TEMPLATE LIBRARY FOR PROGRAMMABLE OPTICAL ANALOGIC ARRAY COMPUTER (POAC)

PROGRAMMING OF OPTICAL CELLULAR NEURAL NETWORK (CNN)

Ph.D. Dissertation

Ahmed El Sayed Ayoub

Analogic and Neural Computing Systems Laboratory

Computer and Automation Institute

Hungarian Academy of Sciences

Scientific advisor

Szabolcs Tőkés, Ph.D.

Budapest, 2004

TO WHOM IT MAY CONCERN

We herby certify that this is a typical copy of the original Doctor Thesis of Mr. **Ahmed El Sayed Ayoub** born in **December 01, 1972 El Sharkia, Egypt.**

Official Seal of the

Faculty of Electrical

Engineering and Informatics

Budapest University of Technology and Economics

"for the memory of my father…"

TABLE OF CONTENTS

ACKNOWLEDGMENTS

This Ph.D. project is a fruit of this wonderful organized tree of cooperative work of the Analogic and Neural Computing Laboratory, MTA-SZTAKI that is being run by the faithful support of Professor Tamás Roska. I do thank him to provide the chance of updating my scientific skills in this excellent environment.

I do thank my supervisor Dr. Szabolcs Tőkés who had save no effort to provide all of his scientific and technical experience available for me. The years I spent in his companion within the Optical Computing Group have their own remarkable features that had shaped a major part of my forthcoming career as a member of the scientific community.

All my deep respect and appreciation to the Hungarian Ministry of Education and the Egyptian Department General for Scholarships for funding this project. Special regards to Mr. András Tokai, director of the International Coordination Department of Budapest University of Technology and Economics and Prof. Endre Selényi the deputy dean of science affairs at the Faculty of Engineering and Informatics for providing and supporting this scholarship. I do thank all of the Egyptian cultural consolers in Budapest to make funding this project possible.

For all of my colleagues and friends who helped me to overcome this period of time in Hungary I do thank them. Special thanks for Hajnalka Fellner for her sincere guidance and for Zsuzsa Tolnai for her enthusiasm to teach me her native language.

The permanent support of my dear family members can never be thanked by words.

PREFACE

Introduction

Cellular neural/nonlinear networks (CNNs) are regular, single or multi-layer, parallel processing structures with analog nonlinear computing units (base cells). The state values of the individual processors are continuous in time and their connectivity is local in space. The program of these networks is completely determined by the pattern of the local interactions, the so-called template, and the local logic and arithmetic (analog) instructions. The time-evolution of the analog transient, driven by the template operator and the processor dynamics, represents the computation in CNN. Results can be defined both in equilibrium and/or non-equilibrium states of the network. Completing the base cells of CNN with local sensors, local data memories, arithmetical and logical units, furthermore with global program memories and control units results in the CNN universal machine (CNN-UM) architecture. The CNN-UM is an analogic (analog and logic) supercomputer, it is universal both in Turing sense as a nonlinear operator, therefore it can be used as a general architectural framework when designing CNN processors. Up to the present there have been various physical implementations of this architecture: mixed-signal VLSI, emulated digital VLSI and optical.

The Programmable Optical Array/Analogic Computer (POAC) is an optical implementation of CNN. It is based on coherent and/or incoherent modified Joint Fourier Transform Correlator (t_2-JTC). Its high capabilities, specifically full parallelism and large array size, provided by the physics of light makes it a candidate for achieving high frame rate of processing.

In this dissertation the focus is put on understanding the present architectures of POAC and find new methods to enhance their functions. Moreover, one goal connected to my research is to find and model a template library for several optical processing tasks. Computer simulations are made for different CNN optical computer architectures, some of them had been investigated experimentally. The proposed optical computing system is to

accomplish image-processing algorithms, including image classification and recognition tasks.

Structure of Dissertation

Chapter 1 gives an introduction to optical computing and its advancements. It also discusses how to involve optics into computing. **Chapter 2** concerns with Cellular Neural Networks (CNN) paradigm and CNN universal machine (CNN-UM) . It shows CNN Core cell and the inter-cell interactions; CNN classes of operations with special regards towards optical implementaiton; and the CNN-UM chip architecture. **Chapter 3** provides a new classification of the integrated optical devices. It also introduces photonic crystals one candidate to perform as pre-programmed optical channels and switches. **Chapter 4** demonstrates the evolution of the programmable optical array/analogic computer (POAC) in chronological order. Starting with its optical memory and going through its several optical architectures then ends with the state of the art laptop POAC version. These chapters can serve as *teaching material*, in addition to text books, for graduate and undergraduate students specialized in the field of cellular neural networks and/or optical computing.

Chapter 5 is one contribution of this dissertation provides a new technique to implement negative values in optics. It also points out the differences between coherent and incoherent correlation from a new practical point of view. **Chapter 6** is a major contribution of this work that shows how the first optical template library for POAC had been developed. The POAC simulator algorithm is also detailed. Furthermore, it provides a classification of optical operations that might be developed in the future. **Chapter 7** concludes this doctoral work by summarizing its new theses. Each chapter is over and done with "*discussion and conclusion*" section to summarize its contents followed by "*one step forward*" section that links the next chapter as a brief introduction.

Appendices show the sub-work connected to the main topic of this dissertation. Mainly, **Appendix A** shows a compilation of several templates and operations that would be

required for optical information processing. **Appendix B** demonstrates my results of the *computer generated holograms* (CGHs) fabricated towards searching for a new optical computing devices, *diffractive optical processors* (DOPs). **Appendix C** provides the hardware driver for *acousto optical deflectors* used in POAC architectures. **Appendix D** includes the technical specifications of Laptop POAC key elements and devices.

The Attached Compact Disk (CD)

I had attached an optical compact disk (CD) to the back cover of this dessertation that includes more details about this work and other related work. I organized it as simple as possible to access its contents via HTML web-based desgin. It includes MATLAB source code for all of my custom software developed for simulators and hologram generators. In addition, it includes a list of all of my coauthored and related POAC publications since 1999 and upto date. Furthermore, I had included presentations that can be very useful for *teaching purposes*. An electronic copy of this dessertation and theses book are also included. The CD requires: (a) PDF file format reader, adobe acrobat is recommended, for the electronic version of the publicaitons; (b) microsoft power point for the presentaitons; and (c) web browser, e.g. microsoft explorer, to access its other contents.

Chapter 1   Optical Computing: An Introduction

The speed of computers was achieved by miniaturizing electronic components to a very small micron-size scale, but they are limited not only by the speed of electrons and elctro-magnetic effects in matter (Einstein's principle that signals cannot propagate faster than the speed of light) but also by the increasing density of interconnections necessary to link the electronic gates on microchips. It is unfortunate that the very large scale integration (VLSI) technology is approaching its fundamental limits in the sub-micron miniaturization process. It is now possible to fit up to 300 million transistors on a single silicon chip. It is also estimated that the number of transistor switches that can be put onto a chip doubles every 18 months. Further miniaturization of lithography introduces several problems such as dielectric breakdown and heat carrier.

All of these factors combine to seriously degrade device reliability. Even if developing technology succeeded in temporarily overcoming these physical problems, we will continue to face them as long as increasing demands for higher integration continues. Therefore, a dramatic solution to the problem is needed, and unless we gear our thoughts toward a totally different pathway, we will not be able to further improve our computer performance for the future.

The *optical computer* (OC) comes as a solution of miniaturization problem. It uses *photons* traveling on optical interconnections or thin films instead of electrons. Optical computing is seen by a number of researchers as: "electronics is the science of the twentieth century, and optics is the science of the twenty-first". Optical interconnections and optical integrated circuits provide a way out of VLSI limitations to computational speed and complexity inherent in conventional electronics.

Optical computers are producing vivid improvements in both *speed* and *quality* of information processing, especially in image processing tasks. A number of optical

approaches are being adopted, including *digital*, *analog* and *neural*. At SZTAKI[1], we are working with analog optical computers using optical Fourier transforms and *holography*. Our basic architecture is an all-optical modified *joint Fourier transform correlator* (JTC) with *templates* that represent correlation kernels for *feed-forward* operations. Using optical *feedback*, one can realize feedback matrix operations, as well. These architectures are dedicated to implement *cellular nonlinear/neural universal machine* (CNN-UM) [1] computations. Moreover, we have proposals to insert CNN chips into the optical array processors to perform *pre-*, *post-* and *intermediate-processing* tasks.

1.1. Advantages of Optical Computing

The semiconductor-based microelectronics developed so fast that other technologies were not able to keep pace with it [2]. The speed of processing and the integration density of switching elements are increased threefold in every two years, according to Moore's Law that apparently is coming to an end within a decade. The limits of the processing speed to the fast progress of microelectronic technologies can be summarized as follows:

Size limit: smallest feature size for lithography has a bottom limit; the number of electrons will be too small for "noise-free switching" in a small volume. However, single-electron transistors are recently created[2].

Heat dissipation at high density hinders further integration: more dense packing causes heat devastation; and

Interconnections are limited: planar layer, interconnects between layers - because of the lack of space (surface and volume is occupied by passive and/or active circuit elements).

---

[1] SZTAKI is the "Computer and Automation Research Institute of the Hungarian Academy of Sciences", Budapest ,Hungary

[2] the emerging nanoelectronic technologies are still in their infancy. They have a great number of problems to be solved before they will give practical solutions. Among other problems, interfacing of the macro-micro-maso-nano world seems to be serious question.

"Optical computers are producing vivid improvements in both speed and quality of information processing"

That is why new alternative technologies, solutions and principles are sought.

Thanks to the progress made in the technologies of key devices of optical information processing, optical computers are getting matured enough to help solving electronic computer's bottlenecks. New computing paradigms were formulated. The essentially parallel opto-electronic computer structures can serve (solve) these paradigms better. New physical and biological principles and materials satisfy the needs of computing. *Optical/opto-electronic*, *bio-*, *quantum* computing and their hybrid versions are foreseeable solutions.

Optical computing has several important and decisive advantages over existing and future electronic computing methods. It is able to implement the well developed electronic paradigms and principles, among them - in general - neural computing and - especially – CNN computing has a distinguished position. The main features of optical computing are:

High degree *of parallelism* enables to process and program flows (streams). A single instruction or command applies not to a *byte* or a *word* but to a whole frame (containing $10^6 – 10^7$ byte data). In one hand, simple optical architectures can perform in a *single step* 2D Fourier or other integral transformation on a frame. In two steps complex image (or matrix) operations, e.g. *correlations* for *pattern-recognition* and *-classification* can be executed. On the other hand, a photon-based processor using different *wavelengths* could generate *many-parallel* processes, drastically increasing computing speed and complexity.

High switching rate (frame rate); presently 1μs switching rate (1 MHz frame rate), in the near future 1ns (GHz frame rate) can be reached, physical limits suggest, that later even pico-second (TeraHz frame rate) will be achievable.

High overall processing speed: it is presently about 10 TeraFlops. In the near future 10^{15} bytes/s, later 10^{19} operations per second on bytes will be performed by optical processors (a consequence of high parallelism and high frame rate).

"The main features of optical computing are: high degree of parallelism; high switching rate; high processing speed; flexible interconnects; huge storage amount; flexible optical processing; and no cross-talk between photons"

Freedom and flexibility in interconnectivity that is able to realize free-space global interconnects, planar and mixed interconnects.

