might indicate the applicability an already implemented general purpose, analog-VLSI network, the Cellular Neural/Non-linear Network Universal-Machine (CNN-UM) (sec. 3.8.2) . Sixthly, unlike OFC based methods, it does not rely on non-neural operations such as division or matrix inversion.

For these reasons we turned to the BM type of approaches and we have found that:

- by the application of diffusion to multiple diffusional lengths (or equivalently time durations) a *multi-scale* view can be generated which method involves a cheap ($\sim \mu sec$) CNN-UM operation only

- these views are incorporated in a Bayesian optimal estimation (sec. 3.7) of the OF, which resolves the trade-off between the spatial bandwidth and accuracy (sec. 3.6) and leads to sub-pixel accuracy (sec. 3.9) outperforming many of the known models, as our simulation and measurements show.

It is difficult to compare the computational complexity of algorithms running on analog massively parallel architecture versus ones running on digital sequential hardware but we made an effort and found that: The powerfulness of the new computational model implemented on CNN-UM versus Neumann computer architecture is pointed out. The performance gain is in the range of $(O(N^3))$, due to its strongly parallel computing architecture (see appx. B.1, appx. B.2) which makes it competitive in this aspect, too.

N.B. motion sensing and OF are essentially the same. Both aim at determining the motion vector field but they can determine the optical flow only. OF is the perceivable vector field, while the motion vector field is the projection of 3D motions onto the focal plane. Motion sensors were, usually, designed considering local information only and are not designed to be able to cope with the blank wall or the aperture problems. Contrary, OF solution are hoped to be able to, to a certain extent. There are numerous aVLSI motion sensors available today (ref. [HKLM88, HKSL90, KM96, ECdSM97, KSK97, Liu98, HS02, RF04]) and there only few OF chips (ref. [SRC98, SD04, MEC04]). There is also an FPGA based soft solution implementing an old theory (ref. [MZC$^+$05]), published recently.

Opposed to the special-, single-purpose circuitry listed above, we used a general purpose aVLSI circuitry. The performance results are demonstrated in sec. 3.9. First in the history a VLSI solution is tested on a standard optic flow test sequence (Yosemite) that is common in vision community. The simulations show performance not very high compared to elaborated numerical solutions such as variational or differential methods but still, the solution seems to be satisfactory and is implementable on a parallel, general purpose aVLSI hardware, which is a great advantage. Interestingly, the chip measurements nicely converge to the theoretical limit that was given by simulations. The speed of the solution is above 100 frames per second (fps). By increasing the quality of the circuitry used, this may be increased by at least one order of magnitude in the near future.

3.4 Problem formulation and solutions

The problem may be written in several forms. Depending on the formulation, diverse types of solutions coexist that may solve the formalized problem effectively; however, they might diverge from the real OF. Consequently the formulation and the solutions, usually, involve heuristics. Here, we wish to give a brief summary based on a generic functional minimization.

The first and most inevitable assumption in the formulation is that exists a spatio-temporal persistency and a continuity of the brightness values or patterns in the image. This may be coded in a term called *brightness constancy (BC)*,

$$||dI|| \to 0 \tag{3.1}$$

By keeping the first term of the Taylor expansion of equ. 3.1, it results in $\left|\left|\bar{v}\nabla I - \frac{\partial I}{\partial t}\right|\right| \to 0$, where $\bar{v}$ and t denotes the velocity and the time, respectively. This is, usually, referred to as *OF Constraint (OFC)* or

the differential form of BC. Equ. 3.1 may be written in time-discretized form as $|I^{(1)}(\bar{p}) - I^{(2)}(\bar{p}+\bar{v})| \rightarrow 0$, where $\bar{p}$ denotes the position in the subsequent images $I^{(1)} = I(t)$ and $I^{(2)} = I(t-\Delta t)$, respectively. This last equation is commonly referred to as block-matching (BM) criteria.

Unfortunately, solving any form of equ. 3.1 for $\bar{v}$, it does not give a unique solution in all cases due to the fact that there is only one linear constraint on a two-dimensional vector field. For this reason, regularized solutions were formalized with additional constraints ensuring smoothness and discontinuity of the vector field: $\phi(\nabla u) \rightarrow 0$ and $\phi(\nabla v) \rightarrow 0$, where $\bar{v}^T \triangleq (u, v)$, and $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a suitable non-linear function[1] with this double objective.

In real-world measurements, noise is a limiting factor difficult to avoid. Methods derived from the considerations above may suffer from noise sensitivity. They can be extended with noise rejective terms $c(\nabla I)$ which penalizes noise both in the image and as a result in the vector field, too.

The generic solution may be searched as minima (e.g. $\bar{v} = \arg\min_{\bar{v}} E(\bar{v})$) of an integral over the image (Ω)

$$E(\bar{v}) = \underbrace{\int_\Omega \left|\left| \bar{v}^T \nabla I + \frac{\partial I}{\partial t} \right|\right| d\bar{p}}_{A} + \alpha^\tau \underbrace{\left[\int_\Omega \phi(\nabla u) d\bar{p} + \int_\Omega \phi(\nabla v) d\bar{p} \right]}_{B} + \alpha^h \underbrace{\int_\Omega c(\nabla I) ||\bar{v}|| d\bar{p}}_{C}, \tag{3.2}$$

where α^τ and α^h are weighting constants.

In this functional, other functionals may be used as a plug-in replacement for terms meeting the objective of the original term. Hence, term A may be replaced by the integral of BM instead of that of the OFC, or term B may be extended with additional constraints too. In ref. [SJ96, GL96] authors found it important to prevent vector field from rotation and divergence. (e.g. $\int_\Omega \phi(\mathbf{div}(\bar{v}), \mathbf{rot}(\bar{v})) d\bar{p}$)

In its generic form, it is nearly impossible to have a fast and uniquely converging minimization of equ. 3.2; however, most OF estimators, which we are currently aware of, can be fit in this equation.

Lucas and Kanade ([LK81]) provided the following form in early times of OF estimators

$$E(\bar{v}) = \int_\Omega W * \left|\left| \bar{v}^T \nabla I + \frac{\partial I}{\partial t} \right|\right| d\bar{p},$$

where a Gaussian convolution kernel W was used for providing smoothness of $\bar{v}$, that has extended the regions in terms of $\bar{p}$ where unique solutions exist. While the solution can be given directly by a matrix inversion (if non singular) the solution of the Horn Schunk model (i.e. $E(\bar{v}) = \int_\Omega \left|\left| \bar{v}^T \nabla I + \frac{\partial I}{\partial t} \right|\right|_2^2 d\bar{p} + \alpha^\tau \int_\Omega ||\nabla u|| + ||\nabla v|| d\bar{p}$) can be approached iteratively, only [HS81].

A comprehensive analysis on the properties of functions left open in equ. 3.2, on the set of admissible solutions (existance, unicity) along with a high quality solution is given in ref. [ADK99, AK02]. Unfortunately, there is still a very hard trade-off between OF quality and processing time. The partial differential equations (PDE) dropping out from the analysis of the equations above, usually, converge very slowly, hence they may not be applied in applications where real-time processing is a criterion by means of limited power and in limited space. However, interesting simplifications may be made, which

[1] Different ϕ's were found to be effective in OF calculations. A short list of the well-known versions is given.

Author	$\phi(s)$
Geman and Reynolds	$\frac{s^2}{1+s^2}$
Malik and Perona	$\log(1+s^2)$
Green	$2\log(\cosh(s))$
Aubert	$2\sqrt{1+s^2}-2$

opens space for invention in analog chip design. Many motion- and OF-chips were fabricated in the past. Without going into details, they share many common properties such as:

- they are made for special purpose,
- they are, usually, called bio-inspired or neuromorphic due to their on-chip photo-sensors and cellular structure (only local connections exist in a matrix structure) resemble to the retina or the primary visual cortex in most engineers' perspective,
- they operate in discrete space and continuous time,
- sometimes even the off-chip communication is made asynchronously via Address Event Representation (AER),
- VLSI elements are used either in mixed or mainly in analog (non saturated) mode,
- they either realize the corresponding Euler-Lagrange equation that are in loose relationship with considerations above or are simple ad-hoc motion sensors (e.g. Elementary Motion Detector)
- the results are rather noisy.

In fact, the circuitry we relied on (CNN) shares many of these charateristics. It is called "bio-inspired", cellularly structured, implemented in analog VLSI, operates in continiuous time, it is compact. But, in contrast to all of them, it is proved to be general purpose (via Turing equivalence) leaving space for analogic programmability. At the first look, CNN may be perceived as a well parameterizable PDE solver. The most straightforward approach would imply solutions that realize some of these PDEs on CNN-UM. This can be done by decomposing the PDE based solution into CNN friendly system of equations. The raising difficulties are, usually, two-fold. On the one hand, the non-linear terms of the resulted PDEs, usually, require calculations of divisions and/or H.O.T. On the other hand, the convolutional structure of the CNN (which involves the spatial invariant criteria) is, usually, broken by the PDEs resulted.

In our solution, we constructed a stochastic framework in which the maximum of the probability distribution function (PDF) $p(v|I^{(1)}, I^{(2)})$ was searched (i.e. $\bar{v} = \arg\max_{\bar{v}} p(v|I^{(1)}, I^{(2)})$). This was done to let multiple-scale view of the images cooporate in a non-deterministic way and lead to a Bayes optimal resolution of the trade-off between spatial bandwidth and accuracy (see the next section).

Interestingly, the algorithm obtained by the considerations above uses very simple parallel operations that can be executed on CNN-UM and exploits its fully parallel nature. The experiments show that it is extremely accurate on simple tests (note the sub-pixel accuracy in fig. 3.6) and on standard tests it remains fair (even on aVLSI) compared to complex digital implementations. Furthermore, it operates over 100 fps, which reserves a unique place for it.

3.5 Block or Region Matching Based Techniques

The major advantages of using BM is the ability to successfully suppress image flicker and its simplicity in hardware realization.

Definition 11. *Region* or *block* is a collection of pixels that are located within the boundaries of size r around $\bar{p}$.

$$R^q_{\bar{p}} = \{(x, y) : ||(x, y) - \bar{p}||_\infty < r \, , \, r = 2^q\},$$

where q refers to the size of the region in logarithmic scale. Without loss of generality we may use any type of norm but for practical reasons the applied norm is l^∞ or maximum norm.

Definition 12. The measure *Summed Squared Difference* of two images at $\bar{p} = (x, y)$ in a given region is,

$$SSD(I^{(1)}, I^{(2)}, R^q_{\bar{p}}) = \frac{1}{|R^q_{\bar{p}}|} \sum_{\bar{\xi} \in R^q_{\bar{p}}} ||I^{(1)}(\bar{\xi}) - I^{(2)}(\bar{\xi})||_2.$$

Without loosing most of the important characteristics of the squared differences, we can also use the norm l^1, which is particularly important in CNN implementation. Note that $|R^q_{\bar{p}}|$ is not a norm but the size of the set (i.e. the area of the region) which is, obviously, constant for a fixed q.

The simplest *BM approach* would mean a search for the best match or fit of a particular region in an image ($I^{(1)}$) around a pixel ($\bar{p}$) by offsetting the region by $\Delta\bar{p} = \bar{v} \cdot \Delta t$ in the image ($I^{(2)}$). The measure is called Displaced Frame Difference (DFD) (see fig. 3.1) and can be defined as

$$DFD(I^{(1)}, I^{(2)}, \Delta\bar{p}, R^q_{\bar{p}}) = \frac{1}{|R^q_{\bar{p}}|} \sum_{\bar{\xi} \in R^q_{\bar{p}}} |I^{(1)}(\bar{\xi} + \Delta\bar{p}) - I^{(2)}(\bar{\xi})|. \tag{3.3}$$

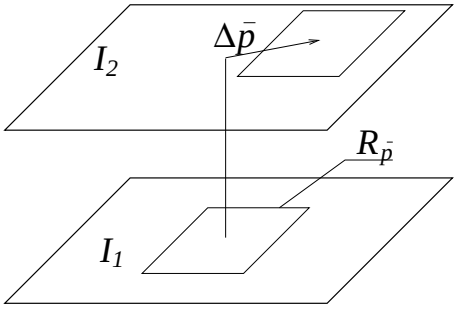


Figure 3.1: Displaced Frame Difference between two frames at a certain position ($\bar{p}$) at a given displacement ($\Delta\bar{p}$) of a region (R) is calculated as $\frac{1}{|R^q_{\bar{p}}|} \sum_{\bar{\xi} \in R^q_{\bar{p}}} |I^{(1)}(\bar{\xi} + \Delta\bar{p}) - I^{(2)}(\bar{\xi})|$

To avoid aliasing effects windowing or pre-filtering is commonly applied. We used a Gaussian window function in its simplest form so as to define the Gaussian DFD measure as,

$$\begin{aligned} DFD_{Gauss}(I^{(1)}, I^{(2)}, \Delta\bar{p}, R^q_{\bar{p}}) &= \frac{1}{|R^q_{\bar{p}}|} \sum_{\bar{\xi} \in R^q_{\bar{p}}} |I^{(1)}(\bar{\xi} + \Delta\bar{p}) - I^{(2)}(\bar{\xi})| \cdot e^{-\alpha \left\|\bar{\xi} - \bar{p}\right\|_2^2} \\ &= |I^{(1)}(\bar{p} + \Delta\bar{p}) - I^{(2)}(\bar{p})| * W, \end{aligned} \tag{3.4}$$

where α is a constant parameter to adjust.

Please note that l_1 norm becomes particularly useful since the equ. 3.4 can be written as

$$DFD_{Gauss}(I^{(1)}, I^{(2)}, \Delta\bar{p}, R^q_{\bar{p}}) = |I^{(1)}(\bar{p} + \Delta\bar{p}) * W - I^{(2)}(\bar{p}) * W|.$$

The objective of OF estimation is to find

$$\bar{v}\Delta t = \arg\min_{\Delta\bar{p}} DFD_{Gauss}(I^{(1)}, I^{(2)}, \Delta\bar{p}, R^q_{\bar{p}}),$$

where $\bar{v}$ refers to velocity which can be determined as $\bar{v} = \Delta\bar{p}/\Delta t$. It simply means that the algorithm should find the displacement of every pixel with the smallest cost. Here the cost should be meant in terms of energy or DFD (above).

Block matching approaches sometimes use *correlation* maxima which has the advantage of being less sensitive to illumination alterations compared to DFD methods but they have the disadvantage of increased calculation needs involved the multiplication. In restricted circumstances we can claim that seeking for the minima of DFD_{Gauss} expression maximizes the correlation (equ. 3.5) as well

$$Corr_{Gauss}(I^{(1)}, I^{(2)}, \bar{v}, R^q_{\bar{p}}) = \frac{1}{|R^q_{\bar{p}}|} \sum_{\bar{\xi} \in R^q_{\bar{p}}} I^{(1)}(\bar{\xi} + \Delta\bar{p}) \cdot I^{(2)}(\bar{\xi}) \cdot e^{-\alpha \left\| \bar{\xi} - \bar{p} \right\|_2^2} = I^{(1)}(\bar{p} + \Delta\bar{p}) \cdot I^{(2)}(\bar{p}) * W \quad (3.5)$$

For this reason DFD based methods are, usually, referred to as correlation based techniques and we will refer to them interchangeably.

For the sake of short writing, we will refer to the measure of matching (DFD) as $DFD(I^{(1)}, I^{(2)}, R^q_{\bar{p}}, \bar{v}) \Leftrightarrow DFD_q(I^{(1)}, I^{(2)}, \bar{v})$, where q refers to the scale that is exponentially related to the window size (R) (i.e. with $q = 0, 1, 2, ...$ refers to region of $R^q_{\bar{p}} = 1x1, 2x2, 4x4, ...$) as mentioned in definition 11.

When larger Gaussian window (W in equ. 3.4) is applied, thanks to the averaging characteristics of the window, will result in an image with degraded spatial bandwidth.

The blur caused by these integrating type of filters (such as window functions) is equivalent to applying a bandwidth degrader or changing the view of an image to a larger scale (see appx. B.1). In the algorithm we propose, we combine multi-scale (i.e. sections of the scale space) information of a single image but we will not estimate the optimal scales to use (see ref. [Wit83]).

3.6 Ambiguities in Optical Flow Techniques

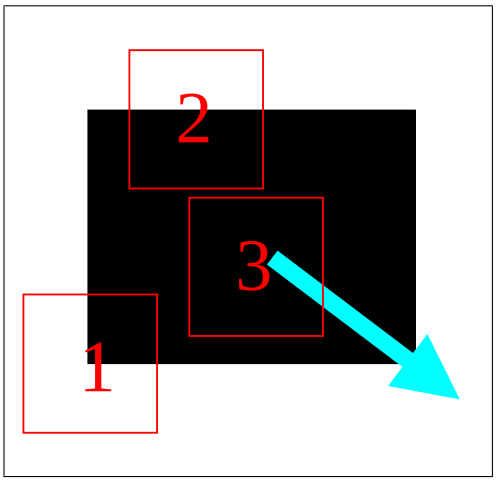
The ambiguities that spoil the measurement of the OF may be explained in terms of block matching, too; There are two typical cases of the ambiguities. In a thought experiment, assume a moving rectangle as indicated in fig. 3.2(a)) which translates in the direction pointed by the arrow. Having a correlation window (i.e. block) of size depicted in the figure one can conclude that correlation maxima will provide correct information at the corners (1) of the rectangle. However, there is no correlation peak inside the black rectangle (3) and the correlation maxima is not an isolated point but a line around the edges (2). In the latter case, the normal flow might be calculated only, which seems to be the best guess based on the local information but one can see it is not optimal in a global sense. These problems they are called as *blank wall problem* and *aperture problem*, respectively.

Considering these error prone cases, the moving object can be partitioned into regions where 1) the correct optical flow can be calculated (light grey area), 2) the aperture problem appears (dark grey area) or 3) the blank wall problem appears (black area) (see fig. 3.2(b)).

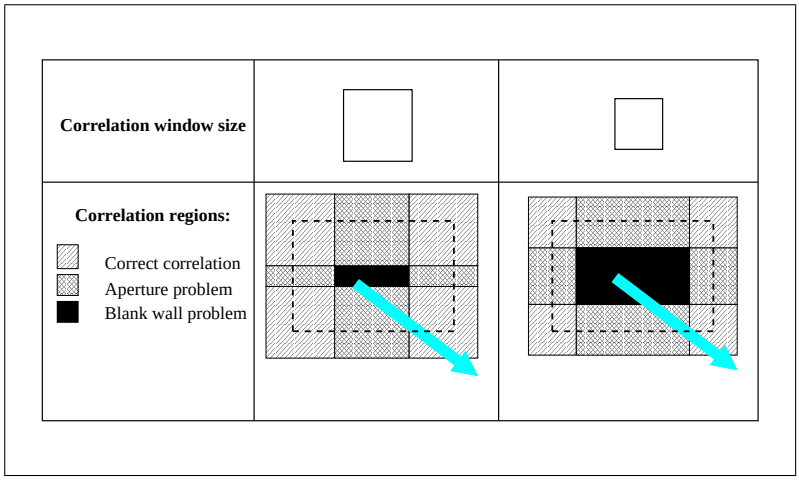
To overcome these problems one might suggests to increase the correlation window size but it would lead

- to the gradual decrease of bandwidth and,
- to the quadratic increase in computational needs (and time).