Optical storage of huge amount of data is possible with high density. Rapid access is possible and divers access schemes (analog/digital, bit-wise, image-wise, serial access of whole frames, random access or associative) have been elaborated. In diverse holographic forms storage density of 10^8 bit/cm^3 can be realized. This huge amount of total storage capacity seems to be reachable, with 10ns frame-access time (what is architecture dependent). A great store of holographic materials is being developed. Its parallel versions fit well to the parallel nature of OC.

Flexible optical processing: it is extremely versatile because it can be *analog*, *digital*, hybrid analog/digital (analogic) or hybrid optical/electronic (photonic), all possessing the advantages mentioned above. Optical processing is directly applicable for matrix operations.

No cross-talking because photons are uncharged and do not interact with one another (except the light intensity reaches the non-linear region of the material) as readily as electrons. Consequently, light beams may pass through one another in full-duplex operation, for example without distorting the information carried. In the case of electronics, loops usually generate noise voltage spikes whenever the electromagnetic fields through the loop changes. Further, high frequency or fast switching pulses causes interference in neighboring wires. Signals in adjacent fibers or in optical integrated channels or in free-space *do not* affect one another nor do they pick up noise due to loops.

1.2. Evolving Optics into Computing

One approach for evolving optics into computing is the optoelectronic approach [3]. It is to replace an existing subsystem with an optical subsystem having an almost exactly identical interface with the electronic subsystem it replaces. Replacing the magnetic disk by optical disk is a typical example. The advantage of this approach is that the user compares

only black box performance and is not required to adjust the system. The disadvantage is that the full capability of optics cannot be used in most cases due to loss of parallelism, *data transfer bottleneck*[3], that leads to low data rates.

An *all-optical* computer will take longer to gain acceptance, when it is realized, because it requires drastic changes to most of the present day technologies. This requires enormous efforts on wide front: *operating systems, systems software, application software, optical construction technologies* etc. An advantage is that whole computers would be optimized to be modular to permit limited application tailoring for higher performance.

Another strategy is to construct special purpose processors for specific applications. This has the advantage that the optical capabilities may be fully utilized. However, special purpose processors for attachment to larger computers are often limited by the capabilities of the interconnection with the computer.

Among these three general approaches, a dilemma of choosing from *all-optical* and *hybrid opto-electronic* architectures exists because all of the different stages of hybridization have more or less favorable features.

Table 1, provides a comparison between three familiar architectures.

"Opto-electronic; all-optical and special purpose optical processor are three strategies to involve optics into computing."

---

[3] The opto-electronic and electro-optic conversion can slow down the signal and because of their limited efficiency result in energy dissipation.

Table 1

Comparison between familiar optical computer Architectures

Architecture Type	Advantages	Disadvantages
All-optical	▪ high parallelism ▪ high speed ▪ no (or simple) interfacing is needed	▪ little or non local adaptivity ▪ hard to implement the additional filtering
Hybrid opto-electronic	▪ high flexibility in both local (near neighborhood) and global adaptivity (phase coding in the input plane and adaptive thresholding in the correlation plane)	▪ loss of parallelism (data transfer bottleneck) ▪ low data rates: slow processing
Hybrid CNN smart pixel (CNN chip with optical I/O)	▪ higher parallelism ▪ high flexibility in local (near neighborhood) adaptivity (phase coding in the input plane and adaptive thresholding in the correlation plane) ▪ no (or simple) interfacing needed	▪ some loss of parallelism ▪ some loss of data rate

1.3. Discussion and Conclusion

Optical computers are producing vivid improvements in both speed and quality of information processing. The main features of optical computing are: high degree of parallelism; high switching rate; high processing speed; flexible interconnects; huge storage amount; flexible optical processing; and no cross-talk between photons. *Opto-electronic*; *all-optical* and *special purpose optical processor* are three strategies to involve optics into computing.

From the reasons introduced in this chapter and from the considerations about the nature of correlation in addition to previous related work, [4], one concludes that the *mostly-optical* solutions are preferable to maintain maximal parallelism and avoid serial transfer and serial processing bottlenecks what would result in tremendous processing slow-down. The CNN-UM chip with pixel-wise optical I/O can adaptively and intelligently process correlograms if it is inserted in the correlation plane. This architecture is realizing a kind of smart-pixel array processing.

In conclusion, one can point out that optical computing provides the following advantages:

- High frequency channels leading to high bandwidth modulation, high data rate. Wavelength multiplexing is also possible.
- High parallelism is possible.
- Polarization effects can be utilized (polarization multiplexing)
- Dimensionality of optical sensing, transmission, processing is wide and variable.

It faces some bottlenecks and open problems like:

- Photoelectric and electro-optic conversion
- Coherent light (laser) is needed in many cases.
- Special sensor, modulator, deflector and combined devices and systems are required

One Step Forward

The next chapter is reviewing the Cellular Neural Networks (CNNs) and the CNN universal machine (CNN-UM) with respect to their theory, operation, implementations and applications. Special regards for their unique template operations toward optical implementation.

Chapter 2   Cellular Neural Network (CNN) and CNN Universal Machine (CNN-UM)

Cellular Neural Networks (CNNs) [5] are n-dimensional arrays of locally and regularly interconnected *neurons*, or *cells*. The global functionality of these cells are defined by a small number of parameters that specify the operation of the component cells as well as the connection weights between them. Many complex computational problems can be reformulated naturally as well-defined tasks where signal values are placed on a regular geometric 2-D and 3-D grid, whose interactions are limited within a finite local neighborhood (sometimes called the receptive field). CNN is the most appreciated candidate towards a visual microprocessor [6]. The complete CNN *notations, definitions and mathematical foundation* are detailed in [7].

An extension of the CNN paradigm is the CNN Universal Machine (CNN-UM), which is a programmable array computer complete with analog and logic registers and instruction storage. It is built around the CNN paradigm, and combines analog CNN operations with logic ones to yield a massively parallel *spatio-temporal* supercomputer, which can run stored "*analogic*" (analog-logic) programs. In recent years, a range of analog VLSI CNN-UM chips have been designed and implemented, e.g. [8]. These developments have opened up the possibility of applying CNNs to solve difficult real-world problems, but at the same time have posed new problems that arise as a result of imperfections in chip manufacturing technology.

In addition to general purpose CNN implementation, CNN-UM, there were several trials to implement special purpose electronic CNNs. Utilizing piecewise-constant resistors and a capacitor one contribution succeeded to implement an architecture for image segmentation and edge detection [9] while another [10] is built for Stereo Vision. Some other interesting digital emulation had been practiced over Field Programmable Gate Arrays (FPGAs) [11]

as well as for image processing [12]. While a new concept of nested CNN had been reported with implementation of oriented coding [13]. Still, the CNN-UM is the most reliable implementation with its vast programing capabilities.

For applications, CNN had covered several challenging areas including time sequence data mining [14,15]; acoustic source localization [16]; solving PDE on a high speed [17]; as well as 2-D linear low pass filters [18]. In addition to the traditional standard image processing functions, CNN provides: 50,000 fps (frame per second) image capturing and classification; and on-line video-flow processing (Bubble-debris separation; echocardiogram analysis; and Multi-modal image fusion). Moreover, if a problem can be modeled in the CNN terms, its solution would be faster, robust and more efficient than the traditional ways.

The implementation of Optical CNN or as it become well-know by *Programmable Optical Array/Analogic Computer* (*POAC*) was started in year 2000 and reported in [19] based on modified Joint Fourier Transform Correlator (JTC) and Bacteriorhodopsin (BR) as a holographic optical memory. Full parallelism, large array size and the speed of light are three promises offered by POAC to implement an optical CNN. They had been investigated during the last three years with their practical limitations and considerations leading to the first portable POAC version, Chapter 4.

2.1. The CNN Paradigm

A Cellular Neural/Nonlinear Network (CNN) is defined by two mathematical constructs [20]:

1. A spatially discrete collection of continuous nonlinear dynamical systems called *cells*, where information can be encrypted into each cell via three independent variables called *input*, *threshold*, and *initial state*; and

2. A *coupling law* relating one or more relevant variables of each cell to all neighboring cells located within a prescribed sphere of influence $S_{r(ij)}$ of radius r centered at *ij*.
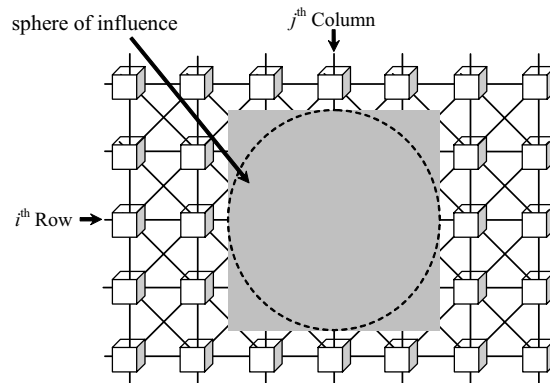
Figure 1

A 2-dimensional CNN defined on a square grid. The ij-th cell of the array is colored by black. Cells within the

sphere of influence, doted-line, of neighborhood radius r = 1 (the nearest neighbors) are colored by gray.

Figure 1 shows a 2D rectangular CNN composed of cells that are connected to their nearest neighbors. Due to its symmetry, regular structure and simplicity this type of arrangement (a rectangular grid) is primarily considered in all implementations.

2.2. CNN Core Cell and the Inter-Cell Interactions

The CNN paradigm does not specify the properties of a cell. As a basic framework, let us consider a two dimensional (*M×N*) CNN array in which the cell dynamics are described by the following nonlinear ordinary differential equation with linear and nonlinear terms (the extension to three dimensions is straightforward allowing similar interlayer interactions). There are three different cell models:

- *First order cell model*: This is the standard first order CNN cell. This model is used in the CNN-UM chip made in Berkeley [21].
- *Second order cell model*: This is a second order CNN cell, which is the same as the previous except for an additional capacitance connected across the output of the cell.
- *Full range first order cell model*: A first order cell, the state and output are the same and the voltage swing of the state transient is limited within [-1,1]. This model is used in the CNN-UM chip made in Seville [22].

"There are three different cell models: First order cell model; Second order cell model; and Full range first order cell model"

2.2.1. State equation of a single layer CNN with first order cell model

The standard first order CNN array dynamics is described by the following equations:

$$C\dot{x}_{ij}(t) = -\frac{1}{R}x_{ij}(t) + z_{ij}$$ — cell dynamics

$$\sum_{kl \in S_r(ij)} A_{ij;kl} \cdot y_{kl}(t) + \sum_{kl \in S_r(ij)} B_{ij;kl} \cdot u_{kl}(t) +$$

$$\sum_{kl \in S_r(ij)} C_{ij;kl} \cdot x_{kl}(t) + \sum_{kl \in S_r(ij)} D_{ij;kl}(\Delta v) +$$ linear cell interactions

$$\sum_{kl \in S_r(ij)} \hat{A}\left(y_{ij}(t), y_{kl}(t)\right) + \sum_{kl \in S_r(ij)} \hat{B}\left(u_{ij}(t), u_{kl}(t)\right) +$$

$$\sum_{kl \in S_r(ij)} \hat{C}\left(x_{ij}(t), x_{kl}(t)\right) + \sum_{kl \in S_r(ij)} \hat{D}(\Delta_1 v) \cdot \Delta_2 v$$ nonlinear cell interactions

(1)

$$y_{ij}(t) = f(x_{ij}(t)) = \begin{cases} 1 & if \ x_{ij}(t) \geq 1 \\ x_{ij}(t) & if \ -1 \leq x_{ij}(t) \leq 1 \\ -1 & if \ x_{ij}(t) \leq -1 \end{cases}$$ output equation

(2)

where

- x_{ij}, y_{ij}, u_{ij} are the *state*, the *output*, and the *input* voltage of the specified CNN cell, respectively, The state and output vary in time, the input is static (time independent), *ij* refers to a grid point associated with a cell on the 2D grid, and $kl \in S_r$ is a grid point in the neighborhood within the radius *r* of the cell *ij*.