This dilemma is called the **bandwidth/accuracy trade-off**.



(a) Assume a moving rectangle in the pointed direction. Correlation maxima identifies the direction of motion in the corners (1) but it is ambiguous at the edges (2) and there is no maxima at all in the center regions (3).



(b) Partitions (slash, cross, filled patterns) around the hypothetic moving rectangle (dashed) where no problem, aperture problem and blank wall problem appears, respectively, depending on a given correlation window size (large on the left, small on the right).

Figure 3.2: Identification of typical problems appearing in OF

In our model, the dilemma is hoped to be gradually removed. In the hierarchic calculation schema (explained later in sec. 3.7) the CNN provides an alternative computational scheme that solves the Gaussian shaping and correlation simultaneously in **square root increase** of time.

The similarity between this dilemma and equ. 3.2 should be noted. Equ. 3.2 is written in a form to overcome the OFC ambiguities by means of the neighborhood while keeping the OFC term minimal.

3.7 Bayesian Incorporation of Scales

According to the considerations of the previous section, one can see that the main source of distortions in the search for DFD minima (or correlation maxima) is rooted in the applied measurement technique. One

has to note that using the differential form of the OFC does not help. The problem appears similarly, as a singularity of the corresponding matrices. To overcome these problems, we propose a stochastic model by associating a measure of fitness (DFD) with a similarity likelihood as,

$$p_q(I^{(2)}|\bar{v}, I^{(1)}) \propto exp\left\{-DFD_q(I^{(1)}, I^{(2)}, \bar{v})\right\}. \tag{3.6}$$

There are several notes we have to make here:

- The form of the fitness (DFD) is somewhat arbitrary which could be appropriate in many other forms.
- The only property we expect from the likelihood is to be inversely proportional to this particular fitness.
- This likelihood defines a measure proportional to the likeliness of being the correct warp of a region of size $2^q x 2^q$ from one image to the other. (e.g. the normalization would be over the $I^{(2)}$).

Based on this likelihood, by Bayes theorem for the *aposterior* we have,

$$p_q(\bar{v}|I^{(1)}, I^{(2)}) = \frac{p_q(I^{(2)}|\bar{v}, I^{(1)})\, p_q(v|I^{(1)})}{p_q(I^{(2)}|I^{(1)})} \propto p_q(I^{(2)}|\bar{v}, I^{(1)})\, p_q(v|I^{(1)})$$

Since the denominator does not depend on $\bar{v}$ for which we are to maximize this expression, we can drop it. As it is common in Bayesian decisions, having a good *prior* hypothesis is crucial in making reasonable decisions. This is the Ockham's razor. In OF calculus, the question of good *prior*s had left space for a lot of debates in the literature too (ref. [AB86]).

Thinking in multiple scales, from the study of ambiguous situations, we might conclude that the prior can be the larger scale $(q+1)$ estimation of the OF. This helps to reconstruct the OF at a given (q) scale with propagating the information available at the larger scale. (i.e. associate $p_q(v|I^{(1)}) \leftarrow p_{q+1}(\bar{v}|I^{(1)}, I^{(2)})$) as,

$$p_q(\bar{v}|I^{(1)}, I^{(2)}) \propto p_q(I^{(2)}|\bar{v}, I^{(1)})\; p_{q+1}(\bar{v}|I^{(1)}, I^{(2)}).$$

This recursive scheme can be written in explicit form as,

$$p(\bar{v}|I^{(1)}, I^{(2)}) \propto p_0(\bar{v}) \prod_q p_q(I^{(2)}|\bar{v}, I^{(1)}) = p_0(\bar{v})\; \exp\left[-\sum_q DFD_q(I^{(1)}, I^{(2)}, \bar{v})\right].$$

Certainly, we are interested in the vector field (or OF) at the finest scale which is,

$$\bar{v} = \arg\max_{\bar{v}} p(\bar{v}|I^{(1)}, I^{(2)}) = \arg\min_{\bar{v}} \left[-\log p_0(\bar{v}) + \sum_q DFD_q(I^{(1)}, I^{(2)}, \bar{v})\right]. \tag{3.7}$$

Note that, however, we formulated the problem is in the log-domain, by disregarding the constant first term in equ. 3.7, the calculation to be done here remains as simple as addition of DFDs and an extremum search.

An alternative formulation was suggested by Cs. Szepesvari in a personal communication leading essentially to the same results (For details see appx. B.4).

The resulted calculation scheme defines a Bayesian incorporation of the multi-scale measurements. (see fig. 3.3)

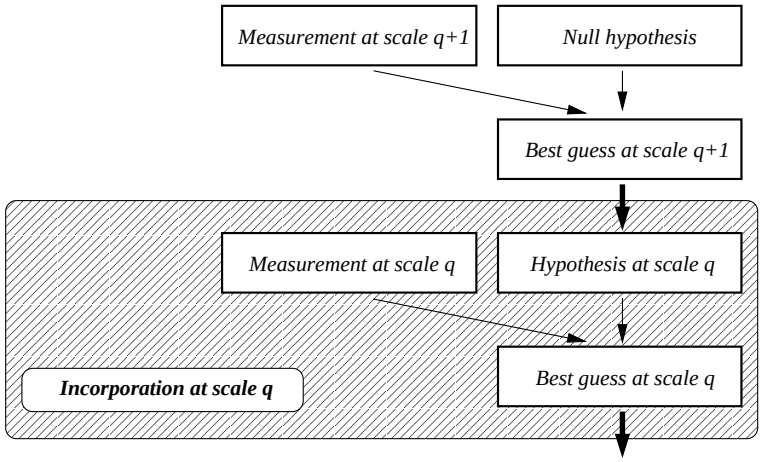


Figure 3.3: Bayesian incorporation of multiple scales: optimal PDF (probablity distribution function) at scale $q+1$ becomes a hypothesis at scale q. This, along with the likelihood (measurement) at scale q, can give the posterior (best guess) at scale q by Bayesian inference.

This figure explains how PDFs are propagated through scales.

Assume a null hypothesis (e.g. $p_q(v) = const$), measure the fittness at every position with every possible motion vector and associate it with the corresponding probability function (equ. 3.6). During this measure, a window function (W in equ. 3.5) was used, which corresponds to the scale (q). Multiply these PDFs then assume it to be the hypothesis at next step. In the next step, use a smaller window for the measurement. And repeat, until the predefined window size is reached.

In this induction-like rule, we will need a *prior* for the whole model. We choose a zero mean Gaussian for this which biases for slow speed. Preference of such has been found to play role in human perception [WA98]. In most cases it is not necessary to have a well parametrized null hypothesis (covariance), since it affects the results only marginally which is, in fact, quite a desirable characteristic of a model. Note that, if Gaussian was chosen as a *prior*, then the calculation of the expression $\log p(\bar{v}_0)$ becomes simply quadratic.

This model should optimally resolve the trade-off in sec. 3.6, where the term *optimal* is meant in Bayesian sense.

The recursive calculation scheme above might recall ref. [WC99], in which the author got similar conclusion to ours with respect to calculating the Bayesian optimal distribution. Noteworthly, he has found that subsequent experiments (or measurements), even with a simplistic model of the "world" (*prior*), can gradually *reinforce* or *refute* hypotheses and, finally, they are incorporated in an optimal estimation.

It has to be noted that numerous multi-scale (ref.[BB95, Ana89, BAK91, Enc86, Gla87, MP]) and also Bayesian OF models (e.g. ref. [Zel04], [LM95]), are known in the literature. Furthermore, in ref. [Sim99] a Bayesian multi-scale differential OF estimation by Simoncelli[2] was published.

[2]At surface level, it may seem that our model is similar to the one by Simoncelli, who also considered a probabilistic multi-scale setup. However, Simoncelli's assumption are rather different from ours. In particular, he worked with OFC-based models where he made assumption on the distribution of the noise corrupting the images' time derivative and the flow field, whilst we do not introduce flow-field noise. He introduced multi-scale framework in which the mean and the covariance of the estimation is propagated (updated and corrected) by a linear Kalman filter from higher scales to lower scales.

3.8 The Algorithm

3.8.1 The First Approach

First, let us overview the steps of the proposed algorithm.

- initiate the DFD with null hypothesis,
- measure DFD at a scale,
- accumulate the DFD in order to obtain the log of optimal PDF at the current scale,
- repeat the last two steps for all scales,
- finally, select a minima of the accumulated DFD.

Writing it in pseudo code see alg. 1.

Algorithm 1.

log PDF generation

```
Initial DFD is either even or zero mean Gaussian

Cycle for scale (q)
    Cycle for position (p̄)
        Cycle for displacement (v̄)
            Cycle for DFD (ξ̄)
              Accumulate for
              DFD(v̄(p̄)) = 1/|R^q_p̄| Σ_{ξ̄∈R^q_p̄} |I^(1)(v̄ + ξ̄) − I^(2)(ξ̄)| · W(ξ − p̄)
            End cycle DFD
        End cycle (v̄)
    End cycle (p̄)
End Cycle (q)
```

MAP selection

```
Cycle for position (p̄)
    Cycle for displacement (v̄)
        copy v̄_MAP(p̄) = arg max_v̄ p(v̄(p̄))
    End cycle (v̄)
End cycle (p̄)
```

Due to the extreme size of calculations ($O(N^7)$), for having results in reasonable time, optimization is desirable. We have found that by means of CNN-UM the reduction of the need of operations is somewhere in the magnitude of $O(N^3)$ (see appx. B.2).

3.8.2 The Concept of CNN-UM

In the previous chapter I detailed several versions of the Cellular Non-Linear/Neural Network along with the analysis of their dynamical behaviour. Here, indeed, I found it important to clarify the ways in which one can exploit the power of the Universal Machine (i.e. CNN-UM). Hence a short review on the structure of the architecture is given in order to let the reader capture the origin of the inherent power of the structure that we are to exploit here. For further explanation, see ref. [CY88b, RC93].

The governing equation of the space invariant CNN core can be written in a convolutional form as,

$$\begin{aligned} \dot{x} &= -x + A * y + B * u + z \\ y &= f(x) = \frac{1}{2}(|x+1| - |x-1|) \end{aligned} \tag{3.8}$$

where A and B are called templates and are realized in a matrix form of size 3x3 in current implementations, x, y, u, z are called *state, output, input* and *bias images*, respectively, and are realized in a matrix form of NxN. On the mixed signal VLSI implementation (ACE16K see ref. [ZDCE02]) we use for demonstration purposes, $N = 128$. The CNN-UM is constituted by the CNN core as a workhorse and other quasi peripheral components, which makes the CNN core programmable. It consists of local and global analog memories (LAM, GAM), local and global logic memories (LLM, GLM respectively), a local logic unit (LLU), etc. In the ACE16k implementation, there is also a binary masking possibility that prevents any cell taking part in the transition of state (x), if the associated binary value in the mask is set.

The advent of the analogue VLSI implementation of the CNN-UM is that, in principle, almost any stable output can be reached in a fixed time independently of the vector (image) size. This is, usually, in the magnitude of few μsec. One of the most outstanding characteristics of the ACE chip series is its very sensitive on-chip optical sensor that enables very high frame-rate image capturing (up to 30,000 frames per second [ZR04]) and parallel processing in place of the focal plane sensors.

The CNN core being a recurrent network, its behavior can be extremely complex but we are interested in region of parameters where a stable output can be reached. There is no need to mention that there is an extremely long list of authors who dealt with the stability of CNN and recurrent networks, in general, but we use only a simple-to-understand constellation of parameters.

In order to get deterministic result of the convolution defined in equ. 3.8 external cells (boundary cells) must be assumed with definite values. The global boundary condition (BC) defines its type, which can be fixed (or constant), periodic (or toroid), zero flux (or Neumann), gradient-flux.

3.8.3 The Exploitation of the CNN Concept

To make the handling of the CNN core easier, one can decompose it into two parts. The first part is the autonomous part ($\dot{x} = -x + A * y$), the second is a fix term part ($B * u + z$) that contributes to the evolution of the equation as a constant.

If, e.g., $A = 0$, called control mode, CNN simply copies the fix term into vector(image) x. Note the usefulness of such a simple operation. It achieves a convolution and vector addition of an image of N^2 pixels in a single step.

In the calculation of DFD (see right hand side (RHS) of equ. 3.3) one can notice a term which is an average of differences of a particular region ($R^q_{\bar{p}}$) around the point $\bar{p}$. This difference (i.e. $I^{(1)} - I^{(2)}$) can

be calculated in control mode by a $[B_{0,0} = -1]$ [3] template. Naturally, a translation operator was applied to generate the shift of images before the subsequent subtraction took place which can be achieved by a single B template operation. By putting an image in u, applying $[B_{0,1} = 1]$ results in a translated image copied into x.

The autonomous part of the CNN core can effectively calculate the average ($\frac{1}{|R|}\sum_{\bar{\xi}\in R}$) of a given region as quick as, for example, a pixel-wise difference calculation by means of an isotropic diffusion operator (i.e. A is central symmetric positive definite with $B = 0$) This equation asymptotically approaches the average (i.e. $x(t) \to [x(0)]_{average}$) if conservative BC is applied.

If we apply a masking window of $R^q_{\bar{p}}$ that enables CNN to operate around a pixel $\bar{p}$, then a diffusion operator is able to calculate the average in this window. This recognition lets us spare the `Cycle for DFD` and save $O(N^2)$ operations as shown below,

Algorithm 2.

log PDF generation

`Cycle for scale (`q`)`

  `Cycle for position (`$\bar{p}$`)`

    `Cycle for displacement (`$\bar{v}$`)`

      `Generate mask (`$R^q_{\bar{p}}$`) for the next CNN operations`

      `Template running:` $Diff(\bar{v}) = Translate(I^{(1)}, \bar{v}) - I^{(2)}$

      `Template running:` *average* `of` $Diff$ `within` $R^q_{\bar{p}}$ `by means of a mask.`

    `End cycle (`$\bar{v}$`)`

  `End cycle (`$\bar{p}$`)`

`End Cycle (`q`)`

Instead of calculating the simple DFD, the Gaussian windowed version (DFD_{Gauss}) can be obtained by dropping the masking window and applying a single diffusion with impulse response exactly the same as the Gaussian windowing function. (see appx. B.1). This step spares 2 cycles for $\bar{p}$ and adds a diffusion with few μsec expense. The gain on the complexity of the algorithms is estimated in appx. B.2. See the modified algorithm below,

Algorithm 3.

log PDF generation

`Cycle for scale (`q`)`

  `Cycle for displacement (`$\bar{v}$`)`

    `Template running:` $Diff(\bar{v}) = |Translate(I^{(1)}, \bar{v}) - I^{(2)}|$.

    `Template running: Diffusion of` $Diff(\bar{v})$ `for` τ_q

  `End cycle (`$\bar{v}$`)`

[3] In general, template elements (i.e. A_{ij} and B_{ij}), if not indicated otherwise, are assumed to be zero.

`End Cycle (`q`)`

One can visualize the steps to be done as in fig. 3.4.

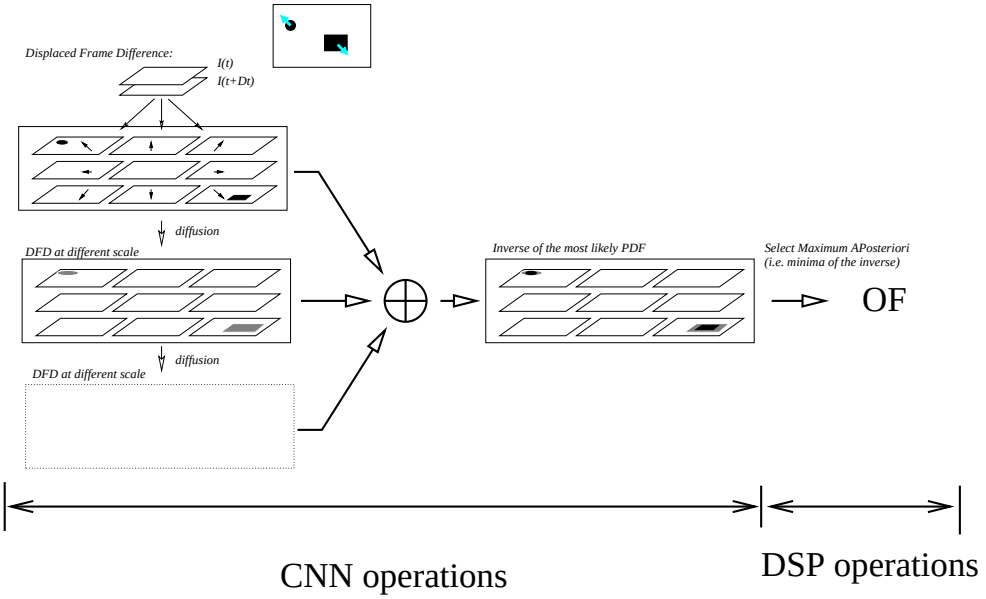


Figure 3.4: Calculation scheme of the proposed algorithm: First, DFD in every direction is calculated between the succeeding frames, then the corresponding log PDF is calculated. Second, diffusion is applied to DFD then its PDF is calculated. It is repeated until the length of the repeated diffusion operations produce the profile of the desired largest Gaussian window size. Finally, a pixel-wise addition of log PDFs is aggregated in a single PDF in which the selection of the peaks gives the MAP decision.

Selecting the minima of $((2v_{max}+1)$ x $(2v_{max}+1))$ number of images at every pixel, can be achieved optimally on CNN, in principle, but unfortunately, not feasable yet. This is since, on the one hand, the number of on-chip implemented LAMs is very small (specifically 8 LAMs on ACE16k), therefore, read-out and write-in operations from/to the chip gets unavoidable, so almost all the benefits we had by on-chip processing is lost. On the other hand, analog memories fade by time passing, therefore, even if one would have enough number of on-chip implemented LAMs, during repeated comparisons the noise increases so much so that the results of these comparisions became useless. Hence, it is preferable to do the maximum selection on a dedicated Neumann processor (DSP - digital signal processor) which, among other calculation tasks, controls the otherwise passive ACE16k and maintains connection to the digital "world".

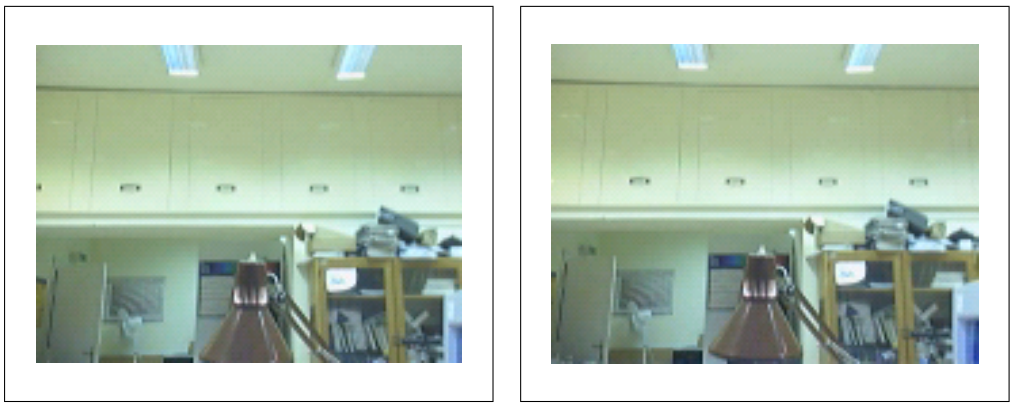
By analyzing the algorithm, one can realize that by increasing the scale, the average of a given region is calculated repeatedly. Due to the commutativity of the summation operator in equ. 3.7, the order of the calculations can be reversed, resulting in a significant contribution to the speed increase if performed by CNN. This is due to the fact that calculating a diffusion can be interrupted at any time and continued with no disturbance. After all, all we have to calculate is the sequence of the only temporal parameter τ_i for which the diffusion must be active in order to reach the next step. Thanks to this rearrangement, the estimation of OF starts from small scale and progresses towards larger scales, while the PDFs multiply in between.

In appx. B.3, we publish on-chip diffusion tests achieved for standard test functions such as the Heaviside step function and the rectangular test function. These tests make possible to estimate the

chip's diffusion time constant by experience and give some heuristic analysis of the deterministic and stochastic noise sources distorting the measurement of the DFD_{Gauss}.

3.9 Performance Analysis

In this section we detail the outcome of several experiments. The first and the second examples are made by using an artificially "panned" camera in order to have a good performance meter (as a constant motion vector at every pixel) and for demonstrating the effect of the diffusion on single scales. For the given purpose, we had set up a steady scene viewed by a low quality camera, then we picked two subsequent frames and cropped them. The first image is done at the top left and the second one at the bottom right (see fig. 3.5) by 3x3 pixels, each. This was to avoid random pattern (and shot noise) noise-correlation between the frames, which would be a significant help for any OF algorithm, using a *single* image, and cropped two times, similarly to ours. In this course, we were also faced with the pulse noise of the un-switchable Auto Gain Control (AGC) of the camera, which significantly decreased the quality of the correlation between frames, too. However, if one is to design an OF algorithm for real life applications, he/she probably needs to cope with external, already existing built-in mechanisms and global noise sources, like the AGC.



(a) Frame #1: Bottom-right band of 1 pixel width is sliced from the first image

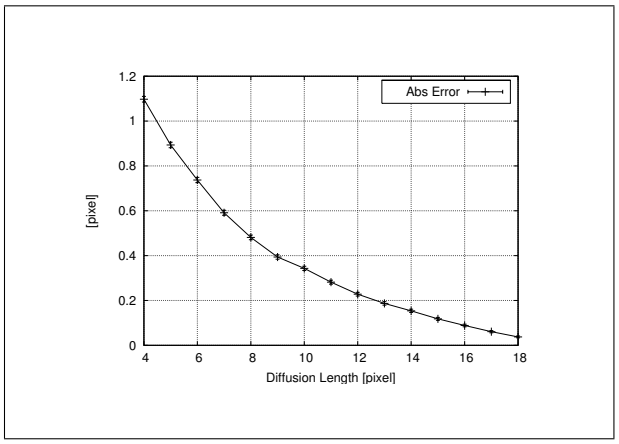
(b) Frame #2: Top-left band of 1 pixel width is sliced from the second image

Figure 3.5: In order to get a vector field that is known at every pixel, an artificial translation (panning) was introduced by slicing the frames of a video flow recorded from a steady scene.

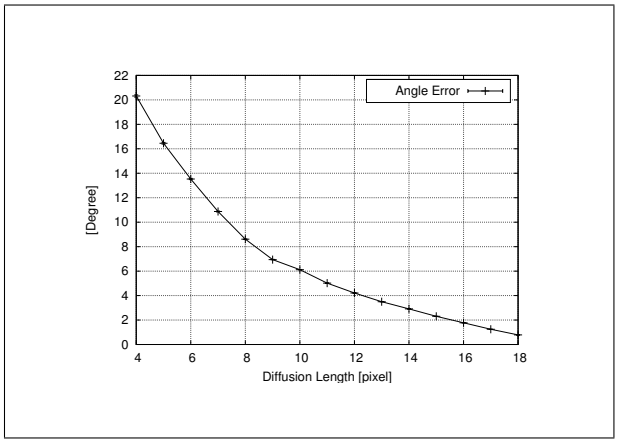
As long as the first experiment that was produced by simulation, the second one was carried out on-chip measurements, using the cropped images. The third and the fourth experiments' input was the standard *Yosemite image sequence* (ref. [Bla]), done by simulation and on-chip, respectively. Finally, the last example is another multi-scale example that was made on-chip, grabbed by the *optical sensor* of the chip at 10,000 fps.

3.9.1 Single Scale Analysis

In this series of experiments, we examined the effect of changing the diffusion length on the artificially panned images. The model performs as one can expect. The measure displayed measure both the motion vector's absolute error and its degree error. Due to the artificial setup of the experiment, the error decreases gradually, as scale increases (see fig. 3.6).



(a) The relative error of the motion vectors' absolute value is depicted as the function of scale.



(b) The relative error of the motion vectors' degree is depicted as the function of scale.

Figure 3.6: The relative error measured on images of fig. 3.5 is demonstrated. Note the increase of the reliability by the increase of the scale.

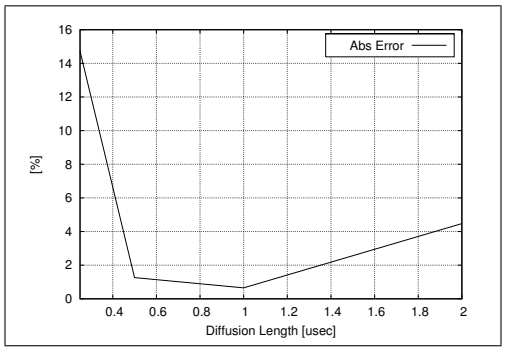
Due to the similarity of fig. 3.6 and fig. 3.7, similar conclusions apply but, apart from the low sampling rate of the later one, one might notice the increase of the error over 1 μsec of diffusion time. Without any in-depth analysis, we think that this is due to the strengthening offset drift of the chip, which is more-or-less proportional to time.

Based on these graphs, two distinct and controversial claims can be stated:

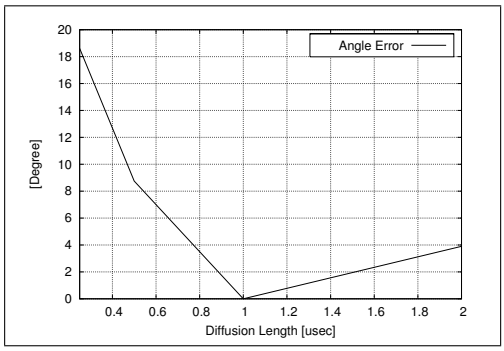
- By decreasing the scale, the reliability of the measurements increases,
- As the diffusional length decreases, the spatial bandwidth of the motion vector information increases.

This is another form of the accuracy/spatial bandwidth dilemma. Hence, the combination of multiple scales seems to be a ultimate trade-off.

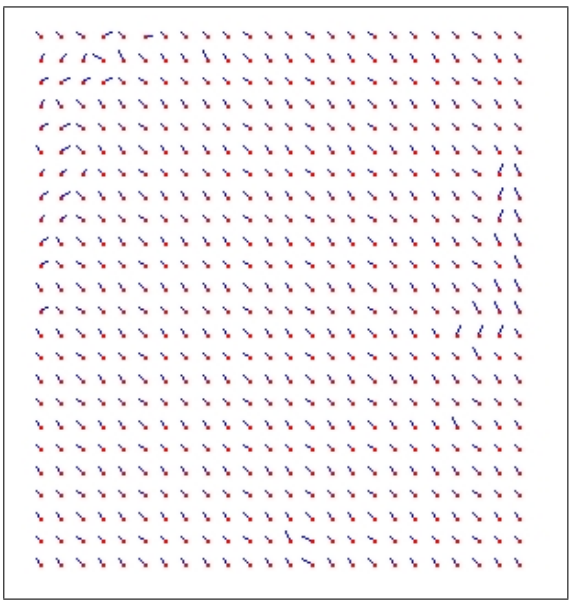
However, in the proposed framework we unified the advantages of multiple scales and tried to bypass the disadvantages, there are leftover questions . These mainly regard the selection of the cuts of the scale-space. Currently, we did cuts in an evenly distributed fashion in the log domain with an arbitrary chosen unit.



(a)



(b)



(c) The vector field by using 4 scales (diffusion to 0, 1, 0.5 and 2 μs on-chip.)

Figure 3.7: The sub-figures (a) and (b) are, essentially, the same graphs as in fig. 3.6 but made on ACE16k chip, using the standard test image Lenna. The resulted vector field is demonstrated in sub-figure (c). N.B. The vector field is generated by $128x128$ resolution but it is sampled for displaying only.

3.9.2 Multi-Scale Analysis

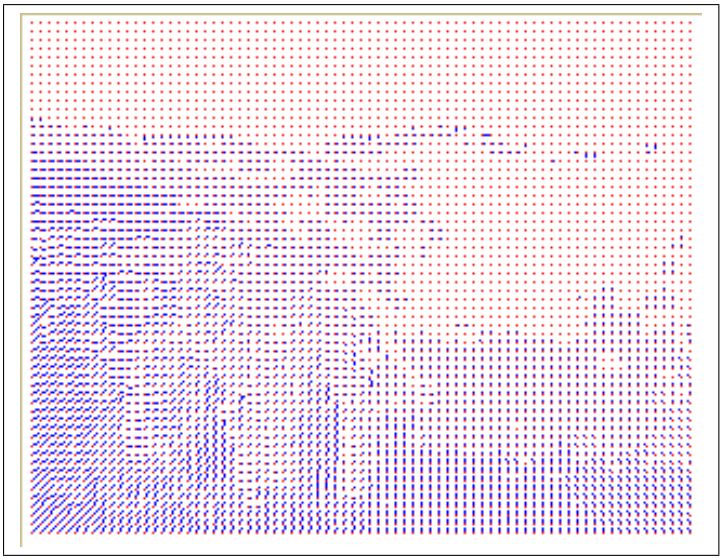
First simulations were made on the Yosemite standard test sequence[4] (fig. 3.8) for demonstrating the quality of the algorithm, then that of a crop of size of $128x128$ at coordinates of $100x50$. Finally the same was tested on-chip (fig. 3.9).

Tests was made with the on-chip sensor at extra high frame rate (fig. 3.10),too. The pill on the image is moving in south-south east direcetion.

In order to choose the right scales to use in the model, we made few simulations with focusing on the combinations of the scales (fig. 3.11), and we have found that by gradually increasing the scale, the effect of the multi-scale information reinforcement is getting more-and-more pronounced, while a low level of information degradation is perceivable, too.

[4]Yosemite sequence is a well known synthetic image sequence for measuring performance of optical flow algorithms. The motion of the camera is like flying above the valley in a way that upper-right corner of the valley remains fixed, while the bottom-left corner moves about 4 pixels per frames. For this reason we used for motion vectors within $\{-4,4\}x\{-4,4\}$

(a) The test image and vector field



(b) The vector field only

Figure 3.8: Evaluation on the Yosemite standard test sequence by simulation.

3.9.3 The qualitative performance

Table 3.2 summarizes the qualitative performance of our method measured on the Yosemite sequence.

Lines marked with $*$ in superscript indicate measurements made on the full size Yosemite image (316x252), while the ones not marked were made in chip size (128x128) only. This a cropped version of the Yossemite at coordinate 100,50.

The difference in the number of pixels, reasons why eventually the average angular error (AAE) of *chip vs. ground truth* is better than *sim vs. ground truth*.

These results were obtained without any fine tuning of the used scales (i.e. diffusion lengths τ_q's and λ_q's in equ B.5). It should also be noted that the model generates a 100% dense vector field. Due to the

range in this tests.

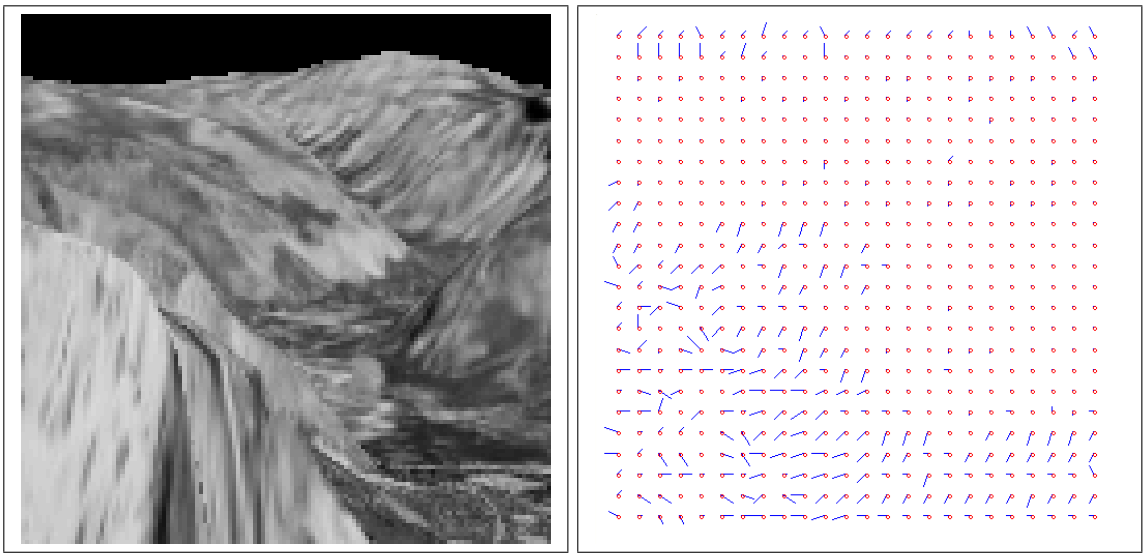


Figure 3.9: The vector field of the Yosemite sequence using 4 scales (diffusion applied for 0, .5, 1 and 2 μs on-chip). Strong noise diffuses from the left bottom corner of the chip (see last figure in appx. B.3).

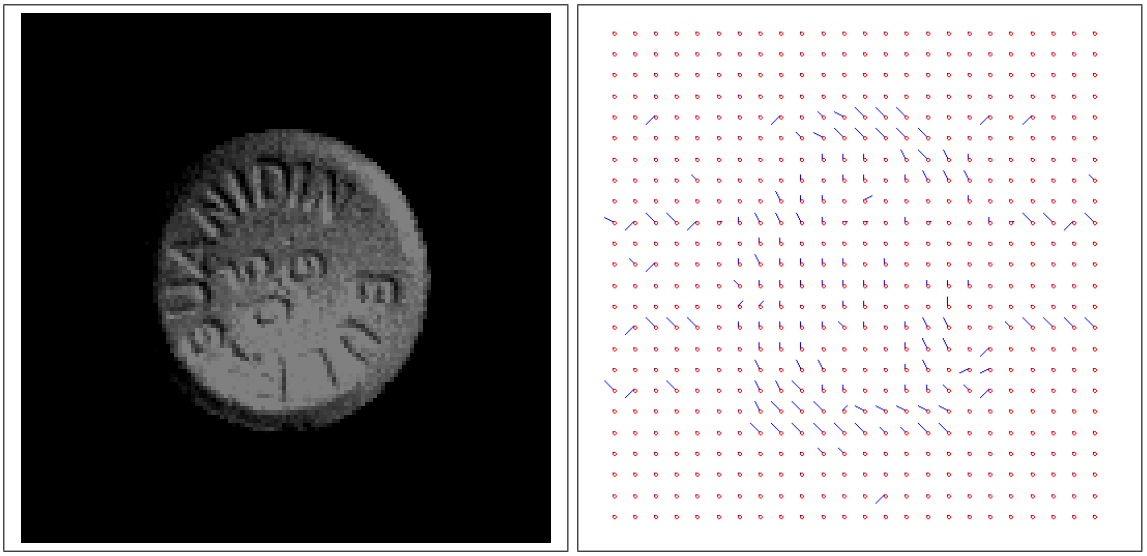


Figure 3.10: The vector field of a pill moving upward grabbed at speed of 10,000 frames per second (diffusions applied for 0, .5, 1 and 2 μs on-chip).

quantization of $\bar{v}$, there is a theoretical limit on the angular error in the model ranging from 6.3402° to 22.5°. The model seems to perform in the tested range with a considerable variance (STDD). In order to make comparisions fit to theoretical limitations of our model, the components of $\bar{v}$ were rounded. In Table 3.2 these lines begin with "(int)", while those measurements that were made with the original performance meter begin with "(float)". This also explains why (int) measurements are considerably better.

In Table 3.3, a far not complete list of methods are given capable of generating 100% dense vector field.

Note that, no qualitative performance results were published on the very standard Yosemite sequence

Table 3.2: Error performance on the "Yosemite without clouds" sequence (100% dense) using only 2 frames. AAE stands for Average Angular Error (measured in degrees) and AVE stands for Average Vector Norm Error (measured in pixels)

Basis for comparison / errors	AVE	AAE.	STDD of AVE	STDD of AAE.
(float) sim vs. ground truth*	0.5	9.6	0.48	11
(float) chip vs. ground truth	0.81	25	0.54	15
(float) chip vs. simulation	0.95	9.9	0.73	20
(int) sim vs. ground truth*	0.35	4.9	0.57	10
(int) chip vs. ground truth	0.41	3.8	0.55	14
(int) chip vs. simulation	0.43	1.3	0.63	10

in case of the few existing aVLSI OF solutions. Solutions with resoltion of 15x15 at 500 fps [SD04], or with resolution of 95x52 at 30 fps [MEC04] publish performance measures on proof-of-concept kind of tests (few geometric objects traslating in predefined directions).

In a recent paper [MZC+05] an FPGA based digital OF solution is given with resolution 256x256 at 60 fps implementing the original Horn & Schunk.

Table 3.3: Comparison of models (100% dense) on "Yosemite without clouds" sequence. AAE stands for Average Angular Error.

Model	AAE
Uras [UGVT88]	10.44
Horn & Schunk (HS,original)	6.79
Lucas & Kanade (LK)	5.71
Black & Anandan	4.56
CLG (combined HS&LK)[BWS05]	4.32
Simoncelli [Sim99]	3.55
Brox *et al* [BBPW04]	0.89

It should also be noted that there are numerous high quality methods and as far as we are concerned, the method of Brox *et al* [BBPW04] and Aubert *et al* [ADK99] are far the best, but their computational requirements are tremendous, which postpones the possibility of using them in on-line, realtime applications in the visible future.