- z_{ij} is the cell current (also referred to as *bias* or *threshold*) which could be space and time variant.

- Term $A_{ij,kl}$ represents the *linear feedback*, $B_{ij,kl}$ the *linear control* (*feedforward*), and $C_{ij;kl}$ the linear feedback from state. $D_{ij;kl}$ is a difference controlled linear template.

- $\hat{A}, \hat{B}, and \ \hat{C}$ are the nonlinear templates and $\hat{D}$ is a difference controlled nonlinear template.

- Terms *C* and *R* are the capacitance and resistance of the cells, respectively, default values are *C=1, R=1*.

- The Δv, $\Delta_1 v$, and $\Delta_2 v$ are the differences (or other arithmetic operations) of any two cell variables within S_r of two layers,
- Term $f(.)$ is the output nonlinearity, in most cases a unity gain sigmoid.
- The t is the continuous time variable.

The first part of Equation (1) is called *cell dynamics*; the following additive terms represent the synaptic linear and nonlinear interactions. Though the *threshold z_{ij}* may be *space-variant*, usually it is added to the template (space-invariant case). Equation (2) is the output equation. A CNN *base cell* corresponding to (Equation 1) is shown in Figure 2.
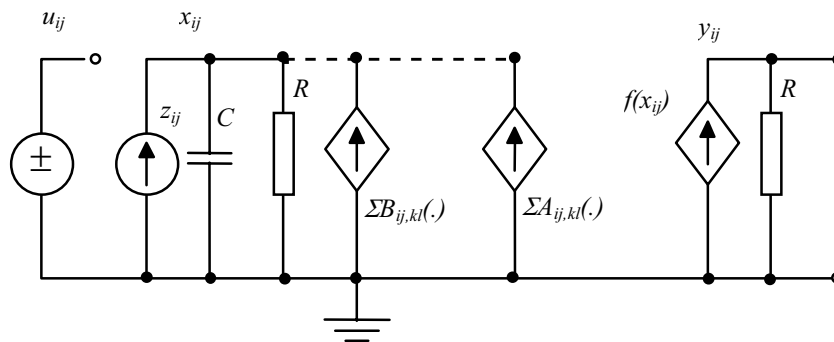
Figure 2

A CNN base cell corresponding to equations (1) and (2). The linear control and feedback terms are represented by voltage controlled current sources ($B_{ij,kl}$ and $A_{ij,kl}$).

The time constant of a CNN cell is determined by the linear capacitor (C) and the linear resistor (R) and it can be expressed as $\tau=RC$. A CNN *cloning template*, the program of the CNN array, is given with the linear and nonlinear terms completed by the cell current.

Equations (1) and (2) define a rather complex framework for computation. This is used only in nonlinear template design to make it clear what exactly the synaptic interactions are solving specific problems.

2.2.2. State equation of a multi-layer CNN with first order cell model

In a multi-layer CNN of L layers, the *p*-th layer dynamics is described by:

$$C_p \dot{x}_{p,ij}(t) = -\frac{1}{R_p} x_{p,ij}(t) + \sum_{q=1}^{L} ... \left\{ T1_p^q(ij;kl) \cdot y_{q;kl} + T2_p^q(ij;kl) \cdot u_{q;kl} + z_{p;ij} \right\} ... \tag{3}$$

where

- L is the number of layers of the network,

- T_q^p -s are linear or nonlinear templates,

- q: from layer. It specifies the layer from which the interactions are computed.

- p: on layer. It determines the actual layer on which the template values and current are.

The last term of Equation 3 denotes the multi-layer synaptic interactions. For sake of simplicity, not all individual template interactions are detailed. In some cases (e.g. biological modeling) it is useful to call the terms T_q^p the *signal transfer* terms since whether they represent a feedback or a control (feed-forward) depends on the whole network model. The default values are $C_p = 1$, $R_p = 1$.

2.2.3. State equation of a single layer CNN with second order cell model

If the cell is a second order type, the state equation is the same as equation (1), except the output equation has the following form (single layer case):

$$C_y \dot{y}_{ij}(t) = -\frac{1}{R_y} y_{ij} + f(x_{ij}(t)),$$

$$f(x_{ij}(t)) = \begin{cases} 1 & if\ x_{ij}(t) \geq 1 \\ x_{ij}(t) & if\ -1 \leq x_{ij}(t) \leq 1 \\ -1 & if\ x_{ij}(t) \leq -1 \end{cases} \tag{4}$$

where C_y is the capacitance connected across the output while R_y is the output resistance. The default values are $C_y = 1$, $R_y = 1$.

2.2.4. State equation of a single layer CNN with full range first order cell model

In case of full range cells the state equation is the same as in case of first order cell except that the cell dynamics and output equation are:

$$\dot{x}_{ij}(t) = -g(x_{ij}(t)) + z_{ij} \qquad \left.\right\} \text{ cell dynamics}$$

$$y_{ij}(t) = x_{ij}(t) \qquad \left.\right\} \text{ output equation}$$

$$(5)$$

where $g(x_{ij}(t))$ is defined as shown in Figure 3
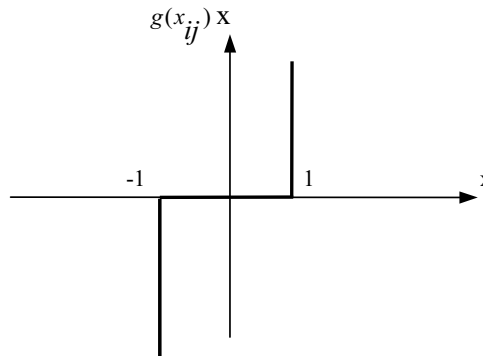
Figure 3

The function $g(x_{ij}(t))$ defined in the cell dynamics and its corresponding output function

One main interest of this work is to build an optical template library for optical CNN architectures. Hence, it is necessary to study the different classes of operations of the CNN.

2.3. Three CNN Classes of Operation

For linear operations, Equation 1 can be rewritten as follows:

$$\dot{x}_{ij} = -x_{ij} + A \otimes y + B \otimes u + z \qquad (6)$$

Where, x, A, B, u, y, and z are the present *state*, f*eedback template*, *feedforward template*, *input*, *output* and *threshold* respectively, while $\otimes$ denotes *spatial correlation*. Every CNN is uniquely defined by the three cloning templates {A, B, z} which consist of 19 *real numbers* for a 3×3 neighborhood (r=1). Since the real numbers are uncountable, there are infinitely many distinct CNN templates, of which the following subclasses are the simplest and hence mathematically tractable.

2.3.1. Zero-input (autonomous) class

A CNN belongs to the *zero-input* class if and only if all feedforward template elements are zero, i.e. B=0 as shown in Figure 4.
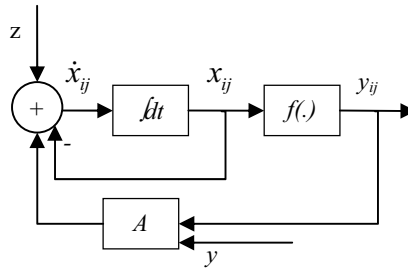
Figure 4

Zero-input (autonomous) CNN; in this case there are no input signals

Each cell of a *zero-input* CNN is described by:

$$\dot{x}_{ij} = -x_{ij} + A \otimes Y_{ij} + z \qquad (7)$$

"There are three CNN classes of Operations: (1) autonomous; (2) Uncoupled, and (3) Feedforward"

2.3.2. Uncoupled (scalar) class

A CNN belongs to the *uncoupled* class if and only if a_{ij}=0 except i=j, shown in Figure 5.
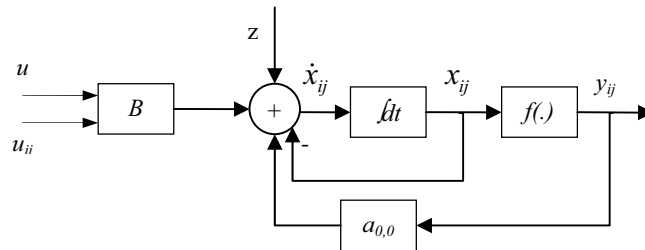
Figure 5

Uncoupled CNN; the data streams degenerate into simple streams indicating only a "scalar" self-feedback, but no coupling from the outputs of the surround cells

Each cell of an *uncoupled* CNN is described by:

$$\dot{x}_{ij} = -x_{ij} + a_{00} f(x_{ij}) + B \otimes U_{ij} + z \qquad (8)$$

2.3.3. Zero-feedback (feedforward) class: Optically implemented via Joint Fourier Transform Correlator (JTC)

A CNN belongs to the *zero-feedback* (*feedforward*) class if and only if all feedback template elements are zero, i.e. *A*=0 as shown in Figure 6.
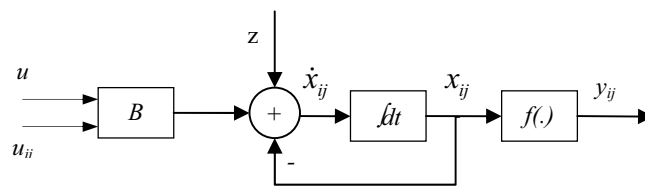
Figure 6

Zero-feedback (feedforward) CNN; there is no self-feedback, and no coupling from the outputs of the surround cells

Each cell of a *zero-feedback* CNN is described by:

$$\dot{x}_{ij} = -x_{ij} + B \otimes U_{ij} + z \tag{9}$$

This *continuous-time* CNN cannot be realized optically by utilizing the present t_2-JTC correlator, *see section 4.3.1*, but only in discrete-time. Hence, Equation 9 becomes:

$$x_{ij}(m+1) = \sum_k \sum_l b_{kl} u_{i+k,j+l}(m) + z \tag{10}$$

It is important to notice that there is no term representing the present state, *x(m)*, in the right hand side of Equation 10. The output function is expressed in Equation 11.

$$y_{ij}(m+1) = f(x_{ij}(m+1)) \tag{11}$$

where *f(.)* could be a nonlinear function representing gray scale or a *hard-limiter* nonlinear function between [0,1] or [-1,1] representing binary output. This work uses a nonlinear optical output function, *f₀(.)*, within [0,1] to represent gray scale and a hard limiter with 0 and 1 to represent binary output, Figure 7.
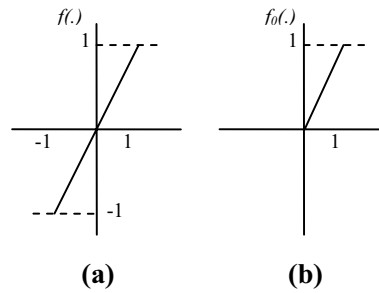
Figure 7

Output function of (a) CNN (b) Optical CNN; doted lines represent hard limiter

The block diagram of discrete-time CNN would be as in Figure 8.