3.10 Conclusion

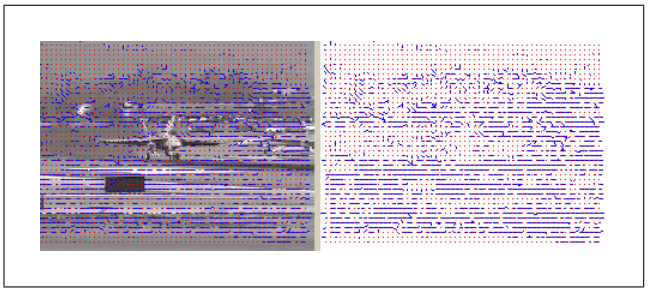
Based on some of the known phenomena attributed to VSS (Bayesian inference, multi-scale view), we have formulated a simple, but usable multi-scale OF algorithm that was shown to be applicable in a mixed signal VLSI technique. As other experiments showed, simple ad-hoc approaches (such as edge detection with center-surround filters) can easily improve the performance of the developed algorithm but here we have concentrated on the formalism itself which proves to be powerful both in computational complexity and in accuracy on a general purpose aVLSI hardware versus special purpose hardware with the largest resolution. The resolution of the chip we used was enough to measure reasonable performance meter on standard image sequence (Yosemite) over 100 fps. Formerly, the conducted simulations pointed out a theoretical limit on the accuracy of the algorithm, which the analog on-chip results seem to get close to, robustly. This is achieved as fast as none of the existing solutions exceeding our accuracy, even in digital hardware at this resolution.

Although the main disadvantage of the algorithm is the inevitably large memory needs, which increases quadratically with the size of the motion vector but, as we have shown, it can be kept limited.

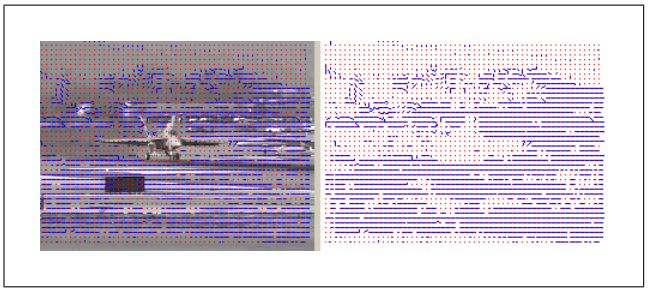
Further investigation should be made to determine the optimal distribution of scales on given images (scale space analysis) what is currently made as a regular sampling only. It is also desirable to determine if the aggregated PDF provides more information than a single vector field. We assume that this increased information may be exploited easily in standard image processing tasks such as segmentation.

(a) Frame #1 (b) Frame #2

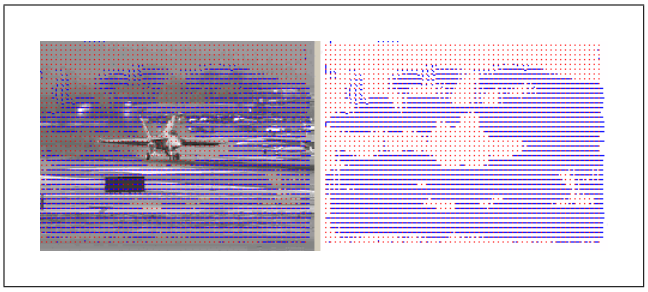
Figure 3.11: An example: the camera is panned to follow the airplane; i.e. the background is moving while the target is appearing steady.



(a) Diffusion of 4 pixels is applied



(b) Diffusion from 4 to 8 pixels is applied



(c) Diffusion from 4 to 16 pixels is applied

Figure 3.12: The gradual reinforcement of the motion vector information by scale change is depicted. Note the "smoothing" of the motion vectors in the background of the airplane, even in regions where no measurable image gradient (i.e. local information) is present, whereas the area of the zero motion vectors get the shape of the airplane.

Chapter 4

Smooth Stochastic Resonance

4.1 Brief

A new class of stochastic resonator (SRT) and Stochastic Resonance (SR) phenomena are described. The new SRT consist of a classical SRT, one or more time derivative circuits and the same number of time integrators. The incoming signal with additive noise is first time derivated, then passes through the classical SRT and finally it is time integrated. The resulting SR phenomena show a well defined SR. Moreover the signal transfer and SNR are the best at the high frequency end. A particular property of the new system is the much smoother output signal due to the time integration. Interestingly enough, the phenomena can be observed in terms of CNN.

4.2 Notation and Symbols

$1(t)$	step function
A	signal amplitude in IDSRT
B	amplitude of LCD's output
δ	dirac delta
$diff(t)$	signal (and noise) after derivation in the IDSRT
DFT	Discrete Fourier Transform
f	frequency in IDSRT
FIR	Finite Impulse Response filter
f_c	cut-off frequency
f_s	frequency of the signal
FT	Fourier Transform
FD	Fourier Domain
$in(t)$	input of the IDSRT (signal + noise)
$int(t)$	output of the integrator in IDSRT
H.O.T.	Higher Order Terms (of the Taylor expansion for example)
IDSR	Integro Differential Stochastic Resonance (phenomena)
IDSRT	Integro Differential Stochastic Resonator (device)
LCD	Level Crossing Detector

$lcd(t)$	output of the LCD in the IDSRT
μ	SNR gain
$n(t)$	noise in IDSRT
ν_0	average mean firing rate
P	Power of signal
PSD	Power Spectral Density
RMS	refers to the Root Mean Square (power of the noise)
$s(t)$	signal in IDSRT
$\hat{s}$	FT of s
$S(f)$	power Spectra
σ	rms of noise
SNR	Signal to Noise Ratio (measure)
SR	Stochastic Resonance (phenomena)
SRT	Stochastic Resonator (device)
τ_c	correlation time
TD	Time Domain
U_t	Threshold level
ω	rotational frequency

4.3 Introduction

Stochastic Resonators (SRTs) (see Refs. [BAS81, WM95, BV97]) are one of the most intensively studied topic concerning noise assisted signal transfer. The particularity of these systems is that the signal power and signal-to-noise-ratio (SNR) at the output expresses well defined maxima versus the input noise strength.

Classical SRTs are threshold based devices. Therefore their output amplitude executes large abrupt jumps. This behavior is often disadvantageous in practical signal processing systems.

The motivation of the present work is to introduce a new class of SRTs and stochastic resonance (SR) phenomena which can produce arbitrary smooth amplitude while their SNR is as good as that of classical systems. Another important property of the new system that, under practical conditions, it has a frequency-resonance.

In the next sections, first the general integro differential stochastic resonator (IDSR) system is introduced in Sec. 4.4; then one appropriate realization we used is introduced in Sec. 4.5; the theoretical results are given in Sec. 4.6; and the simulation results are presented in Sec. 4.7.

Finally, according to the fact, that we rely on the findings of Ref. [Kis96] and it has not been published in journal, we survey the main results of this article with respect to the signal in the linear response limit.

4.4 Integro-Differential Stochastic Resonators

The integro-differential stochastic resonator (IDSRT) is a classical stochastic resonator with a time-derivation unit added before the output and a corresponding time-integration unit added at the output (see Fig. 4.1). Most of known stochastic resonators produce “rough” output containing “discontinutities”, such as sharp spikes, random telegraph signals, abrupt switchings between amplitude levels. In an IDSRT,

the expected "roughness" is significantly less because of the time-integration. Higher-order IDSRTs, with multiple derivators at the input and correspondingly multiple integrators at the output, will obviously produce an even smoother output. It is important to note that the smoothing effect works only on discontinuities added by the original SRT but not against discontinuities which may be found in the input signal.

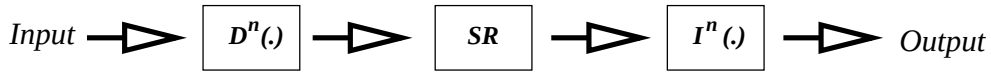


Figure 4.1: Integro-Differential Stochastic Resonator system architecture.

4.5 First order Integro-Differential SRT with a Level-Crossing-Detector

Having a well understood stochastic resonator, the so-called Level Crossing Detector (LCD), we applied it for the realization of the given model.

In general, an LCD emits spikes at its output whenever its input crosses a predefined threshold level. These spikes are now integrated in the new model, so instead of spikes stair like function appears at the output due to the integration. (see fig. 4.2)

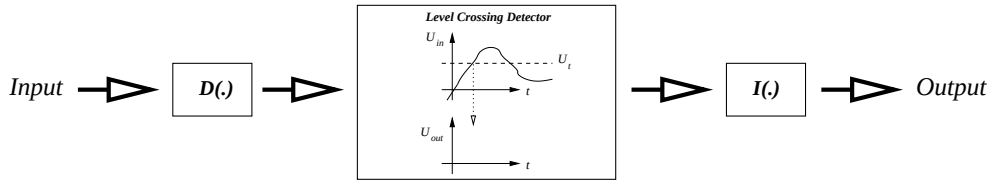


Figure 4.2: Integro-Differential LCD architecture.

With higher order derivatives and higher order integrations we can get parabolic approximations which must be a "smooth" approach to the signal, particularly if it is sine wave function.

One might think that applying a simple linear at the output of an SRT would produce a sufficiently smooth output. True, however the unpleasant phase shift would appear which does not happen if IDSRT is used.

On the other hand, we need to consider the fact that not only the signal is integrated in the post processing filter but the errors, too. In order to reduce its slowly varying component, not only the ideal integrator (i.e. numerical sum called stepwise integrator) but the leaky integrator was tested, as well.

In the next section we show how these particular linear input/output transformations affect the SR models in general.

4.6 Theoretical Results

In the processing line the derivator ($D^n()$) of order of (n) processes its input (denoted by in) providing the input for the LCD (denoted by $diff$). The LCD's output (denoted by lcd) is integrated ($I^n()$) which is the output (denoted by $int = out$) the entire process. (see fig. 4.3).

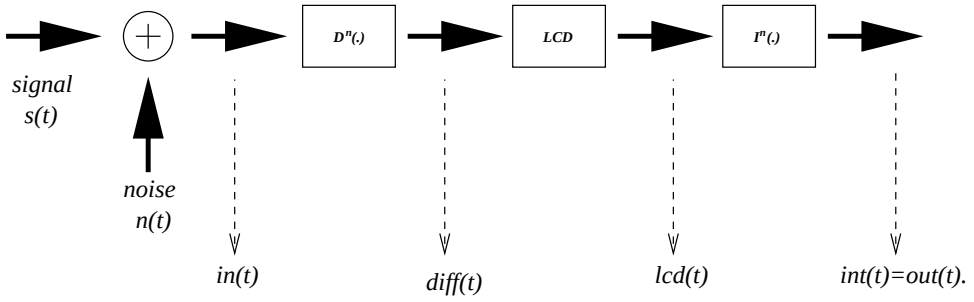


Figure 4.3: Nomenclature of the measurement points of the system we will refer to. The symbol '+' in the circle is an adder.

The SNR is calculated for sinusoidal input singals as

$$SNR(f_s) = \frac{P}{S(f_s)},$$

where P is the signal contribution to the power spectral density (PSD) to the noise power spectra ($S(f)$) at f_s.

The main purpose of the theoretical analysis is to specify $\frac{SNR_{out}}{SNR_{in}}(f)$ analytically and derive its absolute maxima in explicit form.

4.6.1 Input

The input of the system (denoted by $in(t)$) consists of a hard limited white ($S(f) = S$) *noise* in the frequency domain (denoted by $n(t)$) and a signal (denoted by $s(t)$);

$$in(t) = s(t) + n(t).$$

The cut-off frequency of band limited white noise is f_c (i.e. the noise correlation time $f_c \approx 1/\tau_c$). The rms of noise is denoted by σ_{in}.

The *signal* is a sinusoidal wave with amplitude A and frequency $f_s = \omega_s/2\pi$.

The sinus wave signal's Fourier Transform (FT) can be written as $\hat{s}(f) = A\delta(f_s - f)$, thus the power of such a signal is

$$P_{in} = \frac{1}{2}\int_0^\infty (\hat{s}(f))^2 df = \frac{A^2}{2}. \tag{4.1}$$

Since the white noise has uniform PSD, $S_{in} = \frac{\sigma_{in}^2}{f_c}$ holds. In this way,

$$SNR_{in} = \frac{A^2 f_c}{2\sigma_{in}^2}. \tag{4.2}$$

4.6.2 Derivation

A derivation applied to $in(t)$ does not change the $SNR(f)$ albeit changes the PSD of $P_{diff}(f_s)$ and $S_{diff}(f_s)$ compared to P_{in} and S_{in}, respectively.

In general, a derivation in time domain (TD) is a multiplication by jω in the Fourier domain (FD)

$$\hat{s}_{diff}(f) = \mathrm{j}2\pi f A\delta(f_s - f) = \mathrm{j}\omega A\delta(f_s - f).$$

For this reason, the power of the signal (or noise) becomes dependent on its frequency as

$$P_{diff}(f_s) = \frac{1}{2}\int_0^\infty (\hat{s}(f))^2 df = \frac{1}{2}\int_0^\infty (\omega A\delta(f-f_s))^2 df = \omega_s^2 \frac{A^2}{2} \tag{4.3}$$

which applies to the PSD of the noise too as

$$\begin{aligned} P_{diff}(f_s) &= \omega_s^2\ P_{in}(f_s) \\ S_{diff}(f_s) &= \omega_s^2\ S_{in}(f_s). \end{aligned} \tag{4.4}$$

For this reason, in their ratio ω_s^2 always simplifies as

$$SNR_{diff}(f_s) = \frac{P_{diff}(f_s)}{S_{diff}(f_s)} = SNR_{in}(f_s). \tag{4.5}$$

While, the SNR is intact under this linear transformation, the rms of the noise changes as,

$$\sigma_{diff}^2 = \int_0^{f_c} S_{diff}(f)df = \int_0^{f_c} (2\pi f)^2 S_{in} df = \frac{4\pi^2}{3} f_c^3 S_{in} = \frac{4\pi^2}{3} f_c^2 \sigma_{in}^2.$$

Note that, the noise of $diff(t)$ is not white.

4.6.3 Level Crossing Detector

Definition 1. An *asymmetric* level crossing detector (LCD) emits an impulse of amplitude (B) for duration of τ_0 (uptime) on its output ($y(t) = lcd(t)$), whenever its input ($x(t) = diff(t)$) crosses the threshold level (U_t) in increasing direction at time (t_i), i.e.

$$y(t) = LCD_{asym}(x(t)) \ = \ B\sum_i 1(t-t_i) - 1(t-(t_i+\tau_0))$$

where

$$x|_{t_i^-} \leq U_t \quad \texttt{and} \quad x|_{t_i^+} \geq U_t.$$

The output of the LCD is a composed of impulses of size B for duration τ_0, whenever x (input) is less then U_t before t_i but greater then U_t after t_i.

Definition 2. A *symmetric* level crossing detector (LCD) emits a positive impulse or a negative impulse for duration of τ_0 on its output ($y(t) = lcd(t)$), whenever its input ($x(t) = diff(t)$) crosses the positive threshold level (U_t) in increasing direction or crosses the negative threshold level ($-U_t$) in decreasing direction at time (t_i), respectively.

$$y(t) = LCD_{sym}(x(t)) = LCD_{asym}(x(t)) - LCD_{asym}(-x(t)).$$

The SNR gain of an LCD is detailed in Ref. [Kis96] and a brief is given in appx. C.1, regarding the case when SNR is within the linear response limit. From there we recall the results we rely in the calculation of SNR gain of IDSRT.

Unfortunately, in case of the IDSR, the LCD is fed by non-white noise, so the result cannot be applied from there as a one-to-one replacement.

The average mean firing rate (ν_0) of a one-directional zero-crossings can be written similar to Eq. C.1

as

$$\nu_0 = \frac{2}{\sigma_{diff}} \left(\int_0^{f_c} f^2 S_{diff}(f) df \right)^{\frac{1}{2}}. \tag{4.6}$$

Because of the variation in the spectral characteristics ($S_{diff}(f) = (2\pi f)^2 S_{in}$), the integral in Eq. 4.6 can be written as

$$\int_0^{f_c} f^2 (2\pi f)^2 S_{in} df = 4\pi^2 \frac{f_c^5}{5} S_{in} = 4\pi^2 \frac{f_c^4}{5} \sigma_{in}^2,$$

so Eq. 4.6 simplifies to

$$\nu_0 = \frac{2}{\frac{2\pi}{\sqrt{3}} f_c \sigma_{in}} \, 2\pi \frac{f_c^2}{\sqrt{5}} \sigma_{in} = \frac{2\sqrt{3}}{\sqrt{5}} f_c.$$

The SNR_{lcd} is similar to Eq. C.6 up to a substitution of the signal's amplitude in the IDSR setup which is $\omega_s A$, so it gets

$$SNR_{lcd} = \frac{U_t^2}{\sigma_{in}^4} \nu(U_t) \omega_s^2 A^2.$$

Since Eq. 4.5 holds, by means of Eq. 4.2 the SNR gain (μ) is,

$$\mu = \frac{SNR_{lcd}}{SNR_{diff}} = \frac{\frac{U_t^2}{\sigma_{in}^4} \mathbf{exp}\left\{\frac{-U_t^2}{2\sigma_{in}^2}\right\} \frac{2\sqrt{3}}{\sqrt{5}} f_c A^2 \omega_s^2}{\frac{A^2 f_c}{2\sigma_{in}^2}} = \frac{4\sqrt{3}}{\sqrt{5}} \frac{U_t^2}{\sigma_{in}^2} \mathbf{exp}\left\{\frac{-U_t^2}{2\sigma_{in}^2}\right\} \omega_s^2.$$

Assuming $\alpha = \frac{U_t^2}{2\sigma_{in}^2}$ it simplifies to

$$\mu = \frac{8\sqrt{3}}{\sqrt{5}} \alpha \mathbf{exp}\left\{-\alpha\right\} \omega_s^2. \tag{4.7}$$

The extrema of Eq. 4.7 is at $d\mu/d\alpha = 0$ which results in $|\alpha_{ext}| = 1$.

Using the considerations given in appx. C.2 regarding the symmetric LCD case opposed to the asymmetric, the maxima can be written as

$$\mu = \frac{8z\sqrt{3}}{\sqrt{5}\mathbf{e}} \omega_s^2 \tag{4.8}$$

where z is 1 if asymmetric, or 2 if symmetric LCD is examined. One can conclude that the maxima of the system became signal frequency dependent.

4.6.4 Integration

Considerations similar to Sec. 4.6.2 apply to the integration. The only difference is in Eq. 4.3, where instead of multiplying by ω_s, division must take place. Just as in Eq. 4.5, ω_s it will cancel. This means that by integrating the output of the LCD ($lcd(t)$), no SNR variation will appear;

$$SNR_{out} = SNR_{lcd}$$

so theoretically the Eq. 4.8 characterizes the overall performance.

It seems to be very attractive from the point of SNR gain since it provisions $SNR_{gain} > 1$. According to the experiments, we have found that the IDSR hardly reaches the performance of the LCD because the signal amplitude (at the input of the LCD) goes out of the linear response limit as the frequency increases, however in the low frequencies regime the theory fits well to the experiments.

Note that, after the input derivation, the sinusoidal signal amplitude will be proportional to the frequency. So, if we increase the signal frequency, the signal gets greater at the LCD. However the noise is constant. As it is well-known, the non-linear response limit is reached when the signal amplitude at the input of the SRT is greater than the RMS noise. Exactly that happens in this case. This is another important point which shows that not only the input noise is transformed but also the input signal by this operation. We have a new system here which has a new behavior which is radically different from the classical SRTs.

4.7 Simulations

4.7.1 Input - Gaussian White Noise Generation

Large number of uniformly distributed random variables added together according to central limit theorem, provides accurate Gaussian process.

The rms of the noise was measured in every experiment rather than scaled after measuring in single experiment as is common.

Since the analog LCD simulation in Ref. [Kis96], the fill factor of the impulse at the LCD's output was less than 10% in terms of the noise correlation time constant, first we built the experiment similar to that.

In order to reproduce the circumstances of the analog experiment, we needed a noise source at least 10 times slower than the impulse we can generate in the system. Equivalently, the noise had to be filtered by sharp low pass filter of 10th the cut-off frequency of the frequency corresponding to spike's uptime. While in TD, stepwise- (we refer to it as oversampling) and linear-interpolation was tested, in FD, linear Finite Impulse Response filters were tested. By means of these techniques, we have found that the measurements are non sensitive for this particular fill factor a up to 100% fill factor. However we need to point out that aliasing type of errors might appear in simulations that has to be circumvented in all cases. It can be done by buffers of size of least common multiples and/or filtering in FD. We used the fastest case when the noise correlation time constant is equal to the uptime of the LCD, which means that every item in the buffer corresponds to a random sample and the LCD emits an impulse for duration of one sample.

4.7.2 SNR measurements

SNR measurement is based on separate signal and noise estimation.

Because of the structure of the experiment is such that we are in possession of the signal's frequency at the input (hence at the output), we can obtain the power of this particular frequency at the output by a simple "read out" from the measured PSD.

The background noise estimation is not that obvious. The particularly interesting noise is in the vicinity of the signal in the PSD. We measured it by simply averaging the PSD around the signal of a given narrow region. Specifically, we used fixed number of $+/-5$ frequency samples excluding $+/-1$ samples around the signal. (NB: since DFT was used frequency resolution is predefined)

According to the findings of the Sec. 4.6, SNRs were checked on $in, diff, lcd, out$ points of the system. The graphs generated by in this way are the base of our experimental analysis. (see fig. 4.8 and fig. 4.9).

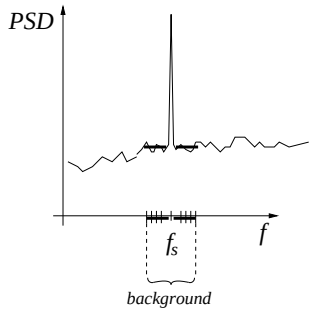


Figure 4.4: The SNR measurement is based on the PSD. Signal's power can be determined by a simple "read out" at given frequency. The background noise is computed by averaging numerous samples of the PSD in the vicinity of the signal's frequency.

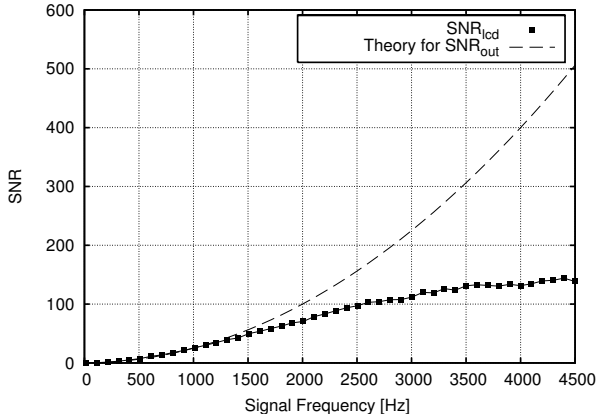


Figure 4.5: SNR vs. signal frequency at different points in the system at a given rms. Theoretical SNR_{out} curve is fitted to the numerical results in the numerically stable region. Theory explains signal transfer in the linear response limit which holds for up to $1500Hz$.

4.7.3 Derivation

According to the numerical instability of the numerical derivation, the numerical curve considerably departs from the theoretical one in fig. 4.7(b). Hence the Eq. 4.5 must hold up to the point of breakdown which implies that our measurement is supposed to be accurate no higher frequency than this. Thus Eq. 4.7 has at least the same high frequency limit. This can be justified in fig. 4.5 where the SNR_{in} and SNR_{diff} versus signal frequency fits well in this region.

4.7.4 Level Crossing Detector

Both asymmetrical and symmetrical LCD were tested with output impulse length equal to the noise sampling time. It is worth noticing that however the input noise of the LCD is colored, the output of the LCD is transformed to white noise in a considerable wide bandwidth which might have large potentialities in signal processing applications.

4.7.5 Integration

Achieving numerical integration in the TD leads to error prone results in the FD. Our experience supports this unreliability of the power spectra already show on fig. 3.20 in Ref. [Ran87]. We have found that this is due to the low representation degree of even conjugate signals with frequency lower than $1/T$ in the FD obtained by FT (including DC component). Unfortunately the stepwise integration biases the spectra into this frequency range. In general, the integration formula widely used in FD, in case of non analytical functions performs very badly with Discrete FT (DFT). There at least two ways to avoid this numerical problem. Either windowing (e.g. Gaussian) or analog (or leaky) type of integration (with time constant $\tau \ll T/2$) of the data in the DFT's buffer. We turned to leaky integration and tested the effect varying the integration time constant at $\tau = 50/f_c$ and $\tau = 10/f_c$.

4.7.6 Experimental results

In this section we give all the graphs of the measured systems. First, time series are shown, then PSDs are given, SNRs, SNR gain (SNR_{out}/SNR_{in}), signal power versus signal frequency and versus rms.

A short time course of an experiment is demonstrated in fig. 4.6.

Note the apparent positive tendency in fig. 4.6(d) in the range of $0...0.05s$ induced by the signal of $f_s = 10Hz$ which reflects the motivation of the IDSR setup, that is a smoother estimation of the signal at input by the output.

In fig. 4.7, the PSD graphs of the same experiment is given in fig. 4.6.

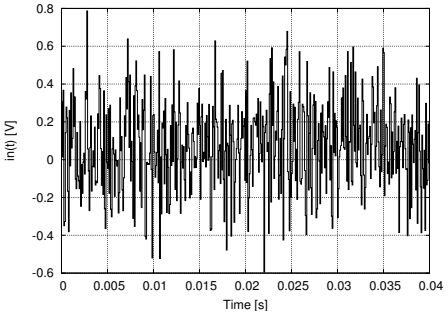
In the experiments, we used the following parametrization: Noise cut-off: $f_c = 12000Hz$, signal amplitude: $A = 0.1$, LCD threshold $U_t = 0.45$, LCD impulse amplitude: $B = 5$, LCD uptime: $\tau_0 = 0.833$. The scanned signal frequency range is $\{14Hz, ..., 4500Hz\}$ The rms was varied within the range of $\{0.01388...0.5625\}$ in logarithmic steps (i.e. multiplied by 1.0559 in each step). Rms values were obtained by measurements rather than scaling the noise according to a single measurement.

In fig. 4.8(a), SNRs are measured at different points in the system at a single rms using asymmetrical LCD with stepwise integrator. According to the expectations the SNR_{in} and SNR_{diff} is equal for the whole regime tested. Note that the equivalence holds even over the breakdown point (4500Hz) of the derivator. The output of the LCD fits well to the theoretical quadratic curve (just as in fig. 4.5) but the output of the integrator departs qualitatively from this. The reason is already detailed in Sec. 4.7.5. As mentioned, the solution is twofold. Applying leaky integrator (see sub-figures c, d, e, f in fig. 4.8) (which reduces the low frequencies in the spectra) and symmetrical LCD (which reduces the DC component) (see sub-figures b, d, f in fig. 4.8).

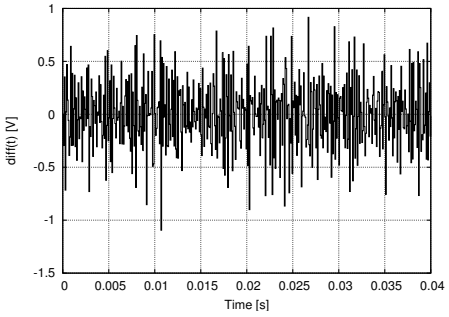
Interestingly in the experimental setup we used in generating all figures, the SNR_{out} reaches its maxima at around $rms = 0.04$ at any signal frequency which would require further analysis. (see fig. 4.9(a)).

It is demonstrated in fig. 4.9(b) the maxima of the signal power and the maxima PSD at the same frequency is not at the same point as function of rms which is the proof for the noise assistance in the signal transfer.

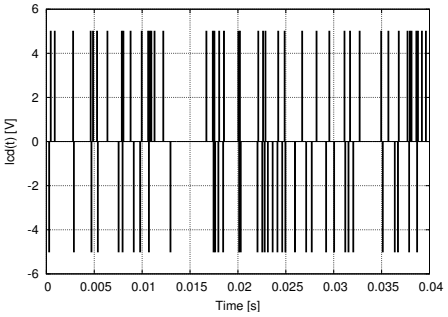
Depending on the rms, the SNR gain increases rapidly in all experiments and reaches its maxima before 0.2. By further increasing the rms the SNR gain remain constant (see in fig. 4.9(c)) which means that the SNR_{out} becomes proportional to the SNR_{in} at almost around the maxima. This absolute maxima of all experiments goes just a little bit above (0.868) the well known LCD's limit (0.849) although it happens in the region where the derivation is declared to be unreliable. (see fig. 4.9 and the Appendix)



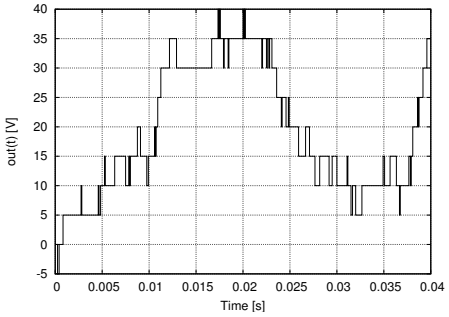
(a) The signal and the noise at the input.



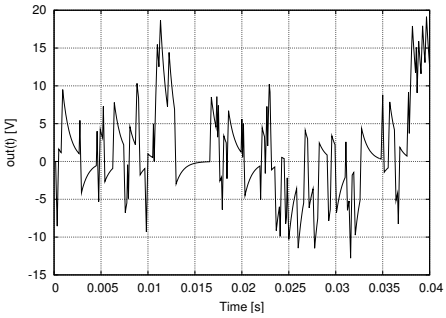
(b) The signal and the noise after derivation.



(c) The signal and the noise after the Stochastic Resonator realized as an LCD (*lcd*).



(d) Stepwise integration at the output of the LCD.



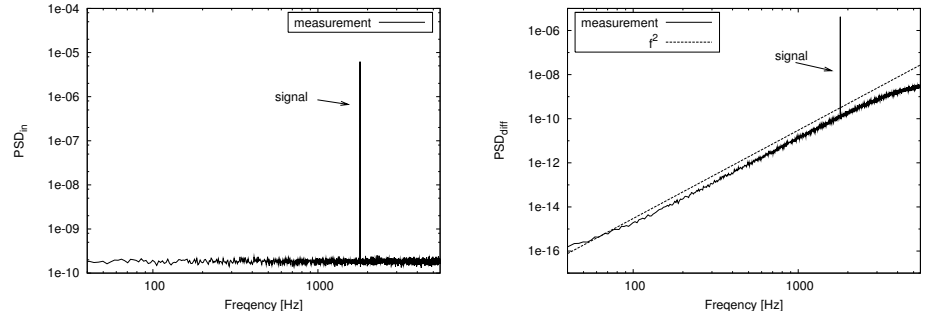
(e) Applying leaky integrator at the output of the LCD.

Figure 4.6: Time course of noise+signal at different measurement points of the system within the same experimental setup. The applied configuration was: Symmetric LCD, $U_t = 0.45, f_c = 12000Hz, A = 0.1, B = 5, \tau_0 = 0.833\mu s, f_s = 10Hz$.

4.8 SR in CNN

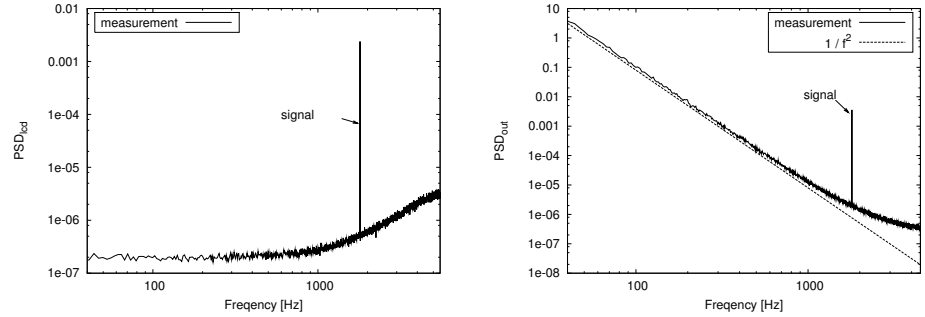
Interestingly, the phenomenon detailed above can be transfered to 2 dimensions. From this perspective, CNN seems to be an appropriate image processing system to test it against noise robustness. The significance of implementation on the phenomenon on a bio-motivated tool is underlined by the fact that many have found that SR like processes may have significant role in neuro-transmission (ref. [BV97, YYL05]) or in celebral cortex (ref. [RD01]).

However, the experiments detailed here are based on the phenomena that is analytically detailed in the former section for 1D case, the proliferation of possibilities, when extending to spatial and temporal directions in a biologically plausible way, did not let us to analyze all aspects of the theory. Indeed, we have a good reason to assume that the theory transforms to these cases since dimensions (i.e. axes)



(a) The PSD of the input: white noise and single band signal.

(b) The spectrum after derivation. Note the diversion from the ideal derivator which implies high frequency limit at about $4500Hz$.



(c) The spectrum after a symmetric LCD Note that the frequency course is mostly white with the exception at the high end.

(d) The output of the system after integration.

Figure 4.7: PSD of 500 averaged experiments with $U_t = 0.45, f_c = 12000Hz, A = 0.1, B = 5, \tau_0 = 0.833\mu s, f_s = 1800Hz, rms = 0.2341$. The number of samples used was $N = 4096$ that corresponds to $T = 0.341s$ time.

are linearly separable. For this reason, we conducted experiments that display the same phenomena, the "smooth stochastic resonance" in 2D while using CNN compatible spatial-temporal operations exclusively.

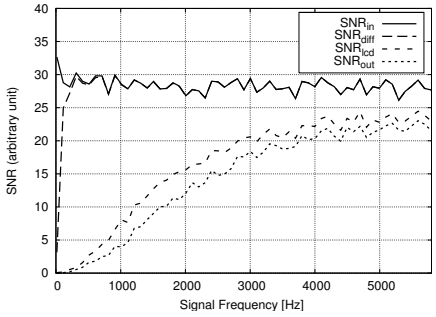
4.8.1 Stochastic Resonance in Images

The receipt we followed in the experiment was the following: Assume an image at the input, then burden it with noise, use an appropriate non-linearity and see what the optimal noise configuration is.

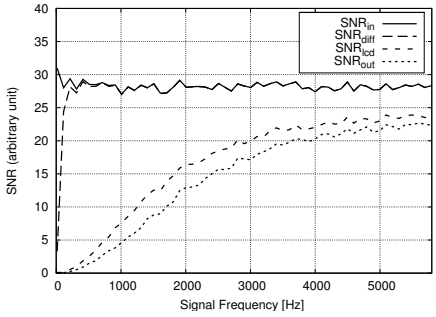
First, we used a simple threshold as a non-linearity with sub-threshold signal (image) and a Gaussian, band-limited, white noise as noise source. This is probably the simplest SRT which can be asserted. The original image (signal), along with the three different loads of noise with a fixed threshold is displayed in fig. 4.10. It is fairly easy to prove that the power of the amplitude distribution of the Gaussian noise is projected on the SNR versus RMS diagram (fig. 4.11).

4.8.2 Smooth Stochastic Resonance in Images

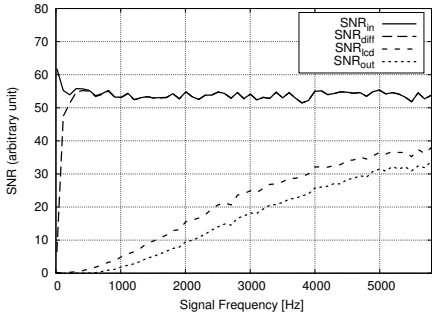
In the next series of experiment we showed that the spatial derivation and integration behaves similar to the one that we have found in case of 1D signals. In terms of image processing, the derivation and integration can be translated to different operations. In fig. 4.12, the diversity of possibilities for spatial and temporal, derivation and integration is hinted.



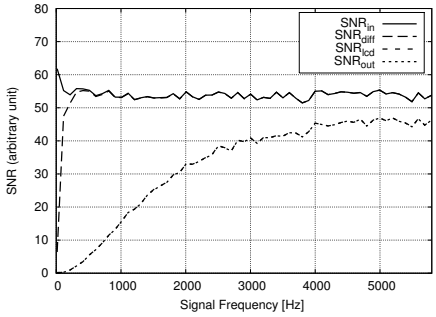
(a) The asymmetrical integro-differential system with stepwise integration.The maximal measured SNR gain is 0.827.



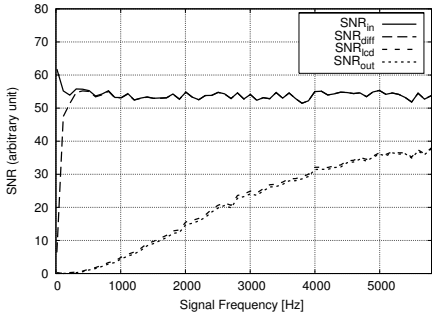
(b) The symmetrical integro-differential system with stepwise integration. The maximal measured SNR gain is 0.806.



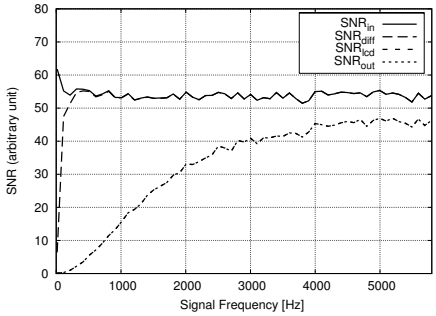
(c) The asymmetrical integro-differential system with leaky integrator ($\tau = 50/f_c$). The maximal measured SNR gain is 0.626.



(d) SNR_{out} versus signal frequency at a single rms (0.19). The symmetrical integro-differential system with leaky integration ($\tau = 50/f_c$). The maximal measured SNR gain is 0.868.



(e) The asymmetrical integro-differential system with leaky integration ($\tau = 10/f_c$). The maximal measured SNR gain is 0.70.

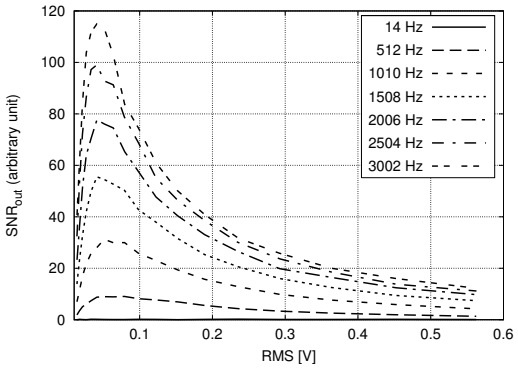


(f) The symmetrical integro-differential system with leaky integration ($\tau = 10/f_c$). The maximal measured SNR gain is 0.866.