"The bias term is a constant and thus can be combined with the threshold function"

z

u(m)

| b(m) | → (+) x(m+1) | f(.) | vᵢᵢ(m+1)

uᵢᵢ(m)
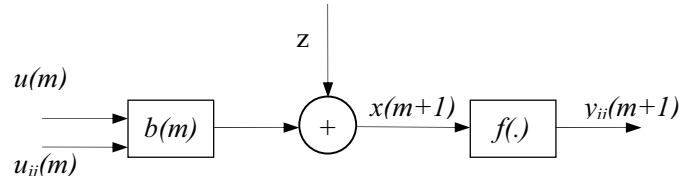
Figure 8

Zero-feedback (feedforward) discrete time CNN block diagram.

It is seen from Equations 9 and 11 that the CNN state equation is the sum of two cross correlations and a bias. The bias term is a constant and thus can be combined with the threshold function [23]. Accordingly, Equations 9 and 11 can be rewritten as:

$$x(m+1) = A \otimes y + B \otimes u \tag{12}$$

$$y_{ij}(m+1) = f_1(x_{ij}(m+1)) \tag{13}$$

where $f_1(.)$, the modified output function is:

$$f_1(x) = f(x+z) \tag{14}$$

In this case the block diagram of the discrete time CNN with a modified output function appears as shown in Figure 9.

u(m)

| b(m) | → (+) x(m+1) | f₁(.) | vᵢᵢ(m+1)

uᵢⱼ(m)
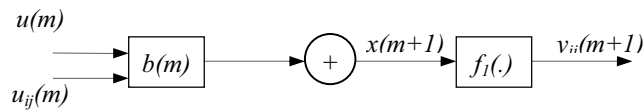
Figure 9

Zero-feedback (feedforward) discrete time CNN block diagram with modified output function that includes thresholding.

In the following chapters, the simulation and the optical implementation results of this section will be detailed.

2.4. The CNN Universal Machine and CNN Universal Chips

All early neural network chip realizations had a common problem: they implemented a single instruction only, thus the weight matrix was fixed when processing some input. Reprogramming (i.e. changing the weight matrix) was possible for some devices but took in order of magnitudes longer time than the computation itself.

This observation motivated the design of the CNN Universal Machine (CNN-UM) [1,24], a stored program nonlinear array computer. This new architecture is able to combine analog array operations with local logic efficiently. Since the reprogramming time is approximately equal to the settling time of a non-propagating analog operation it is capable of executing complex analogic algorithms. To ensure programmability, a global programming unit was added to the array, and to make it possible an efficient reuse of intermediate results, each computing cell was extended by *local memories*. In addition to local storage, every cell might be equipped with *local sensors* and additional circuitry to perform cell-wise analog and logical (analogic) operations. The architecture of the CNN-UM is shown in Figure 10.
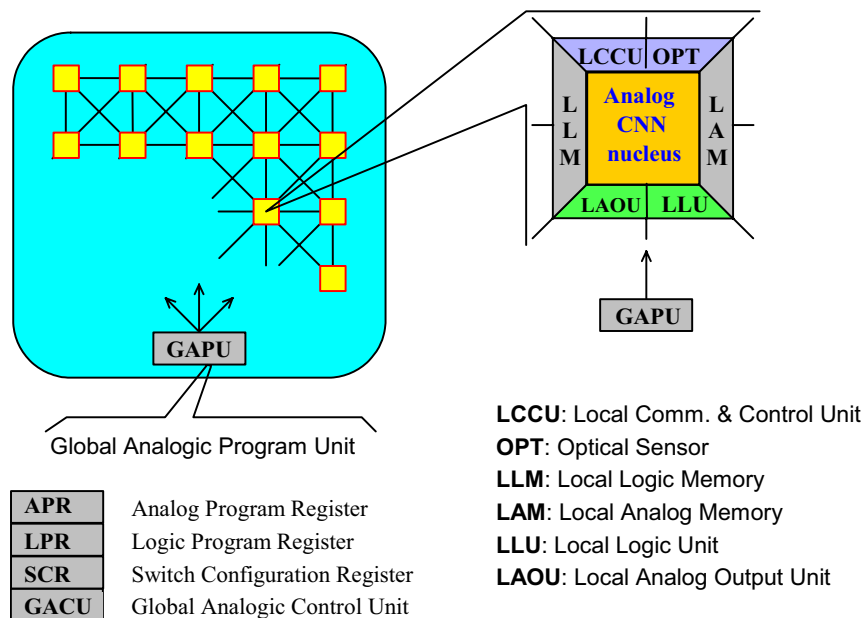
Figure 10

The architecture of the CNN Universal Machine, the analogic array supercomputer.

As illustrated in Figure 10, the CNN-UM is built around the dynamic computing core of a simple CNN. An image can be acquired through the sensory input (e.g. OPT: Optical Sensor). Local memories store analog (LAM: Local Analog Memory) and logic (LLM: Local Logical Memory) values in each cell. A Local Analog Output Unit (LAOU) and a Local Logic Unit (LLU) perform cell-wise analog and logic operations on the stored values. The output is always transferred to one of the local memories. The Local Communication and Control Unit (LCCU) provides for communication between the extended cell and the central programming unit of the machine, the Global Analogic Programming Unit (GAPU). The GAPU has four functional blocks. The Analog Program Register (APR) stores the analog program instructions, the CNN templates.

In case of linear templates, for a connectivity $r = 1$ a set of 19 real numbers have to be stored (this is even less for both linear and nonlinear templates assuming spatial symmetry and isotropy). All other units within the GAPU are logic registers containing the control codes for operating the cell array. The Local Program Register (LPR) contains control sequences for the individual cell's LLU, the Switch Configuration Register (SCR) stores the codes to initiate the different switch configurations when accessing the different functional units (e.g. whether to run a linear or nonlinear template). The Global Analogic Control Unit (GACU) stores the instruction sequence of the main (analogic) program. The GACU also controls the timing, sequence of instructions and data transfers on the chip and synchronizes the communication with any external controlling device.

Synthesizing an analogic algorithm running on the CNN-UM the designer should decompose the solution in a sequence of analogic operations. A limited number of intermediate results can be locally stored and combined. Some of these outputs can be used as a bias map (space variant current) or fixed-state map (space-variant mask) in the next operation adding spatial adaptivity to the algorithms without introducing complicated inter-cell couplings.

Analog operations are defined by either a linear or a nonlinear template. The output can be defined both in fixed and non-fixed state of the network (equilibrium and non-equilibrium computing) depending on the control of the transient length. It can be assumed

that elementary logical (Not, And, Or etc.) and arithmetical (Add, Sub) operations are implemented and can be used on the cell level between LLM and LAM locations, respectively. In addition data transfer and conversion can be performed between LAMs and LLMs.

Different CNN Universal Chips analog VLSIs had been implemented. The first fully working implementation that can run analogic algorithms is the 1995 mixed-signal version (it has an optical input) from Seville [25] (a revised version of the 1994 prototype that was only partially functional). This chip, embedded into the CNN Prototyping System [26], was used in various experiments validating some of the templates and algorithms. One of the most promising is developed in 1998 [27], it has a 64×64 CNN array and allows the use of fixed-state map techniques, global logical lines and ARAMs [28] during the algorithm synthesis. It is expected that this chip will be a good candidate in some industrial applications.

2.5. Discussion and Conclusion

This chapter introduced the CNN and the CNN-UM with their fundamentals, applications and concept of processing. The simplest three CNN classes: zero-input, uncoupled and zero-feedback are presented. The feedforward class was adopted for the optical implementation of the CNN. One reason is that the bias term, z, is constant and thus can be combined with the threshold function being suitable for optical realization.

One Step Forward

Towards the optical implementation of CNNs and CNN-UM, the following chapter is classifying the optical processing systems, in general, and the integrated optical systems, in specific. This yields how to classify the present programmable optical array computer (POAC) and hence understanding its bandwidth and limits for applications.

Chapter 3   Classification of Integrated Optical Processing Devices

"I established a new definition to classify integrated optical computing processors and devices based on system complexity, design and technology"

In the chapter 1, a general basic classification according to the hybridization type between optics and electronics had been presented. This chapter is introducing a classification of the present enormous varieties of the optical processing devices (OPDs) according to their level of *integration* and their *function*. The integration of *photodetecting* elements and processing circuits on the same chip, for obtaining better performance from sensors, or for making the sensing and processing system more compact, is not a new idea. What is relatively new is the concept of *smart sensing*, i.e. sensor information processing without redundant and unnecessary data acquisition, and with *at-sensor-level* processing. One popular detailed reference that collects most of the available integrated optical smart sensing devices is [29].

3.1. Classification of Integrated Optical Processors

According to the level of integration, one can classify the most recent integrated optical processors into main three categories:

1. Low level optical integration (LOI);
2. Medium level optical integration (MOI); and
3. High level optical integration (HOI)

Figure 1 shows such classification as well as some examples
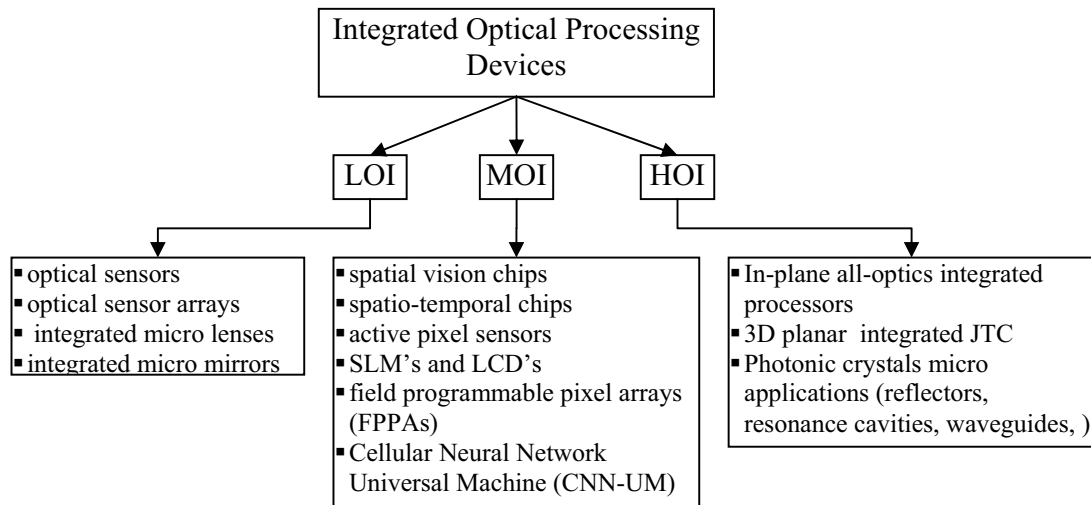
Figure 11

Classification of integrated optical processing devices

3.2. Low-Level Optical Integration (LOI)

The *LOI are* those systems include devices that can only *sense* or *project* optical data without any processing capabilities. That includes optical sensors; optical sensor arrays; integrated micro lenses, integrated micro mirrors, micro-opto-electro-mechanical systems (MOEMS), micro-displays, spatial light modulators (SLMs), laser arrays. They usually need electronics to manage the processing. These are hybrid opto-electronic systems with minor optical existence in the processing level. They are usually applied in optical sensing in digital cameras, optical switching and optical interconnects.