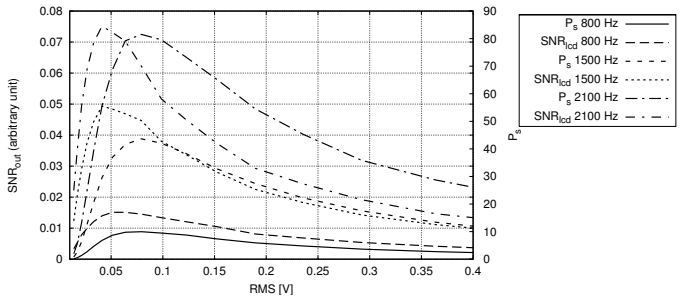
Figure 4.8: SNR_{out} versus signal frequency at a single rms (0.19) The experimental setting was as in fig. 4.7.

For the single experiment we made, we used an invertible kernel for integration, its inverse for the derivation and a Gaussian, band-limited, white noise, again, as a noise source. The three kernels that we relied on can be interpreted as templates of size $5x5$ in terms of CNN. These are as follows.

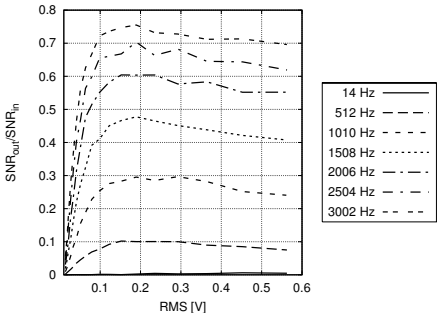
For spatial derivation



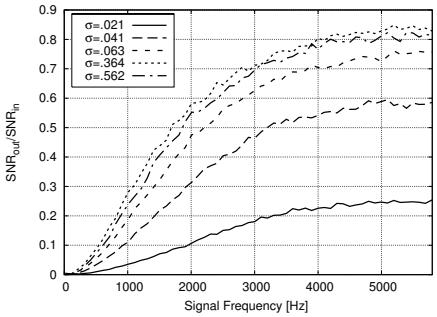
(a) SNR_{out} versus rms at different signal frequencies.



(b) SNR_{out} and signal power versus rms at different signal frequencies.



(c) SNR_{gain} versus rms at different signal frequencies.



(d) SNR_{gain} versus frequency at different rms's.

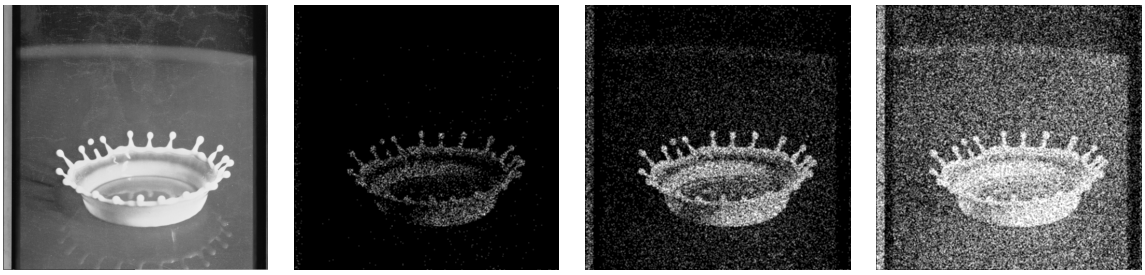
Figure 4.9: The symmetrical integro-differential system with leaky integration ($\tau = 10/f_c$). The experimental setting was as in fig. 4.8.

$$B = \begin{array}{|c|c|c|c|c|} \hline 0.04 & 0.12 & 0.20 & 0.12 & 0.04 \\ \hline 0.12 & 0.36 & 0.60 & 0.36 & 0.12 \\ \hline 0.20 & 0.60 & 0.00 & 0.60 & 0.20 \\ \hline 0.12 & 0.36 & 0.60 & 0.36 & 0.12 \\ \hline 0.04 & 0.12 & 0.20 & 0.12 & 0.04 \\ \hline \end{array}$$

while $A = 0$, $z = image$.

For the thresholding $A_{00} > 1, B = 1, z = image$.

And finally for the integration



(a) The original image (b) Noisy image with low noise level (c) Noisy image with optimal noise level (d) Noisy image with high noise level

Figure 4.10: The simplest stochastic resonator is demonstrated. Whenever an image undergoes a thresholding, then the SNR throughput can be maximized as a function of noise.

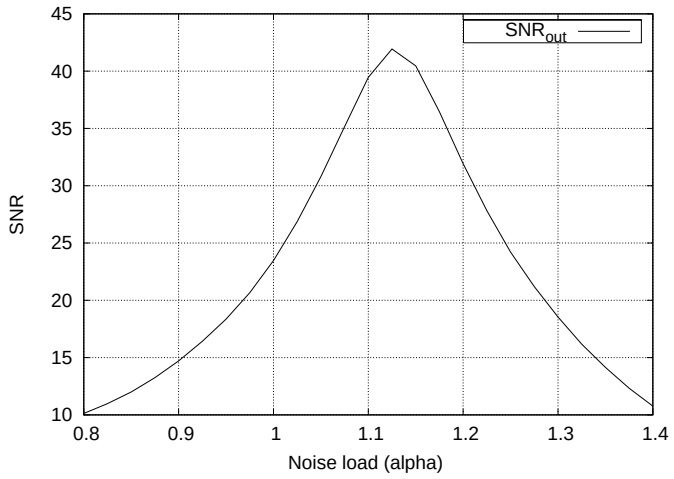


Figure 4.11: SNR_{out} is maximized as a function of noise (RMS)

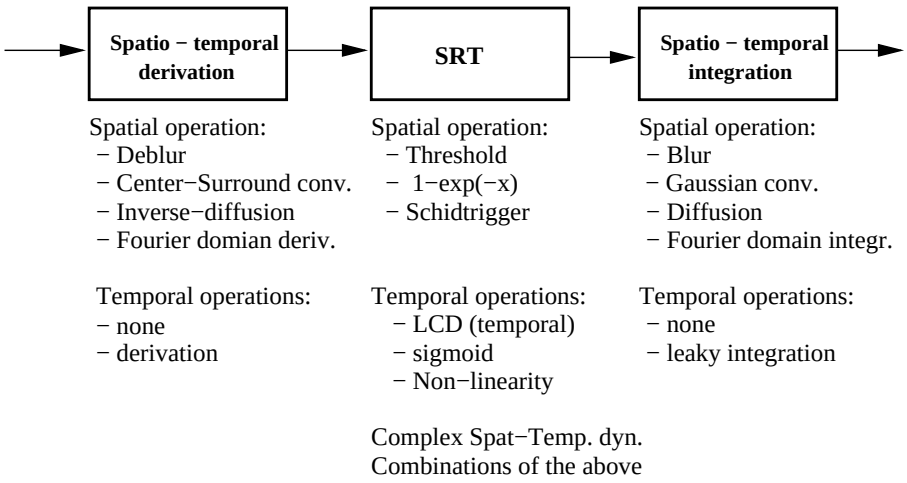


Figure 4.12: Possibilities for realizing the IDSRT in 2D.

$$A = \begin{array}{|c|c|c|c|c|} \hline -0.04 & -0.12 & -0.20 & -0.12 & -0.04 \\ \hline -0.12 & -0.36 & -0.60 & -0.36 & -0.12 \\ \hline -0.20 & -0.60 & 1.00 & -0.60 & -0.20 \\ \hline -0.12 & -0.36 & -0.60 & -0.36 & -0.12 \\ \hline -0.04 & -0.12 & -0.20 & -0.12 & -0.04 \\ \hline \end{array}$$

with $B = 1$ and $z = image$.

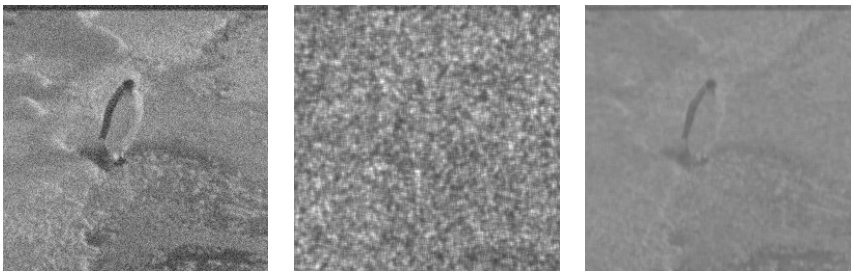
Next, the theoretical limitation of the system without noise and nonlinearity (fig. 4.13) is displayed. This simply shows the loss of information caused by the application of pre- and post- filters.

(a) The original test image (b) The test image (a) after derivation (c) The integration of (b)

Figure 4.13: The noiseless reconstruction ability of the system without non-linearty.

When noise is applied and the signal is pushed under the non-linearity, this kind of single-shot reconstruction makes non-sense. Hence the time integrated version is displayed in fig. 4.14(c).

It is worth noting that the whole system has some non-linear transmission characteristics in the frequency domain as showed first in fig. 4.5 which causes the theoretical limit hard to approach when natural signals are used as it happens in case of images.



(a) The test image after noise load (b) The image after derivation, thresholding and spatial integration (blur operation) (c) And after long term time integration

Figure 4.14: The noisy reconstruction ability of the CNN based Spatio-Temporal SRT. The torsion of the image is due to the variance of the frequency transmission ability of the IDSRT as displayed on fig. 4.9(a).

Interestingly enough, the system in this constellation shows SR behavior (see fig. 4.15), too.

Assuming an LCD in the temporal axis, which many has attributed to be the simplest model of "integrate and fire" neuron model, we can obtain the same effect as it was measured in ref. [Kis96]. Since axes are handled in a linearly separable manner (i.e. the spatial and temporal axes), results do it so, too.

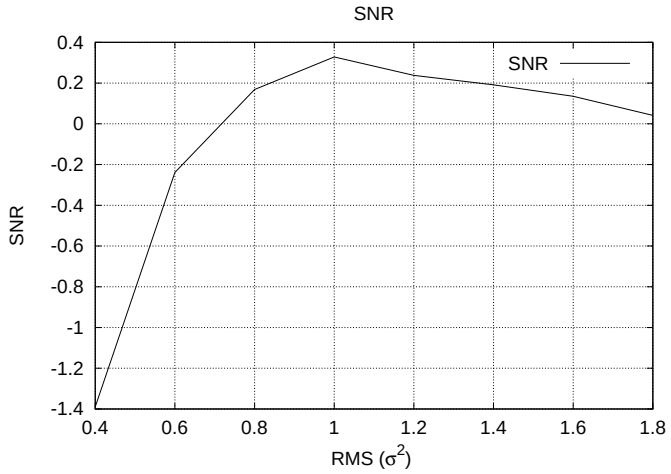


Figure 4.15: SNR_{out} can maximized as a function of noise (RMS), hence the phenomena of "smooth SR" appears in CNN, too.

Here we do not aim to give comprehensive experiments with respect to spatial-temporal smooth SR but to display the appearance of the effect which was successfully achieved.

4.9 Conclusion

We have introduced and demonstrated the viability of a new class of SR. An important characteristics of the system is that, the signal and noise characteristics are the bests at the high frequency end, where most Shannon Information is transfered. (see Ref. [BK02, Sha49, KHA01]).

For the reason that many have hinted that SR mat have significant impact in visual and in cortical processing, we have demonstrated that the phenomenon of IDSR can appear not only in single dimensional systems, but for example in spatial-temporal systems. Being the CNN a bio-motivated device, the application of CNN became a demonstrative example not only for the SR only, but for the new phenomenon, the IDSRT, too.

Appendix A

Appendix to Stability of MLCNN

A.1 Proof of Symmetric Case

Proof for Lyapunov Function in *Theorem 1*[CR02] (rewritten here).

$$V(x) = \sum_{i=1}^{2MN} \left[\int_{\theta}^{y_i} f^{-1}(v)dv \right] - \frac{1}{2} y^T A y - y^T B u - y^T z$$

Since A is positive definite,

$$\dot{V}(x) = \sum_{i=1}^{2MN} f^{-1}((y)_i)(\dot{y})_i - \dot{y}^T A y - \dot{y}^T B u - \dot{y}^T z$$

By substituting $f^{-1}(y) = x$,

$$\begin{aligned} \dot{V}(x) &= \dot{y}x - \dot{y}^T A y - \dot{y}^T B u - \dot{y}^T z \\ &= -\dot{y}^T(-x + Ay + Bu + z) \\ &= -\dot{y}^T \dot{x} \end{aligned}$$

By $\Delta f(x) \equiv \mathbf{diag}\{f'((x)_1), f'((x)_2), ..., f'((x)_{2MN})\}$ one can write $\dot{y} = \Delta f(x)\dot{x}$, which results in

$$\dot{V}(x) = -\dot{x}^T \Delta f(x) \dot{x}$$

Since $\Delta f(x)_{ij} > 0 \; \forall i, j \quad \Rightarrow \quad \dot{V}(x) < 0$.
Q.E.D

A.2 Proof of τ Symmetric Case

In the proof of *Theorem 2* we relied on the stability of the Eq. (2.15).

Definition 13. A diagonal mapping $H \in \mathcal{P}$ if and only if for $i = 1, ..., N, h_i(x_i) : \mathbb{R} \to \mathbb{R}$ is a continuous, non-decreasing bounded PWL function such that $h_i(0) = 0$.

In ref. [For02] a fundamentally new approach is used for proving the theorem in which the author claims that a neural network (N) written in the form,

$$\dot{x} = -Dx + Ah(x) + c \tag{A.1}$$

is completely stable if $h \in \mathcal{P}$, where D is diagonal, c is an arbitrary constant and A is symmetric. The *trajectory limit length theory* which Mauro Forti *et al* develops ensures stability even in cases where non-linearity (h) is totally not invertible caused piece-wise linear (PWL) with horizontal sections. These horizontal sections lead to degenerate cases (where the infinitely many non-isolated equilibrium points exist) hence, only non-strict Lyapunov function can be given. A Lyapunov function like this involves the analysis by LaSalle's invariant sets principle that may be rather complicated.

These horizontal sections play crucial role in increasing the networks learning capability (Vapnik-Chervonenkis dimension) compared to sigmoid (**tanh**()) like neural input/output activation functions. As the number of these horizontal sections increase, the pattern formation capability increases too, which simply enriches the family of stable patterns.

The following global Lyapunov function Eq. (A.2) is associated with Eq. (A.1) that overcomes the problem of the assumption of invertible $f()$. It was assumed in Appendix A, too, indeed it is rooted in the methodology of ref. [Hop84].

$$V(x) = -\frac{1}{2}h^T(x)Ah(x) - h^T(x)c + \sum_{i=1}^{N} D_i \int_0^{x_i} \rho \frac{dh_i}{d\rho} d\rho \tag{A.2}$$

which holds for all the affine systems, too.

The steps of the proof are recalled from ref. [For02] with variables remapped to our system and with standard Chua-Yang non-linearity ($f()$) applied instead of ($h(x)$) as,

1) A non-strict global energy function is introduced that overcomes the inversion of the non-invertible (PWL) input/output activation function,

$$V(x) = -\frac{1}{2}y^T A^* y - y^T(B^* u + z^*) + \sum_{i=1}^{2MN} (\tau)_{ii} \int_0^{x_i} \rho \frac{dy_i}{d\rho} d\rho.$$

The deduction of $\dot{V}(x) \leq 0$ is, practically, equivalent to the one in Appendix A, but with Dini derivatives at non-differentiable points of $f()$.

2) The existence of a function like this is exploited to prove that the total length of the trajectory

$$L(y_\infty) = \lim_{t\to\infty} \int_0^t ||\dot{y}(\sigma)||_2 d\sigma < \infty$$

is finite which means that $y(t)$ must be convergent,

3) which implies the convergence of $x(t)$, as well.

Q.E.D

A.3 Proof of a Non-symmetric Case

A typical quadratic Lyapunov function (Eq. (A.3)) is assumed in *Theorem 3* coming from ref. [Gil93] and the proof is rewritten here with slight modifications.

$$V(x) = -\frac{1}{2}y^T P y \tag{A.3}$$

We rely on the same Lyapunov function, where the existence of a D matrix is hypothesed which will symmetrize $H = (A - I)$ (I refers to the identity matrix) in order to get a symmetric P (i.e. $P = DH = H^T D^T$), positive definite matrix. By $\Delta f(x) \equiv \mathbf{diag}\{f'((x)_1), f'((x)_2), ..., f'((x)_{2MN})\}$ one can write

$$\begin{aligned} \dot{V}(x) &= -y^T P\dot{y} = -y^T P\Delta f(x)\dot{x} \\ &= -y^T P\Delta f(x)(-x + Ay) \end{aligned}$$

Because $\Delta f(x)x = \Delta f(x)y$, by substitution,

$$\begin{aligned}\dot{V}(x) &= -y^T P\Delta f(x)(-y + Ay) \\ &= -y^T P\Delta f(x)(A - I)y \\ &= -y^T P\Delta f(x)Hy \\ &= -y^T H^T D^T \Delta f(x)Hy \\ &= -y^T H^T (D^T \Delta f(x))Hy\end{aligned}$$

Since $(D^T\Delta f(x))_{ij} > 0\ \forall i,j \quad \Rightarrow \quad \dot{V}(x) < 0$

Q.E.D

Remark 6. In contrast to the original theorem, where D is assumed to be diagonal, we found that this matrix property is not required to be met, at all. However, it can be proved that only Toeplitz matrices may appear in such a role but resulting in no extra constraints on structure of matrix A. Consequently it would make no sense to complicate the calculations with these cases.

A.4 Proof of the τ-non Symmetric Case

In the proof of *Theorem 4* the quadratic Lyapunov function may be written as,

$$V(x) = -\frac{1}{2}y^T Py$$

associated to the non-unity τ system with non-symmetric templates. Hence with assuming symmetric $P = DH$ where $H = (A - \tau)$, similar to that of in Appendix C, equations develop as,

$$\begin{aligned}\dot{V}(x) &= -y^T P\dot{y} = -y^T P\Delta f(x)\dot{x} \\ &= -y^T P\Delta f(x)(-\tau x + Ay) \\ &= -y^T P\Delta f(x)(-\tau y + Ay) \\ &= -y^T P\Delta f(x)(A - \tau)y \\ &= -y^T H^T (D^T \Delta f(x))Hy\end{aligned}$$

Since $(D^T\Delta f(x))_{ij} > 0\ \forall i,j \quad \Rightarrow \quad \dot{V}(x) < 0$

Q.E.D

A.5 The Conjecture

During the support of *Conjecture 1* we relied the complete stability of a non-reciprocal 1D CNN in ref. [CR90]. In this case matrix A will have the following structure,

$$\hat{A} = \begin{pmatrix} 0 & 0 & 0 \\ \mathrm{s} & \mathrm{p} & \text{-s} \\ 0 & 0 & 0 \end{pmatrix} \qquad \tilde{A} = \begin{bmatrix} p & -s & & & \\ s & p & -s & & \\ & s & \ddots & \ddots & \\ & & \ddots & & -s \\ & & & s & p \end{bmatrix}$$

These types of matrices ($\tilde{A}$) belong to the class P, which possess the unique property that all of its principle minors are strictly positive as well as the real part of their eigenvalues.

For similar reasons, simple cases when all cells are either in the linear region (α) or in saturated region (β), lead to a Jacobian matrix with either positive eigenvalues ($\Delta f(x) = \tilde{A} - I$) only, or -1 eigenvalues ($\Delta f(x) = -I$) only, respectively. While the calculation of the principal minors in the latter case is obvious, the first one can be done by an elegant recursive rule proposed by *Chua, et al.*

Eigenvalue calculation when a single cell is in region β and the rest is in region α, may be decomposed to the above cases, which enables the author to conclude the following remarks:

Remark 7. Regarding the trajectories of anti sign-symmetric CNNs,

- No trajectory can settle outside the saturated region,
- Any trajectory in the saturated region converges to a stable equilibrium point inside the same region,
- All trajectory in the linear region tends to leave the region.

This support the conjecture of complete stability.

Appendix B

Appendix to aVLSI based OF

B.1 The diffusion PDE and the corresponding IIR equivalence

B.1.1 Diffusion PDE

Next we brief the solution of a diffusion PDE, and compare it to the use of a Gaussian window function, and ,finally, point out their equivalence.

The following differential equation describes a diffusion process

$$\frac{D}{2}\nabla f(x,t) - \frac{\partial f(x,t)}{\partial t} = 0.$$

Let the solution be $f_j(x,t)$, where j refers to the initial conditions. First, set it to

$$f_1(x, t=0) = \delta_{x,0},$$

then solution is

$$f_1(x,t) = \frac{1}{\sqrt{2\pi t}} \exp\left\{-\frac{x^2}{2t}\right\}. \tag{B.1}$$

Now, having a vector composed of samples $s(i)$ in discrete space as an initial condition as,

$$f_2(x, t=0) = \sum_i s(i)\, \delta_{x,i},$$

then the solution is,

$$f_2(x,t) = \sum_i s(i) f_1(x+i,t) = s * f_1(x,t), \tag{B.2}$$

where '$*$' denotes the convolution operator, and i iterates through the discrete space.

B.1.2 Gaussian convolution window

Infinite Impulse Response (IIR) filtering means

$$y = s * IIR \tag{B.3}$$

where y is the filter output, IIR is the filter and s is the input. If IIR is assigned to a certain Gaussian function (e.g. equ. B.1), then equ. B.3 turns to be equivalent to equ. B.2 with a proper time step (i.e. $t = t_0$), which means, applying diffusion and using convolution of Gaussian filter provides the the same result.

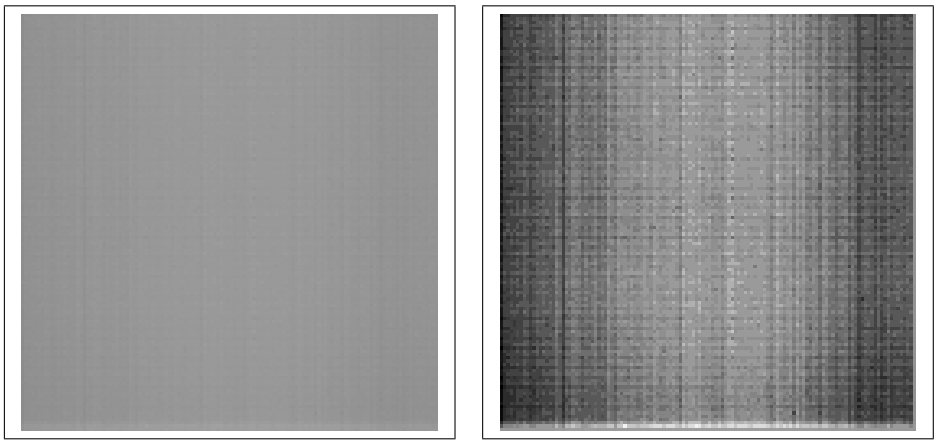
From equ. B.3, one can also conclude that if diffusion is done as an elementary operator, then Gaussian window filtering is accelerated the by $O(N^2)$ orders of magnitude.

B.2 Algorithm Complexity

The complexity of algorithms is given in this section.

First, the alg. 1 is analyzed.

- It has to calculate the DFD at every pixel $O(N^2)$, for all possible vectors $O(M^2)$,
- then, the convolution for kernel size K at every scale (S), for $O(N^2, M^2, K^2, S)$ cost.



(a) Fix-pattern noise of ACE16k after diffusion of $10\mu s$, starting from a zero level input

(b) Attenuated fix-pattern noise of (a) with offset = 139, gain = 11.0

Figure B.1: Fix-pattern noise characteristics

- Finally, get the maxima on a per pixel basis for $O(N^2, M^2)$ const.

Hence, the upper limit is $O(N^2, M^2, K^2, S)$.

Second, the complexity of alg. 3 is given.

- The DFD is done in $O(M^2)$ time.
- The generation diffused versions for $O(M^2, K)$
- Finally, get the maxima at every pixel on DSP $O(N^2, M^2)$

Note that, this solution has is only $O(N^2, M^2)$ complexity since all the rests is done in CNN, in the device that is essentially parallel and the speed of which does not depend on the number of pixels.

Probably, it is not fair to compare a sequential (Neumann) architecture to a fully parallel analog one (CNN) in terms of complexity. For this reason, we might refer to the complexity term as a *complexity in CNN sense* opposed to the traditional complexity concept; *complexity in Neumann sense*.

B.3 On-chip Diffusion

Unfortunately, analog circuits always face noise problems, hence implemented algorithms need tuned to be robust against it. For this reason, we have recorded in-diffusion fix-pattern noise transfer characteristics of the Heaviside step-function (fig. B.1) and the rectangular function on the ACE16k (fig. B.3).

Fix-pattern noise was measured by loading a zero level picture input into a LAM, then after a few micro-seconds of diffusion, it was read-back from the same LAM.

We registered the following type of errors

- There was a bias towards white (in CNN terms it is referred to as -1 and associated with the negative supply voltage) in the middle region of the chip that is probably due to the supply problems of the chip. (see fig. B.2)
- Structured noise along the y axis; the source is probably the A/D converters' unequalizedness (narrow arrows in fig. B.2) which is associated to the cells on a per column basis,

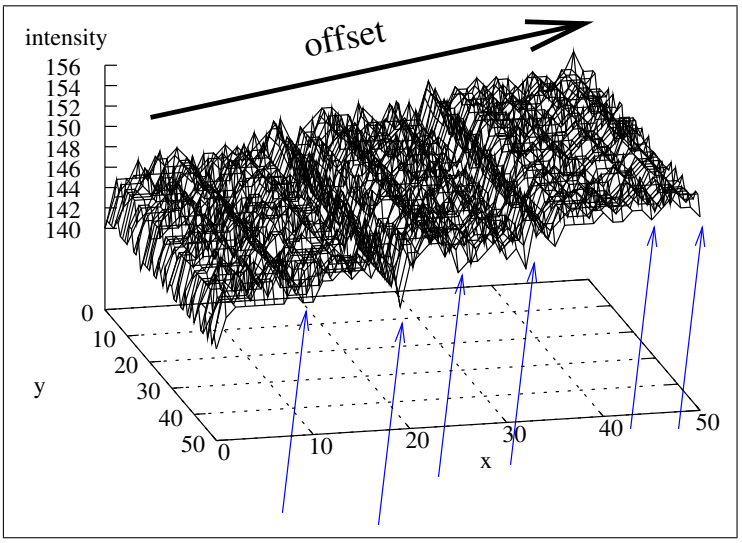


Figure B.2: 3d plot of section 0-50, 0-50 pixels (x,y) of B.1. The wide arrow points into the direction of the offset bias, while the weak arrows point to a somewhat regular noise along the y axis.

- Cell bias which is probably the least significant factor,
- Shot-noise which is fairly suppressed as multiple experiences show.

Similar to this, in recording the step function characteristics, test images (fig. B.3(a),B.3(b)) were loaded into a LAM, few micro-seconds of diffusion was applied then the image was read back. Along with the considerable fix pattern noise, note the significant zero level error, too.

B.4 Alternative model background

An alternative formulation to multi-scaling was suggested by Cs. Szepesvari in a personal communication leading, essentially, to the same results.

Let us define a multi-scale representation of the image:

$$I \rightarrow \{I_1, I_2, \ldots, I_{q_{max}}\}$$

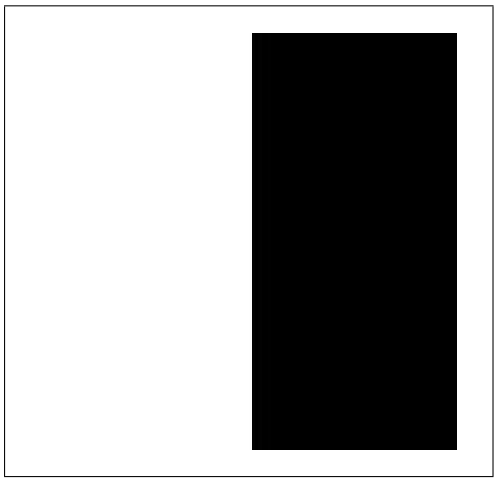
where $I_{q_{max}} \triangleq I$. The images $(I_1, I_2, \ldots, I_{q_{max}-1})$ are generated by the respective operators $f_2, \ldots, f_{q_{max}} : [-1,1]^m \rightarrow [-1,1]^m)$ by $f_q(I_q) \triangleq I_{q-1}$ where $q = 2, \ldots, q_{max}$ and where m is the number of pixels. We call I_q the image I at q^{th} scale. Here f_i is a non-invertible scaling operator. In practice f_i is a blur operation.

We assume that $I^{(2)}$, the image of the next time-step depends stochastically on $I^{(1)}$, (the current image) and the flow field $\bar{v}(\bar{v} \in \mathbb{R}^{2m})$. The following assumptions are made on this dependence.

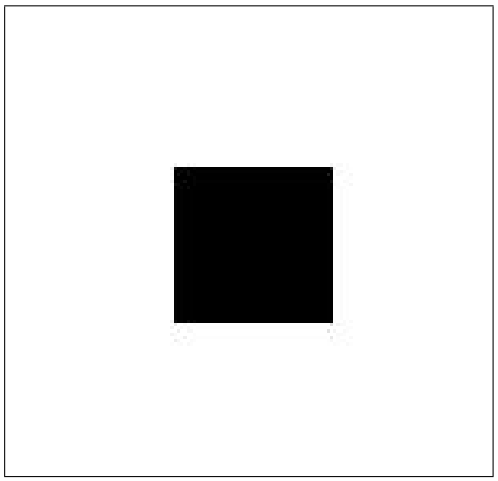
A1.1 $p(I_q^{(2)}|I_{q-1}^{(2)}, I^{(1)}, \bar{v}) = p(I_q^{(2)}|I_{q-1}^{(2)}, I_q^{(1)}, \bar{v})$ which means that the q^{th} scale of image $I^{(2)}$ depends on image $I^{(1)}$ only through the q^{th} scale of I, assuming $I_{q-1}^{(2)}$, image $I^{(1)}$ and $\bar{v}$.

A1.2 Specifically, for $q = 1$, $p(I_1^{(2)}|I^{(1)}, \bar{v}) = p(I_1^{(2)}|I_1^{(1)}, \bar{v})$.

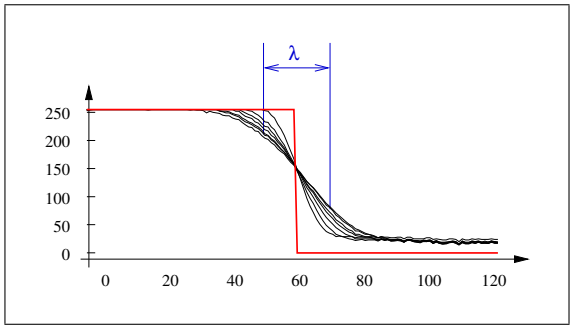
A2.1 $\mathbf{supp}\, p(I_q^{(2)} = .|I_{q-1}^{(2)}, I_q^{(1)}, \bar{v}) = f_q^{-1}(I_{q-1}^{(2)})$ where the inverse of $I_{q-1}^{(2)}$ under f_q is defined as $f_q^{-1}(I_{q-1}^{(2)}) = \left\{I_2^{(2)} | f_q(I_q^{(2)}) = I_{q-1}^{(2)}\right\}$.



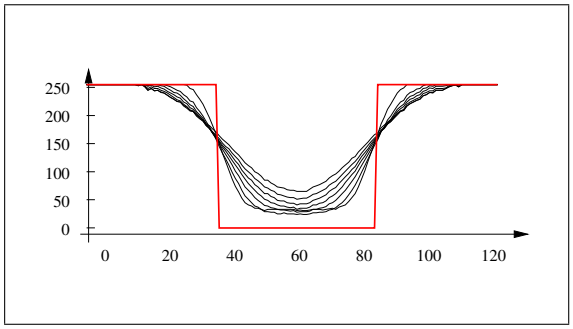
(a) The Heaviside test function



(b) The rectangular test function



(c) The evoltion of the trajectory of CNN as a response to Heaviside test function



(d) The evoltion of the trajectory of CNN as a response to rectangular test function

Figure B.3: Evolution of the intensity profile (fig. B.3(c),B.3(d)) at the horizontal center line of the images (fig. B.3(a),B.3(b)) when subject to diffusion. The maximal diffusion time applied in these experiments was $10\mu s$. Within this period the *full-length half-width* (λ) distance was 20 pixels.

A2.2 For any $I_q^{(2)} \in \mathbf{supp}\, p(I_q^{(2)} = .|I_{q-1}^{(2)}, I_q^{(1)}, \bar{v})$ and $p(I_q^{(2)}|I_{q-1}^{(2)}, I_q^{(1)}, \bar{v}) = \exp\left(-\lambda_q\, DFD(I_q^{(2)}, I_q^{(1)}, \bar{v})\right)$ with $\lambda_q > 0$.

A2.3 Specifically, for $q = 1, p(I_1^{(2)}|I_1^{(1)}, \bar{v}) \propto \exp\left(-\lambda_q\, DFD(I_1^{(2)}, I_1^{(1)}, \bar{v})\right)$

Our aim is to compute the MAP estimate of $\bar{v}$ given $I^{(1)}, I^{(2)}$ and $\bar{v}$, i.e. $\arg\max_{\bar{v}} p(\bar{v}|I^{(1)}, I^{(2)})$. By Bayes theorem and dropping the normalization term,

$$p(\bar{v}|I^{(2)}, I^{(1)}) \propto p(I^{(2)}|I^{(1)}, \bar{v})\, p(\bar{v})$$

By the definition of $I_{q_{max}}^{(2)}$, $p(I^{(2)}|I^{(1)}, \bar{v}) \triangleq p(I_{q_{max}}^{(2)}|I^{(1)}, \bar{v})$. Let us define an expression for $p(I_q^{(2)}|I^{(1)}, \bar{v})$, $q = 2, \ldots, q_{max}$. By the properties of conditional likelihood, since $I_{q-1}^{(2)}$ is a function of $I_q^{(2)}$,

$$p(I_q^{(2)}|I^{(1)}, \bar{v}) = p(I_q^{(2)}|I_{q-1}^{(2)}, I^{(1)}, \bar{v})\, p(I_{q-1}^{(2)}|I^{(1)}, \bar{v}). \tag{B.4}$$

The equation allows us to derive a recursive expression for $p(I_q^{(2)}|I^{(1)}, \bar{v})$. Recognize the recursion in this expression (i.e. left hand side of the equation and the last term). From A1.1 and A2.2

$$p(I_q^{(2)}|I_{q-1}^{(2)}, I^{(1)}, \bar{v}) = p(I_q^{(2)}|I_{q-1}^{(2)}, I_q^{(1)}, \bar{v}) = \exp\left(-\lambda_q\, DFD_q(I^{(2)}, I^{(1)}, \bar{v})\right)$$

Hence, by equ. B.4 and using A1.2 and A2.3, we get

$$p(I^{(2)}|I^{(1)},\bar{v}) \propto \exp\left(-\sum_{q=1}^{q_{max}} \lambda_q \, DFD_q(I^{(2)},I^{(1)},\bar{v})\right).$$

Because we have no smoothness constraint on $\bar{v}$, this becomes a separable expression of $v_1,\ldots,v_m$:

$$\sum_{q=1}^{q_{max}} \lambda_q \, DFD_q(I^{(2)},I^{(1)},\bar{v}_i) \tag{B.5}$$

hence, the MAP estimate of $\bar{v}$ can be obtained componentwise and is the same as equ. 3.7 extended with the ability to weight scales with λ_i.

Appendix C

Appendix to Integro-Differential SR

C.1 Analysis of Level Crossing Detector

Here we brief the results of Kish from Ref. [Kis96] with respect to a signal in the linear response limit of an LCD (see Def. 1. 2.) fed by white noise and sinusoid signal.

The question of our interest is the maximal SNR gain ($= SNR_{in}/SNR_{out}$) that can be obtained from this system. SNR_{in} can be formalized in a way as in Sec. 4.6.1 so Eq. 4.2 still holds.

To determine $SNR_{out} = SNR_{lcd}$, first, the mean impulse production frequency ($\nu(U_t)$) is given as a function of constant input $s(t) = 0$ with fixed threshold U_t (see Rice formula in Ref. [Ric44, Ric45] for zero crossings and its generalization, for example, in Ref. [Kis96] for non-zero threshold levels) as

$$\nu(U_t) = \frac{2}{\sigma}\left(\int_0^{f_c} f^2 S_{in}(f)df\right)^{\frac{1}{2}} \mathbf{exp}\left\{\frac{-U_t^2}{2\sigma_{in}^2}\right\} \tag{C.1}$$

adding a signal $s(t)$ to the noise is essentially the same as modulating the threshold U_t as $U_t^s = U_t - s(t)$, hence

$$\nu(U_t^s) = \underbrace{\frac{1}{\sigma_{in}}\left(\int_0^{f_c} f^2 S_{in}(f)df\right)^{\frac{1}{2}}}_{\nu_0} \cdot \mathbf{exp}\left\{\frac{-(U_t - s(t))^2}{2\sigma_{in}^2}\right\}. \tag{C.2}$$

As a part of the expression, ν_0 denotes the mean uni-directional zero crossing frequency of the input noise.