3.3. Medium-Level Optical Integration (MOI)

One more advanced level is the *MOI* where the *LOI* devices are integrated on one chip along with processing elements for both *special purpose* and *general purpose* processing. For special purposes, several vision chips had been reported in [29] for *spatial vision*, *spatio-temporal*, *active pixel sensors* (smart pixel arrays) etc. For *general purpose*, integrated systems with programmability are also developed. A recent one is the *field programmable pixel arrays* (FPPAs) [30,31]. It is mainly applicable for modeling purposes.

Another is the Cellular Neural Network Universal Machine (CNN-UM) which is developed for universal practical applications. The CNN-UM is presented in a separate chapter of this work.

The FPPA is also known as field-programmable smart pixel array (FP-SPA). An FP-SP is a smart pixel capable of having its electronic circuitry dynamically programmed in the field. Because of their functional flexibility, FPPA's can implement a wide range of optical interconnection architectures and functions, which is not possible with custom-designed special purpose smart-pixel arrays.

The flexibility of FPPA's, as with most other programmable devices, has some economic advantages. FPPA's can eliminate the need for the custom digital and VLSI design of an application-specific optoelectronic smart-pixel array, which is costly. FPPA's can also eliminate months of turnaround time associated with the fabrication of such a device. Currently, the design of a custom optoelectronic device can require six months. In contrast, the functionality of a FPPA device can be programmed dynamically in the field in a matter of minutes, typically by the downloading of a control bit pattern into the device. FPPA's can also be batch fabricated, leading to a significant cost reduction. In addition, they can also be made compatible with standardized I/O pitches, packaging assemblies, and optomechanical support structures.

Field-programmable smart pixels that integrate optical I/O onto complementary metal-oxide semiconductor (CMOS) substrates were proposed in 1994 [32] and have since generally been accepted as both feasible and practical. The most recent FPPA is developed and demonstrated in details by Sherif et. al. in [30]. One first application is programmed to implement an array of *free-space optical binary switches*, which can be used in an optical multistage network. In a second application the FP-SPA is programmed to implement an optoelectronic *transceiver* for a reconfigurable intelligent optical backplane, called the *hyperplane*. One can see that the future of hybrid optical processing will be built on the models running on these integrated chips.

"MOI: optical systems that LOI devices are integrated on one chip along with processing elements"

"FPPA and CNN-UM are general purpose MOI examples"

3.4. High-Level Optical Integration (HOI)

This category is still an innovative area for researchers as it involves *all-optical* processing tasks packed in an integrated small size. Microoptic integration is a key to many modern applications of optics and photonics. On example of such integrated systems is explained in [33] and shown in Figure 12.
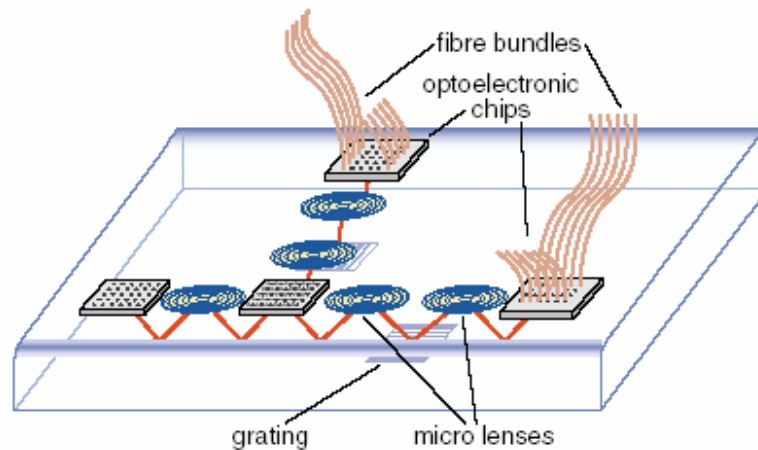
*Figure 12 (*J. Jahns et. al. 2002*)*

Schematic representation of an integrated free-space optical system. Light propagates inside a transparent substrate along zigzag path. The substrate thickness is between 1 and 10 mm. input/output devices are-surface mounted to the substrate using hybrid integration techniques

A specific example is the *planar integration* of JTCs [34]. Systems integration allows one to build compact optical correlator systems with significantly improved stability. Various different system designs for planar integrated VanderLugt correlators are now available. The most compact system consists of a single reflective diffractive optical element which combines the filter function and the Fourier transforming lenses. This element can be used on-axis with interlaced arrays of emitters and detectors [35]. In order to increase the space-bandwidth product, it is possible to build systems with an oblique optical axis. In this case the system can be folded into the typical planar optical configuration with the light propagating along a zigzag path inside the substrate.

3.5. Photonic Crystals: An Overview

In the last decade a new frontier has emerged with a goal to control the *optical* properties of materials [36]. The concept is to prepare materials that prohibit the propagation of light, or allow it in only certain directions at certain frequencies, or localize light in specified areas. Lasers, spectroscopy and optical computers are a few examples of the fields that will reap the benefits.

The *photonic crystal*, in an analogy to traditional crystal, has its periodic "potential" due to a lattice of macroscopic dielectric media instead of atoms. If the dielectric constants in the material are different enough, and the absorption of light by the material is minimal, then scattering at the interfaces can produce many of the same phenomena of *photons* (light modes) as the atomic potential does for *electrons*. One solution to the problem of optical control and manipulation is thus a *photonic crystal*, a low-loss periodic dielectric medium. In particular, one can design and construct photonic crystals with *photonic band gaps*, preventing light from propagating in certain directions with specified energy.

If, for some frequency range, a photonic crystal reflects light of *any* polarization incident at *any* angle, then the crystal has a *complete photonic band gap*. In such a crystal, no light modes can propagate if they have a frequency within that range. A simple dielectric mirror cannot have a *complete* band gap, because scattering occurs only along one axis. In order to create a material with a complete photonic band gap, one must arrange the contrasting dielectrics in a lattice that is periodic along three axes.

"Photonic crystals, 1-, 2- and 3-dimension, will play a crucial role in the all-optical integrated optical interconnects"
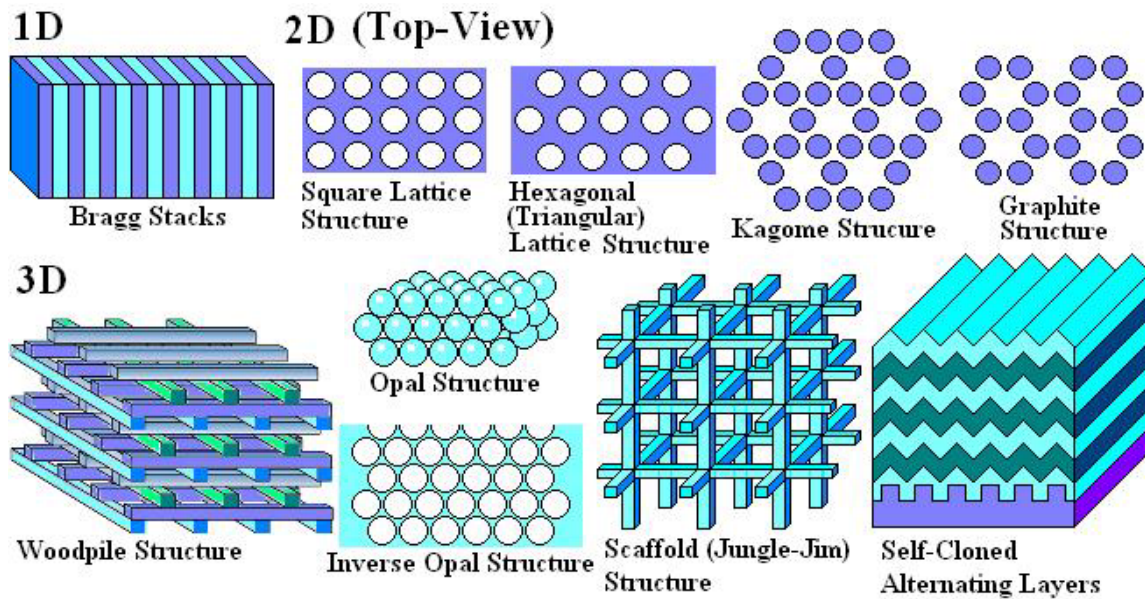
Figure 13

(Photonics Applications Lab - Dalhousie University)

Photonic crystal structures in 1-, 2-, and 3-dimensions

The study of photonic crystals is likewise governed by the Bloch-Floquet theorem, and intentionally introduced defects in the crystal (analogous to electronic dopants) give rise to localized electromagnetic states: linear waveguides and point-like cavities. The crystal can thus form a kind of perfect optical "*insulator*", which can confine light losslessly around sharp bends, in lower-index media, and within wavelength-scale cavities, among other novel possibilities for control of electromagnetic phenomena [37]. Figure 13 shows photonic crystals structures in 1- 2- and 3-dimentions.

The simplest possible photonic crystal, 1-D, consists of alternating layers of material with different dielectric constants. This arrangement is commonly used in dielectric mirrors and optical filters. The physical origin of the photonic band gaps is the difference in field energy locations. Off-axis propagation exhibits no band gaps in contrast to on-axis.

A two-dimensional photonic crystal is periodic along two of its axes and homogenous along the third. For certain values of the column (or square) spacing, this crystal can have photonic band gap in the *xy*-plane. Inside this gap no extended states are permitted and incident light is reflected. The symmetry of the crystal characterizes the electromagnetic

"Photonic crystals are applied into light reflectors, resonant cavities and wave guides"

modes. Defects in this structure localize modes in the plane, and that the faces of the crystal can support surface states.

The three-dimensional case is periodic along the three axes. It has *complete* photonic band gap. The discovery of particular dielectric structures which possess a complete photonic band gap is one of the most important achievements in this field. These crystals are sufficiently complex to allow localization of light at point defects and propagation along linear defects.

One recent publication [38] had introduced the use of holographic lithography to create three different types of photonic crystals with large 3D photonic band-gaps (PBGs). An optimized geometrical structure within the unit cell enables the lattice to exhibit PBG comparable to that of widely studies diamond lattices.

Photonic crystals are applied into light reflectors with specific wavelength range (usually narrow band) without appreciated absorption. A *resonant cavity* is one other application to trap a very narrow frequency of light. By using line defects, *photonic-crystals wave guides* become one more relevant application for monochromatic light.

Related to optical computing, Photonic crystals are one strong candidate to perform as pre-programmed optical channels and switches.

3.6. More General Classification of Optical Devices

Because integrated optics still in progress and not yet commercially available in a wide sense, one find more general classification for optical devices and functions. This includes *optical sensors* (OS), *optical transmission* (OT), *optical storage/memory* (OM), *optical processing* (OP) and *optical computing* (OC) *systems*.

The aspects of classification are:

- Analog/digital/hybrid, (analogical)
- Sampling and quantizing (spatially, in time, by polarization, by wavelength),
- Multiplexing for *sensing*, *transmission* (modulation/demodulation), *processing*, *combined* and *integrated* transmission and processing of information/data, *storage*, *retrieving* and *display*.

- ▪ Discrete channel/non-discrete channel.

- ▪ Dimensionality/multiplexing

- ▪ All-optical/hybrid opto-electronic or mostly optical

- ▪ Storage

- ▪ I/O devices, transmitting media (free-space, fiber, linear/nonlinear material, gradient index material), storage media, modulating, deflecting devices

From this classification, one can immediately predict the complexity of any optical system. Figure 14 shows a family-tree of optical system classification.
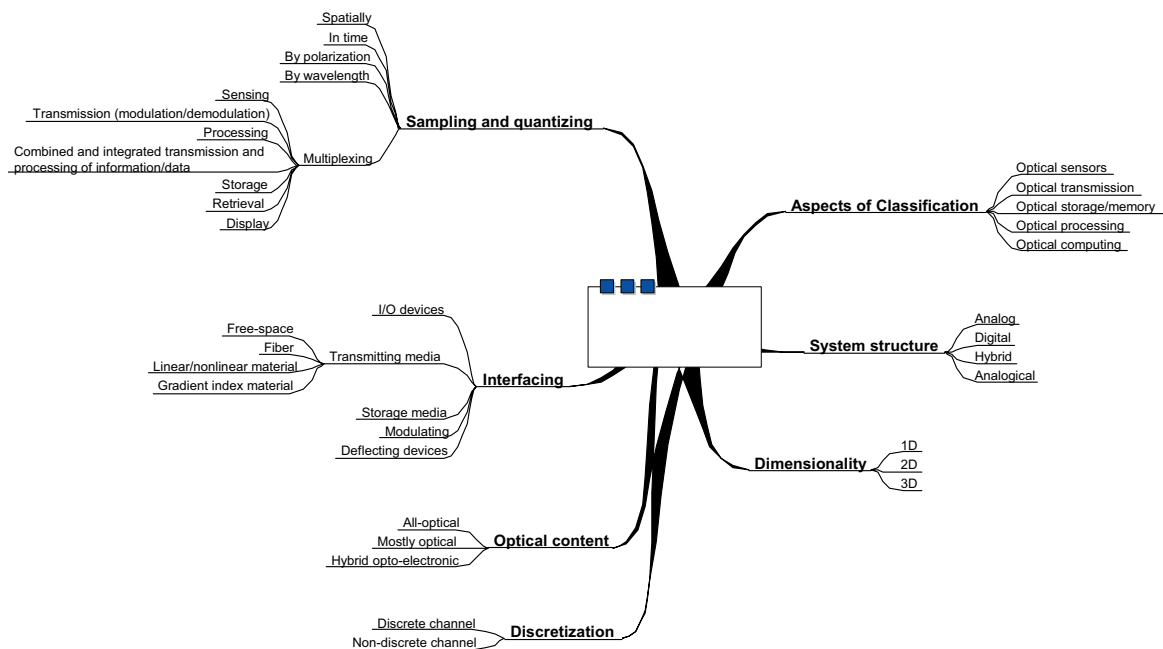
Figure 14

General optical system classification based on several optical aspects

3.6.1. Dimensionality

It was always a practical aspect to determine the dimensionality of the optical system as a figure of complexity. Table 2 summarizes the complexity of the optical system based on its dimensions (1-, 2- or 3-D). Moreover, the parameters of the spatial optical array that is being processed are listed. Hence, one can simply indicate the variables controls the system

complexity. The 0D spatial array dimension is defined here as only one beam, no pixelization exists. The array size is also one indicated factor. Furthermore, multiplexing of spatial (*x,y,z*), time (*t*), wavelength (*λ*) and polarization are also included. Spatial multiplexing can be in x-direction, x and y-direction or in the three xyz-direction. Polarization multiplexing (*p*) implies linear, circular and elliptic polarization.

Within this dimensionality classification, this work lays in the 3D processor's spatial dimension and the 2D spatial array dimension. No multiplexing factors had been utilized.

Table 2

Dimensionality of optical processors and their corresponding parameter: spatial parameters (xyz); time (t); wavelength (λ); and polarization (p)

Processor's spatial dimension	Spatial array dimension (spatial and its size in bracket)	Multiplexing (spatial, time, wavelength, polarization)
1D	0D: (0), t, λ, p, (0)	t, λ, p
2D (classical planar integrated optics)	1D: (x), t, λ, p, (<5000)	x, t, λ, p
3D	2D: (x, y), t, λ, p, (<5000×5000)	x, y, t, λ, p
3D	3D: (x, y, z), t, λ, p, (<5000×5000×100)	x, y, z, t, λ, p

3.7. Discussion and Conclusion

According to the level of integration one can classify integrated optical systems into: (1) Low level optical integration (LOI); (2) Medium level optical integration (MOI); and (3) High level optical integration (HOI). LOI is a system that can only sense or project optical data without any processing. MOI is the system that integrates the LOI devices along with processing elements on one chip, FPPA and CNN-UM are general purpose examples. HOI

includes all-optics integrated processing elements, e.g. planar integrated JTC. Photonic crystals, 1- , 2- and 3-dimension, will play a crucial role in the all-optical integrated optical interconnects being applied into *light reflectors*, *resonant cavities* and *wave guides*.

One more general classification was proposed and implies *optical sensors* (OS), *optical transmission* (OT), *optical storage/memory* (OM), *optical processing* (OP) and *optical computing* (OC) *systems*. It considers the dimensionality as the most important practical aspect of classifying the optical systems.

One Step Forward

The following chapter will demonstrate the Programmable Optical Array/Analogic Computer (POAC) through its evolution. It would be considered as one present realization in the frame of special purpose opto-electronic computer. Furthermore, it is a direct optical implementation to the CNN-UM.

Chapter 4   **Evolution of Programmable Optical Array/Analogic Computer (POAC)**

In the last three years there were distinguished advancements in designing and implementing the present optical setup of Cellular Neural Networks (CNNs), Programmable Optical Array Computer (POAC). In this chapter, its evolution is described along with its milestones. First, the Bacteriorhodopsin as an embedded holographic memory of the system is reviewed. Then, the previous several architecture versions of POAC are detailed and finally compared.

4.1. Bacteriorhodopsin as an Optical Holographic Memory

Bacteriorhodopsin (BR) is an integral membrane protein present in the purple membrane patches of the Halobacterium salinarium cells (formerly Halobacterium halobium). The function of BR is to convert solar energy into electrochemical energy (a transmembrane proton gradient) which is used by the cell to produce ATP from ADP. In order to capture photons, BR possesses a retinal chromophore linked to the lysine 216 through a protonated Schiff base. After the retinal is excited by a photon, a photocycle is initiated, in which both the retinal and the protein pass through a series of intermediate conformations, Figure 15.
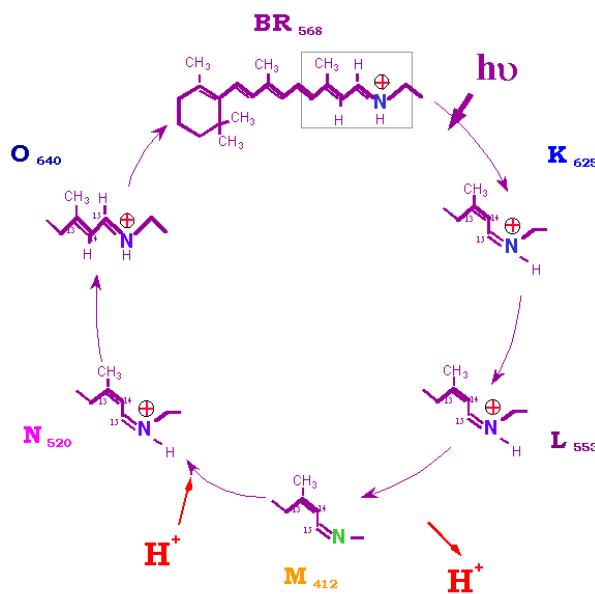
Figure 15

Bacteriorhodopsin complete photocycle

A simplified version of the photocycle, showing the different states of protonation of the Schiff base and, importantly, the absorption maxima. Thus, after getting a photon, BR changes its colour from purple to blue (625 nm in K intermediate), then to reddish - purple (520 nm in L), to yellow in M (412 nm), to blue in O (640 nm) and back to purple. The entire photocycle takes about 10 ms at ambient temperature, but there are several ways to slow down the photocycle (including the design of mutants) and, especially, it is possible to stop it in a given intermediate. An interesting property is that when an intermediate has been stabilized, illumination by light corresponding to its absorbance maximum drives it back to the purple form. Several studies of BR had been reported, e.g. [39, 40, 41], concerning the mechanism and parameters' control.

The different states of the photo cycle have different absorption spectra and it can be used to store information. Bacteriorhodopsin resolution is more than 5000 lines/mm, while the liquid crystal Optically Addressed Spatial Light Modulator (OASLM) is typically 60

"The different states of the photo cycle have different absorption spectra and it can be used to store information with a resolution more than 5000 lines/mm''

lines/mm. It worth to mention here that information processing capacity is growing with the square of the resolution.
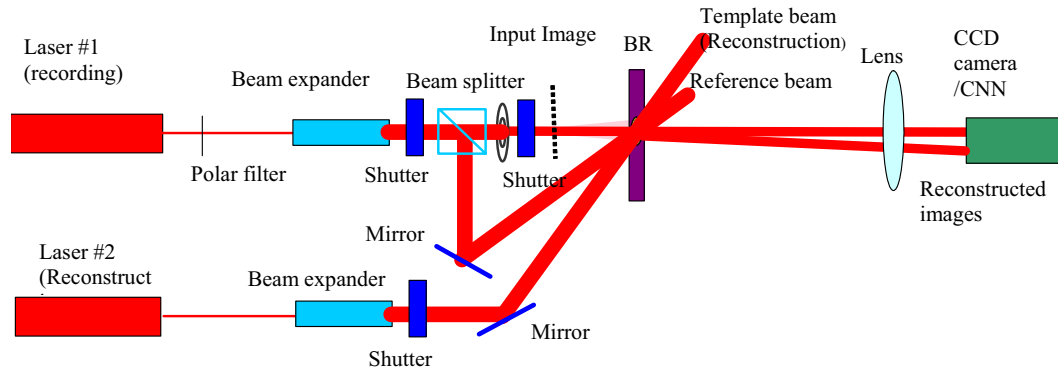
Figure 16

The scheme of the experimental setup that estimates the template size limitations of the BR based optical CNN implementation.

A part of the first experimental results utilizing the bacteriorhodopsine as a holographic memory for POAC [42] is shown in Figure 16 and Figure 17. Figure 16 represents the experimental setup that estimated the template size limitations of POAC BR memory. In Figure 17, an original image (a) had been recorded to the BR then reconstructed as in (b). The high similarity between both images proved that BR is a good candidate as a holographic memory for POAC.

(a)   *(b)*
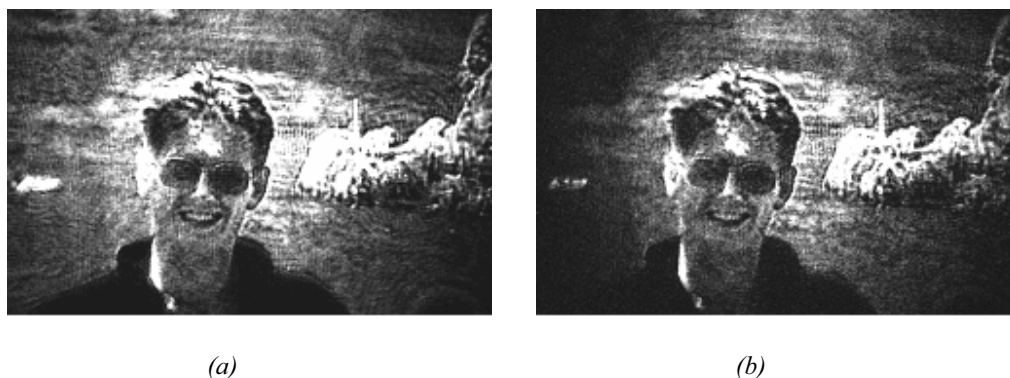
Figure 17

BR's holographic recording capabilities can be estimated from the comparison of the original gray scale test image (a) and the reconstructed image (b)

4.2. The Fundamental Optical Correlators

In 1964, Van der Lugt had introduced his correlator architecture, Figure 18, it is also known as *matched filter correlator*. Here, the input image, on the SLM, Fourier Transform (FT) is passed through a mask, LCD, containing the precalculated FT of a stored image (template). A *matched filter* based correlator is not suitable for real-time operation since a complex filter must be fabricated for each new input.