Assuming a white PSD for the noise ($S_{in}(f) = S_{in} = \sigma_{in}^2/f_c$), ν_0 can be written as,

$$\nu_0 = \frac{1}{\sigma_{in}}\left(S_{in}\int_0^{f_c} f^2 df\right)^{\frac{1}{2}} = \frac{1}{\sigma_{in}}\sqrt{S_{in}}\sqrt{\frac{f_c^3}{3}} = \frac{f_c}{\sqrt{3}} \tag{C.3}$$

and the Taylor expansion of Eq. C.2 around U_t is

$$\begin{aligned}\nu(U_t^s) &= \left(1 + \frac{-U_t}{\sigma_{in}^2}s(t) + \ldots + h.o.t.\right)\nu(U_t) \\ &\approx \nu(U_t) + \underbrace{\left(\frac{-U_t}{\sigma_{in}^2}\nu(U_t)\right)}_{\nu_{gain}(U_t)} s(t) = \nu(U_t) + \nu_{gain}(U_t)s(t)\end{aligned} \tag{C.4}$$

where *h.o.t.* refers to *higher order terms*. Note that, the contribution of the signal is merely the last term. Since the signal is very slow compared to the mean firing rate of the Poisson process. (i.e. $f_s \ll \nu$)), the time average of the impulse sequence emerged by reason of the modulation can be approximated as

$$U_{av} = B\tau_0\nu_{gain}(U_t)s(t). \tag{C.5}$$

A device like this can be interpreted as an impulse-width modulator where the modulation is a sinus wave, which has the power

$$P_{lcd} = (B\tau_0\nu_{gain}(U_t)A)^2.$$

Not like this is the background noise, which is a steady Poisson process with $\nu(U_t)$ mean repetition rate.

Campbell's theorem (for $f > 0$) applies to this as

$$S_{lcd}(f) = \nu(U_t)B^2\frac{(\mathbf{sin}(\omega\tau_0))^2}{\omega^2}.$$

In the low frequency range, $\mathbf{sin}(x)/x \to 1$ so

$$S_{lcd} = \nu(U_t)B^2\tau_0^2$$

which leads to

$$SNR_{lcd} = \frac{\nu_{gain}(U_t)^2}{\nu(U_t)}A^2.$$

By substituting ν_{gain},

$$SNR_{lcd} = \frac{\frac{U_t^2}{\sigma_{in}^4}\,\nu(U_t)^2A^2}{\nu(U_t)} = \frac{U_t^2}{\sigma_{in}^4}\nu(U_t)A^2. \tag{C.6}$$

Finally with Eq. 4.2 in the denominator, the SNR gain (μ) is

$$\mu = \frac{SNR_{lcd}}{SNR_{in}} = \frac{(U_t^2/\sigma_{in}^4)\,\nu(U_t)A^2}{A^2f_c/(2\sigma_{in}^2)} = \frac{2}{f_c}\frac{U_t^2}{\sigma_{in}^2}\nu(U_t). \tag{C.7}$$

By substituting Eq. C.1 (i.e. $\nu(U_t)$) in this by means of Eq. C.3,

$$\mu = \frac{2}{f_c}\frac{U_t^2}{\sigma_{in}^2}\frac{f_c}{\sqrt{3}}\mathbf{exp}\left\{-\frac{U_t^2}{2\sigma_{in}^2}\right\} = \frac{2}{\sqrt{3}}\,\frac{U_t^2}{\sigma_{in}^2}\,\mathbf{exp}\left\{-\frac{U_t^2}{2\sigma_{in}^2}\right\}.$$

This means that the SNR gain is independent of the applied signal frequency. Indeed, it seems to be independent of signal amplitude, too, but this is only due to linearization we made in Eq. C.4, so it is dependent but only up to the higher order terms effect.

To get the maxima consider using $\alpha = \frac{U_t^2}{2\sigma_{in}^2}$ so

$$\mu = \frac{1}{\sqrt{3}}\,\alpha\,\mathbf{exp}\{-\alpha\}$$

from which it can be seen that $d\mu/d\alpha = 0 \to |\alpha_{ext}| = 1$, hence the maxima is

$$\mathbf{max}\left\{\frac{SNR_{lcd}}{SNR_{in}}\right\} = \frac{4}{\mathbf{e}\sqrt{3}} \tag{C.8}$$

C.2 Symmetric Level Crossing Detector

Symmetric LCD's SNR can be concluded by assuming two asymmetric LCDs with $U_t^+ = U_t$ and $U_t^- = -U_t$ in Eq. C.3, so the positive (ν^+) and negative threshold crossing rates (ν^-) are,

$$\begin{aligned}\nu^+(U_t^s) &= \nu_0 \cdot \mathbf{exp}\left\{\frac{-(U_t - s(t))^2}{2\sigma_{in}^2}\right\}\\ \nu^-(U_t^s) &= \nu_0 \cdot \mathbf{exp}\left\{\frac{-(-U_t - s(t))^2}{2\sigma_{in}^2}\right\}\end{aligned} \tag{C.9}$$

After linearization as in Eq. C.4,

$$\begin{aligned}\nu^+(U_t) &= \nu(U_t) + \nu_{gain}(U_t)s(t)\\ \nu^-(U_t) &= \nu(U_t) - \nu_{gain}(U_t)s(t).\end{aligned}$$

Since the positive and negative impulses,

$$\begin{aligned}U_{av}^+ &= B\,\tau_0\nu_{gain}(U_t)s(t)\\ U_{av}^- &= (-B)\tau_0\nu_{gain}(U_t)(-s(t))\end{aligned}$$

are summed in the time domain, the average value

$$U_{av}^{sym} = U_{av}^+ + U_{av}^- = 2\,U_{av}$$

in terms of Eq. C.5, which means the signal power is $P_{lcd}^{sym} = 4P_{lcd}$, while the background noise power is doubled only (i.e. $S_{lcd}^{sym} = 2S_{lcd}$) since the two independent Poisson processes' power can be added. Thus their ratio, the SNR, is doubled compared to the asymmetric case ($SNR_{lcd}^{sym} = 2SNR_{lcd}$) which means the SNR gain is doubled, as well. By Eq. C.8, the maximal SNR gain, can be written in general form as,

$$\mathbf{max}\left\{\frac{SNR_{lcd}}{SNR_{in}}\right\} = \frac{4z}{\mathbf{e}\sqrt{3}}$$

where z is 1 in asymmetric and 2 in symmetric LCD case.

This result is important since the this is the first time, as far as we are concerned, to get real SNR gain in the small signal limit with sinusoidal signals.

Bibliography

[AB86] E.H. Adelson and J.R. Bergen. The extraction of spatio temporal energy in machine and human vision. In *Proc. IEEE Workshop on Visual Motion*, pages 151–166, 1986.

[ADK99] G. Aubert, R. Deriche, and P. Knorprobst. Computing of via variational techniques. *SIAM, J. of Appl. Math (JAM)*, 60(1):156–182, 1999.

[AK02] G. Aubert and P. Knorprobst. *Mathematcial Problems in Image Processing,Partial Differential Equations and Calculus of Variations*. Springer-Verlag, New York, 2002.

[Ana89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Intl. Jour. of Computer Vision (IJCV)*, 2:283–310, 1989.

[BAK91] R. Battiti, E. Amaldi, and C. Koch. Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy. *Intl. Jour. of Computer Vision (IJCV)*, 2(6):133–145, 1991.

[BAS81] R. Benzi and A. Vulpiani A. Sutera. The mechanism of stochastic resonance. *Jour. of Phys: Math. Gen.*, 14:453–457, 1981.

[BB95] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In J. Matas T. Pajdla, editor, *Computer Vision - ECCV 2004, Lecture Notes in Computer Science*, volume 3024, pages 25–36, Berlin, 2004. Springer.

[BJ96] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. Technical Report RBVC-TR-96-50, Technical Report at Univ. of Toronto, Dept. of Comp. Sci, Oct 1996.

[BK02] S. M. Bezrukov and L. B. Kish. How much power does neural signal propagation need? In *Smart Mater Struct*, pages 800–803, 2002.

[Bla] Michael J. Black. *Yosemite image sequences*. http://www.cs.brown.edu/people/black/Sequences/yosemi

[BV97] S. M. Bezrukov and I. Vodyanoy. Stochastic resonance in non-dynamical systems without response thresholds. *Nature (NAT)*, 385:19–321, 1997.

[BWS05] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *Intl. Jour. of Computer Vision (IJCV)*, 61(3):211–231, 2005.

[CC92] L. O. Chua and C.W.Wu. On the universe of stable cellular neural networks. *International Journal of Circuits Theory and Applications (CTA)*, 20:497–518, 1992.

[CG03] F. Cortino and M. Gilli. Comparision between the dynamic behavior of chua-yang and full-range cellular neural networks. *International Journal of Circuits Theory and Applications (CTA)*, 31(5):423–441, 2003.

[CR90] L. O. Chua and T. Roska. Stability of a class of nonreciprocal cellular neural networks. *IEEE Trans. on Circuits and Systems (TCAS)*, 37(12):1520–1527, 1990.

[CR02] L. O. Chua and T. Roska. *Back to basics: Nonlinear dynamics and complete stability*, chapter 8, pages 205–231. Cambridge University Press, Cambridge, 2002.

[CS91] L. O. Chua and B.E. Shi. *Multiple layer cellular neural networks*, chapter Tutorials, pages 137–168. Elsevier Science Publishers B.V., Amsterdam, 1991.

[CTTA00] Rekeczky Cs., Serrano T., Roska T., and Rodríguez-Vázquez A. A stored program 2nd order / 3-layer complex cell cnn-um. In *Proc. of the 6th IEEE Int. Workshop on Cellular Neural Net. and Their Appl.*, pages 213–218, Catania Piscataway, 2000.

[CY88a] L. O. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Trans. on Circuits and Systems (TCAS)*, 35(10):1257–1272, Oct 1988.

[CY88b] L. O. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Trans. on Circuits and Systems II (TCAS)*, 35:1273–1290, Oct 1988.

[ECdSM97] R. Etienne-Cummings, J. Van der Spiegel, and P. Müller. A focal plane visual motion measurement sensor. *IEEE Trans. on Circuits and Systems I (TCAS)*, 44(1):55–56, Jan 1997.

[Enc86] W. Enckelmann. Investigation of multigrid algorithms for the estimation of optical flow fields in images sequences. In *Proc of. IEEE Workshop on Motion: Representation and Analysis*, pages 81–87, Charleston, South Carolina, May 1986.

[ERVDVC94] S. Espejo, A. Rodríguez-Vázquez, R. Dominíguez-Vázquez, and R. Carmona. Convergence and stability of the fsr cnn model. In *Proc. of IEEE Int. Workshop on Cellular Neural Networks and Their App.*, pages 411–416, Rome, 1994.

[FLMM93] M. Forti, A. Liberatore, S. Manetti, and M. Marini. Global asymptotic stability for a class of nonsymmetric neural networks. In *IEEE Int. Sim. on Circ. And Sys. (ISCAS'93)*, volume 4, pages 2580–2583, Chicago, 1993.

[For02] M. Forti. Some extension of a new method to analyze complete stability of neural networks. *IEEE Trans. on Neural Networks (NN)*, 13(5):1230–1238, 2002.

[Gil93] M. Gilli. A lyapunov functional approach to the study of the stability of cellular neural networks. In *IEEE Int. Sim. on Circ. And Sys. (ISCAS'93)*, volume 4, pages 2584–2587, Chicago, 1993.

[GL96] F. Guichard and L.Rudin. Accurate estimation of discontinuous optical flow by minimizing divergence related functionals. In *Proc. Int'l Conf. on Image Proc. (ICIP96)*, volume 1, pages 497–500, Lausanne Switzerland, Sept 1996.

[Gla87] F. C. Glazer. Computation of optical flow by multilevel relaxation. Technical Report COINS-TR-87-64, Univ. of Massachusetts, 1987.

[Hee88] D. J. Heeger. Optical flow using spatio-temporal filters. *Intl. Jour. of Computer Vision (IJCV)*, 1:279–302, 1988.

[Hir85] Morris W. Hirsch. System of differential equations that are competitive or cooperative ii: Convergence almost everywhere. *SIAM, J. of Appl. Math (JAM)*, 16(3):185–203, 1985.

[HKLM88] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer*, 21(3):52–64, March 1988.

[HKSL90] J. Harris, C. Koch, E. Staats, and J. Luo. Analog hardware for detecting discontinuities in early vision. *Intl. Jour. of Computer Vision (IJCV)*, 4:211–223, 1990.

[Hop84] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. In *Proc. of National Acad. of Sci.*, volume 81, pages 3088–3092, 1984.

[HS81] B.K.P. Horn and G. Schunk. Determining optical flow. *Artificial Intelligence (AI)*, 17:185–203, 1981.

[HS02] C. M. Higgins and S. A. Shams. A bilogocially inspired modular vlsi system for visual measurement of self-motion. *IEEE Trans. on Sensors and Systems (SS)*, 2(6):508–528, Dec 2002.

[KHA01] L. B. Kiss, G. P. Hammer, and D. Abbott. Information transfer rate of neurons: stochastic resonance of shannon's information channel capacity. *Fluct. Noise. Lett. (FNL)*, 1:13–17, 2001.

[Kis96] L. B. Kiss. Possible breakthrough: Significant improvement of signal to noise ratio by stochastic resonance. In *Chaotic, Fractals and Nonlinear Signal Processing, Mystic*, volume 375, pages 880–897, Connecticut, USA, 1996. American Institute of Physics Press.

[KM96] C. Koch and B. Mathur. Neuromorphic vision chips. *IEEE Spectrum*, 33(5):38–46, May 1996.

[Kos88] B. Kosko. Bi-directional associative memory. *IEEE Trans. on Sys., Manag. and Cyb. (TSMC)*, 18:49–60, 1988.

[KSK97] J. Kramer, R. Sarpeshkar, and C. Koch. Pulse-based analog vlsi velocity sensors. *IEEE Trans. on Circuits and Systems II (TCAS)*, 44:86–101, Feb 1997.

[LaS67] J. P. LaSalle. *An invariant principle in the theory of stability*, pages 277–286. Academic Press, 1967.

[Liu98] S. C. Liu. Silicon retina with adaptive filtering properties. In S. A. Solla M. I. Jordan, M.J. Kearns, editor, *Advances in Neural Information Processing Systems (ANIPS)*, volume 10, pages 712–718. MIT Press, 1998.

[LK81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of image understanding workshop*, pages 121–130, 1981.

[LM95] R. Laganiere and A. Mitiche. Direct bayesian interpretation of visual motion. *Robotics and Autonomous Systems (RAS)*, 14(4):247–254, June 1995.

[MEC04] S. Mehta and R. Etienne-Cummings. Normal optical flow measurement on a cmos aps images. In *IEEE Int. Simp. on Circ. And Sys. (ISCAS'04)*, pages 23–26, Vancouver, Canada, May 2004.

[MP] Etienne Mémin and Patrick Pérez. Dense estimation and object based segmentation of the optical flow with robust techniques. *IEEE Trans. on Image Processing (IP)*, 7(5):703.

[MZC+05] Jose' L. Martl'n, Aitzol Zuloaga, Carlos Cuadrado, Jesu's La' zaro, and Unai Bidarte. Hardware implementation of optical flow constraint equation using fpgas. *Comp. Vision and Image Understanding (CVIU)*, 98:462–490, 2005.

[Rad00] A. Radványi. On the rectangular grid representation of general cnn networks. In *Proc. of the 6th IEEE Int. Workshop on Cellular Neural Net. and Their Appl.*, pages 387–393, Catania Piscataway, 2000.

[Ran87] R. B. Randall. *Detectors*, chapter 3.2, pages 87–101. (Bruel & Kjaer) J.T.Broch & C.G. Wahrman, 1987.

[RC93] T. Roska and L. O. Chua. The cnn universal machine. *IEEE Trans. on Circuits and Systems II (TCAS)*, 40:163–173, 1993.

[RD01] M. Rudolph and A. Destexhe. Do neocortical pyramidal neurons display stochastic resonance? *Jour. of Comp. Neurosci.*, 11:19–42, 2001.

[RF04] F. Ruffier and N. Franceschini. Visual guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In *Proc. of Int'l Conf. on Robotics & Autom (ICRA2004)*, New Orleans, April 2004.

[RFR+02] Carmona G. R., Jiménez Garrido F., Domínguez Castro R., Espejo Meana S., and Rodríguez Vázquez A. *CACE1K User s Guide*. Instituto de Microelectrónica de Sevilla, Sevilla, 2002.

[Ric44] S. O. Rice. Mathematical analysis of random noise 1. *Bell. Systems Technological Journal*, 23, 1944.

[Ric45] S. O. Rice. Mathematical analysis of random noise 2. *Bell. Systems Technological Journal*, 24, 1945.

[SD04] A. Stocker and R. Douglas. Analog integrated 2-d optical flow sensor with programmable pixels. In *IEEE Int. Simp. on Circ. And Sys. (ISCAS'04)*, volume 3, pages 9–12, Vancouver, Canada, May 2004.

[Sha49] C. E. Shannon. Communication in the presence of noise. In *Proc. IRE*, volume 37, pages 10–21, 1949.

[Sim99] Eero P. Simoncelli. *Bayesian Multi-Scale Differential Optical Flow*, volume 2, chapter 14, pages 397–422. Academic Press, April 1999.

[SJ96] S.N.Gupta and J.L.Prince. On div-curl regularization for motion estimation in 3-d volumetric imaging. In *Proc. Int'l Conf. on Image Proc. (ICIP96)*, volume 1, pages 929–932, Lausanne Switzerland, Sept 1996.

[SL01] B. Szatmáry and A. Lörincz. Indepent component analysis of temporal sequences subject to constraints by lgn inputs yields all the three major cell types of the primary visual cortex. *Jour. of Computational Neurosci. (JCN)*, 11:241–248, 2001.

[SRC98] B. Shi, T. Roska, and L. O. Chua. Estimating optical flow with cellular neural networks. *International Journal of Circuits Theory and Applications (CTA)*, 26:343–364, 1998.

[UGVT88] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cibernetics (BC)*, 60:79–87, 1988.

[WA98] Y. Weiss and E.H. Adelson. Slow and smooth: Bayesian theory of combination of local motion signals in human vision. Technical Report Tech. Rep 1624, AI Lab. Massachusetts Institute of Technology, Boston, Cambridge, MA, 1998.

[WC99] Ranxiao Frances Wang and James E. Cutting. A probabilistic model for recovering camera translation. *Comp. Vision and Image Understanding (CVIU)*, 76(3):205–212, Dec 1999.

[Wit83] A. P. Witkin. Scale space filtering. In *Proc. of Int. Joint Conference on Artificial Intelligence (IJCAI)*, Karlsruhe Germany, 1983.

[WM95] K. Wiesenfeld and F. Moss. Stochastic resonance and the benefits of noise: from ice ages to crayfish and squids. *Nature*, 373:33–36, 1995.

[YYL05] R. Romero Y. Yu and T. S. Lee. Preference of sensory neural coding for 1=f signals. *Phys. Rev. Lett. (PRL)*, 94:1–4, 2005.

[ZDCE02] Á. Zarándy, R. Domínigues-Castro, and S. Espejo. Ultra-high frame rate focal plane image sensor and processor. *IEEE Trans. on Sensors and Systems (SS)*, 2(6):559–565, Dec 2002.

[Zel04] J. S. Zelek. Bayesian real-time optical flow. *Elsevier Vision & Image Computing Journal (VICJ)*, 22:1051–1069, 2004.

[ZR04] Á. Zarándy and Cs. Rekeczky. Bi-i: a standalone cellular vision system, part i. architecture and ultra high frame rate processing examples. In *Proc. of the 8th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA2004)*, pages 4–9, Budapest, 2004.