<div style="writing-mode: vertical">"The classical JTC suffer from that a new hologram pickup is needed for every new input or new template"</div>
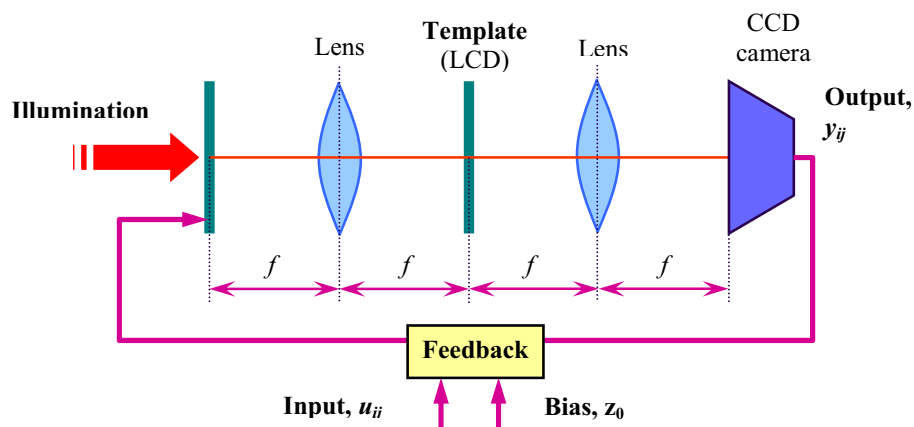
Figure 18

A 4f system with programmable templates, Van der Lugt 4f system

In 1966, Weaver and Goodman [43], and at nearly the same time Rau [44] had shown that the precalculation of the template FT can be avoided using their architecture, Joint Fourier Transform Correlator (JTC), Figure 19. It is inherently suitable for real time matching and tracking operations since no complex filter correlation is needed. A classical JTC suffer from that a new hologram pickup is needed for every new input or new template.
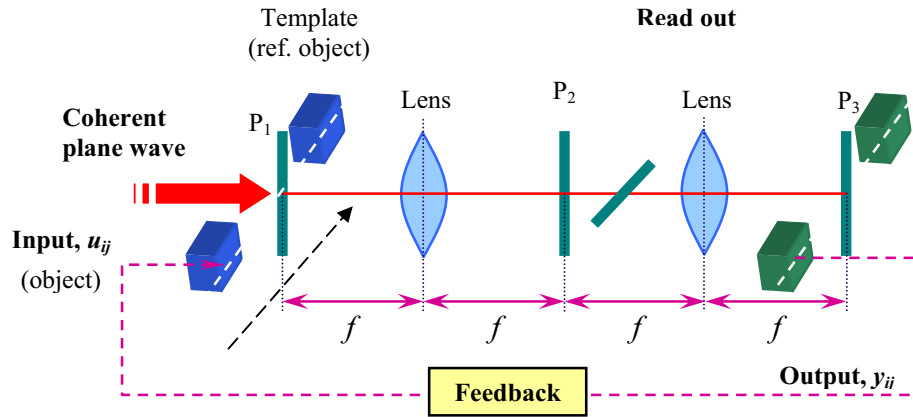
Figure 19

Systems based on Joint Fourier Transformation (Classical joint transform processor)

The following equations (15,16,17) mathematically describe the operations of JTC.

$$input = s(x + x_0, y) + t(x - x_0, y) \tag{15}$$

$$
\begin{aligned}
Joint\ Power\ Spectrum &= S^2(\alpha, \beta) + T^2(\alpha, \beta) \\
&+ S(\alpha, \beta)e^{-i\varphi_s(\alpha,\beta)}T(\alpha, \beta)e^{-i\varphi_t(\alpha,\beta)}e^{-i2x_0\alpha} \\
&+ S(\alpha, \beta)e^{i\varphi_s(\alpha,\beta)}T(\alpha, \beta)e^{-i\varphi_t(\alpha,\beta)}e^{i2x_0\alpha}
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
output &= s(x, y) * \bar{s}(x, y) + t(x, y) * \bar{t}(x, y) \\
&+ \bar{s}(x - 2x_0, y) * t(x, y) + s(x + 2x_0, y) * \bar{t}(x, y)
\end{aligned}
\tag{17}
$$

Where s and t corresponds to the input image and to the template, respectively and $S(\alpha,\beta)e^{i\varphi_s(\alpha,\beta)}$ and $T(\alpha,\beta)e^{i\varphi_t(\alpha,\beta)}$ corresponds to the Fourier transform of $s(x,y)$ and $t(x,y)$.

4.3. Chronological Milestones of the Programmable Optical Array/Analogic Computer (POAC)

4.3.1. Year 2000 : t_2-JTC

In 2000, the classical JTC was modified to enhance its correlation capabilities [45]. Figure 20 shows the t_2-JTC architecture where the hologram is dynamically fixed within the JTC scheme, while additional programming (by template #2) is done in the readout step. It is not needed to record a new hologram before new correlation of the input image is not changed, but with a great number of different templates one could correlate the same input image. So extremely high speed was achieved.
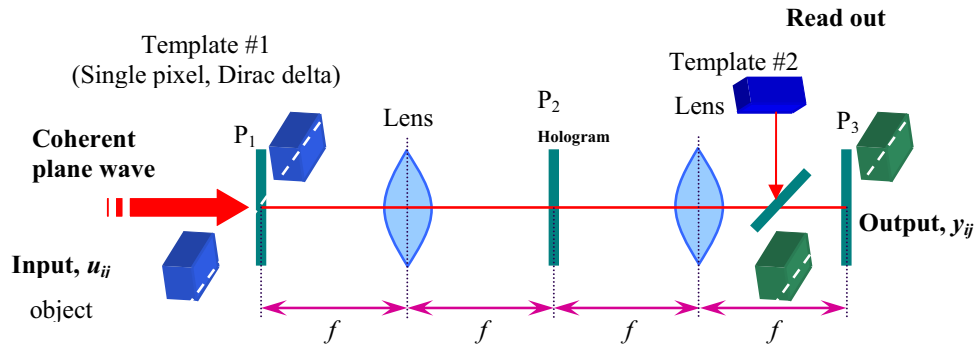
Figure 20

Scheme of the improved JTC architecture introducing the application of t_2 template

The operation of this architecture is described mathematically as follows, Equation (18):

$$
\begin{aligned}
output &= F^{-1}((F(s(x-x_0)+F(t_1(x+x_0))))^2 F(t_2(x)))) \\
&= F^{-1}(T_2(\alpha)(F^2(\alpha)+T_1^2(\alpha))) \\
&\quad + \overline{s}(x-2x_0)*t_1(x)*t_2(x) \\
&\quad + s(x+2x_0)*\overline{t_1}(x)*t_2(x)
\end{aligned}
\tag{18}
$$

where F and F^{-1} are the Fourier transform and its inverse.

"2000: the t_2-JTC was developed to enhance the correlation capabilities, with a great number of different templates that one could correlate the same input "

Main advantage of this optical implementation, that there is no harsh limitation on the size of the applied templates as in the case of VLSI implementations. So by the POAC computer it is easy to implement *multi-scale* image processing tasks by applying a set of scaled templates.

4.3.2. Year 2001: Breadboard Model of Semi-Incoherent Correlator

In 2001, a breadboard model of the semi-incoherent optical correlator architecture was built [46], Figure 21, and demonstrated that it is functional and superior to other optical correlator architectures in several aspects. By alternative scanning mechanisms (e.g. with a vertical cavity laser array or an array of micro-mirror devices) the speed of the reconstruction can be further enhanced and the size of the correlator can be dramatically decreased.
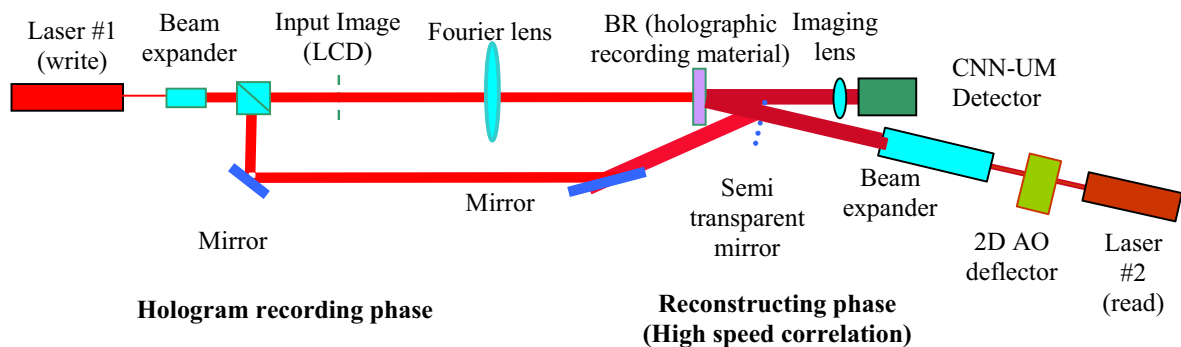
Figure 21

Semi-incoherent optical CNN implementation architecture

It is an extended version of the conventional JTC. This model device determines correlation in two consecutive steps. First, a hologram of the input image is recorded in bacteriorhodopsin (BR) film, which is a fast dynamic holographic material. In the following step (steps) the recorded hologram is reconstructed with specially adjusted parallel laser beams. Angular coding of the template pixels is accomplished by a 2D acousto-optical deflector (AOD) followed by a beam expander. This way, each beam

"2001: By alternative scanning mechanisms, the speed of the reconstruction can be further enhanced and the size of the JTC correlator can be dramatically decreased."

reconstructs an appropriately shifted version of the original image and forms the correlogram.

These beams are coherent but *mutually incoherent* due to the AOD causes different Doppler shifts and time difference. Within this architecture one can apply reference templates either in the hologram-recording step or in the readout-reconstructing step, or in both. Despite both the holographic recording and reconstruction are coherent, the shifted images are summed incoherently. It is especially beneficial, when different kinds of phase errors are present in the system (it is the case frequently in customary liquid crystal displays (LCD)). This way the majority of the otherwise always-existing coherent noise is eliminated, while errors of shadow casting are still not present in the system. In this setup reflection type holograms are recorded, but transparent holograms have also been tested.

4.3.3. Year 2002: Architecture Designs with Filtering

In 2002, the POAC development was accelerated and reported several new architectures and designs as well as starting to simulate the software (template) library for the hardware architectures, for both coherent and incoherent optical CNN implementations [47, 48]. The design shown in Figure 22 includes an optical spatial filter postprocessor as well. The fundamental t_2-JTC architecture is extended with an optical spatial filtering post-processor unit (confined in a dashed box). This design had enhanced the correlation peaks incoherent mode of correlator. However, in incoherent mode the filter acts only on the input image resulting in, e.g. correlation of edge enhanced inputs.
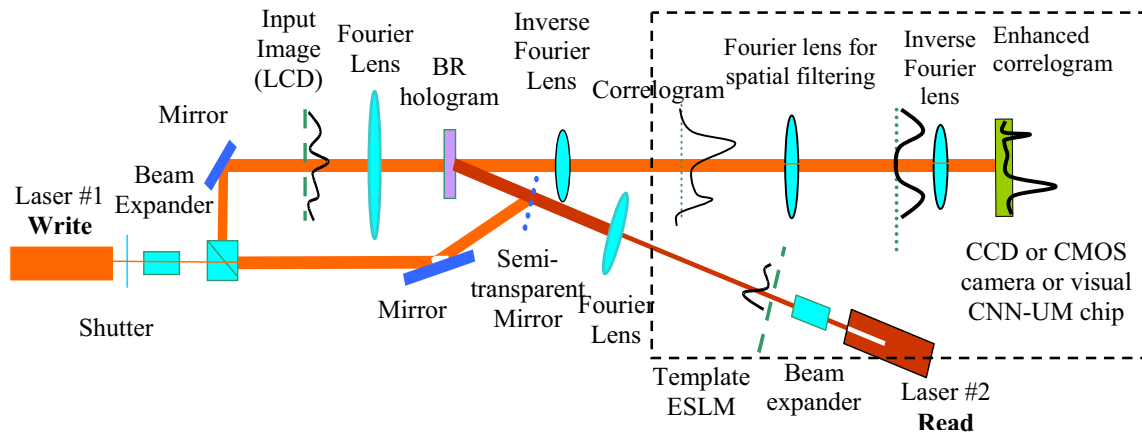
Figure 22

Architecture of our semi-incoherent optical CNN implementation

The simulation of the template library, will be discussed in details later, revealing that the architecture in Figure 22 should include a *phase-only* and also an *amplitude-only* modulator in the template arm of the architecture to implement complex valued templates. Complex valued pixels can introduce a π phase shift to implement negative template values. Figure 23 shows how two SLMs can be used to realize this.
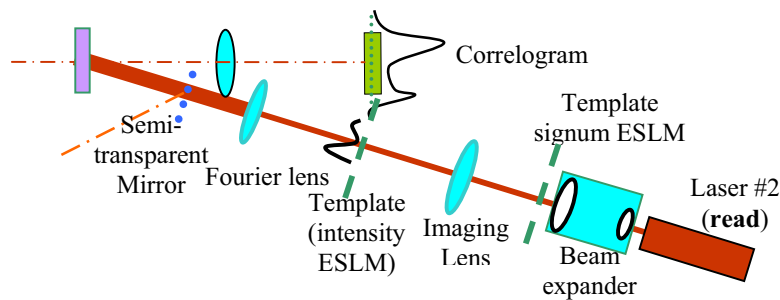
Figure 23

Architecture of our semi-incoherent optical CNN implementation with modified template arm utilizing two SLMs as phase-only and amplitude-only modulators

"2002: The design had enhanced the correlation peaks via an optical post-processor unit. phase-only and amplitude-only modulators in the template arm of the architecture to implement complex valued templates"

4.3.4. Year 2003: The Two-Wavelength POAC

In 2003, our point of view to POAC had been more obvious to demonstrate how it could meet real-time applications in addition to static images. Several untested designs had been tried to generalize our previous architectures and imply both coherent and incoherent modes of operations in real-time sense. By that, our vision becomes complete about how POAC should be implemented for *real-time* applications. Our recent published architecture [49], Figure 24, utilizes two wavelengths for both writing and reading processes.
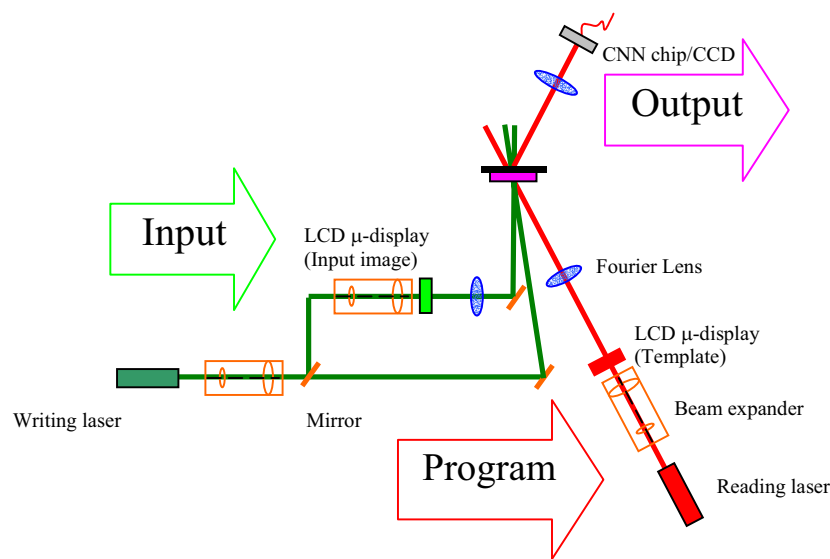
Figure 24

The recent two-wavelength POAC architecture

In this architecture, a bacteriorhodopsin-based two-wavelength optical correlator (CNN implementation) has been worked out and measured. It has several advantages: it is possible to write dynamic holograms of changing input images into BR (with green light) while a great number of different templates can be correlated on every one of the input images (with red light). Multi-instruction (template size: >100×100 pixel, 100 frames/s), multi-data (input image size: >1000×1000 pixel, 100 frames/s) optical computing can be implemented.

The red light's absorption is being much lower in intensity for the BR than that of the green light, the reading red light can only negligibly overwrite the input image's hologram.

Not withstanding, the refractive index modulation that carries the holographic information is even slightly higher for the applied read-out wavelengths, than for the recording ones. It is possible to use BR films of different time constants to optimize the lifetime of the input image's hologram, so we can adapt our correlator for rapidly changing input scenes and/or we can analyze a rarely changing input scene with a great number of different templates.

The grating system generated by the recording green laser light sets Bragg condition against the reading/reconstructing red laser beams direction. More exactly, the carrier interference fringes haves to be regarded (which are modulated by the information of the input image).

The governing equations are given as follows: Equation (19) is the grating constant d due to interference between the central object beam and the reference beam:

$$d = \frac{\lambda_g}{\sin \alpha'} \quad and \quad \sin \alpha' = \frac{\sin \alpha}{n} \tag{19}$$

where α and α' are the angles between the object beam and the reference beam *outside* and *inside* of the BR film accordingly; the refractive index of the BR film: *n=1.5*; the writing wavelength: λ_g = *533 nm*;

Equation (20) defines Bragg condition that should be satisfied by the reconstructing red light:

$$\sin \beta' = \frac{\lambda_r}{2 \cdot d} = \frac{\lambda_r}{2\lambda_g / \sin \alpha'} = \frac{\lambda_r}{2 \cdot \lambda_g} \cdot \sin \alpha';$$

$$\sin \beta = n \cdot \sin \beta' \tag{20}$$

where β' and β is th ered Bragg angle inside and outside of the BR accordingly; λ_r=*633nm*; So the following simple expression is defined:

$$\frac{\sin 2\beta}{\sin \alpha} = \frac{\lambda_r}{\lambda_g} \tag{21}$$

In this *two-wavelength* architecture one can spatially separate the recording and the read out process and apply them simultaneously, which improves the system performance considerably. The applied laser power can also be utilized more efficiently.

4.3.5. Year 2004: Laptop POAC

The fruit of the collaporative investment in POAC hardware realization appread in 2004 when the first compact size *laptop* version was reported to the community [50]. The portable programmable opto-electronic analogic CNN computer (Laptop-POAC) has been built. Its *kernel* processor is the novel type of high performance optical correlator based on the use of bacteriorhodopsin (BR) as a dynamic holographic material. This optical CNN implementation combines the optical computer's *high speed*, *high parallelism* and *large applicable template size* with the *flexible programmability* of the CNN devices. Unique feature of this optical array computer that the programming templates can be applied either by a 2D *acousto-optical* deflector (up to 32×32 pixel size templates) incoherently or by an LCD-SLM (up to 128×128 size templates) coherently. Input images are fed-in by a second LCD-SLM of 600×800 pixel resolution. The target applications of the laptop version are: *optical document security*; *adaptive target tracking* (single and multi object detection and tracking) and *collision avoidance*.

Both Bragg and *Lippmann-Denisyuk* type architectures have been built and successfully tested. However, because two different wavelengths for recording and reading the holograms are being used [49], *Bragg*-holograms are proven to be more appropriate, having less wavelength selectivity. Even in the case of *Bragg*-type, thick holograms it is important to choose a proper thickness of holographic medium. Two main aspects have to govern this choice. On one hand, diffraction efficiency grows with the increasing thickness. On the

other hand, thicker holograms have narrower angular (and color) selectivity that limits the angular size both of the input image and that of the templates, as well. The optimum thickness lies in the *Raman-Nath* region. In our case it is found to be about 40 μm. One of the main advantages of the two-wavelength correlator [49] is based on the recyclable light sensitivity of BR. Figure 25 shows the object (hologram recording) arm architecture of the laptop POAC system.
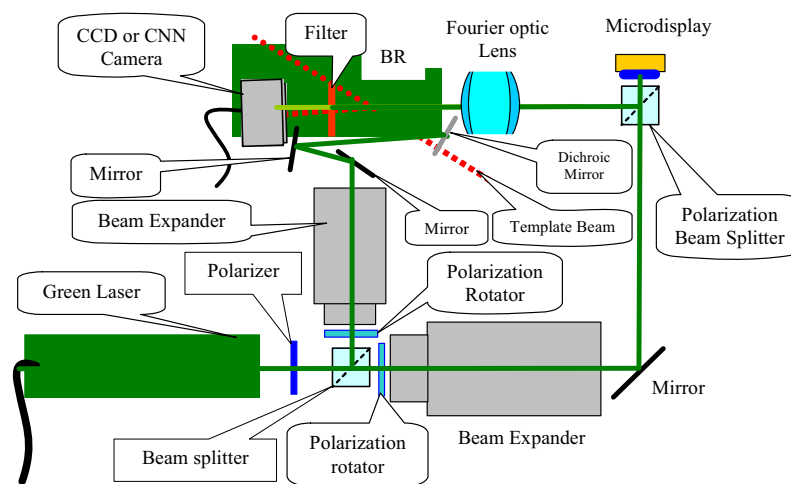
Figure 25

Object (hologram recording) arm architecture of the laptop POAC system. Appropriate beam attenuation, expansion and path lengths are set in both image and reference paths to assure the high quality dynamic hologram recording into the Bacteriorhodopsin film. Read out (template path) beams are denoted by dashed lines. Before the sensor (CNN VLSI chip or CCD camera) an appropriate filter cut the green laser of the object path.

"The laptop POAC provides coherent and/or Incoherent modes of operations as well as using Lippmann and Bragg holograms with two different wavelengths"

The earlier breadboard models, due to the size of the used general-purpose optical components, were rather large. By the use of several mirrors the optical system is folded to decrease its size. The newly designed small-size vary-focal lens has replaced the earlier used bulky commercial zoom lens. Thanks to the engineering work the system size has

been considerably decreased. By the application of different *dichroic mirrors* and filters the reference and template beams could be collected and a much shorter path-length has been reached. The photograph of the final architecture is shown in Figure 26.
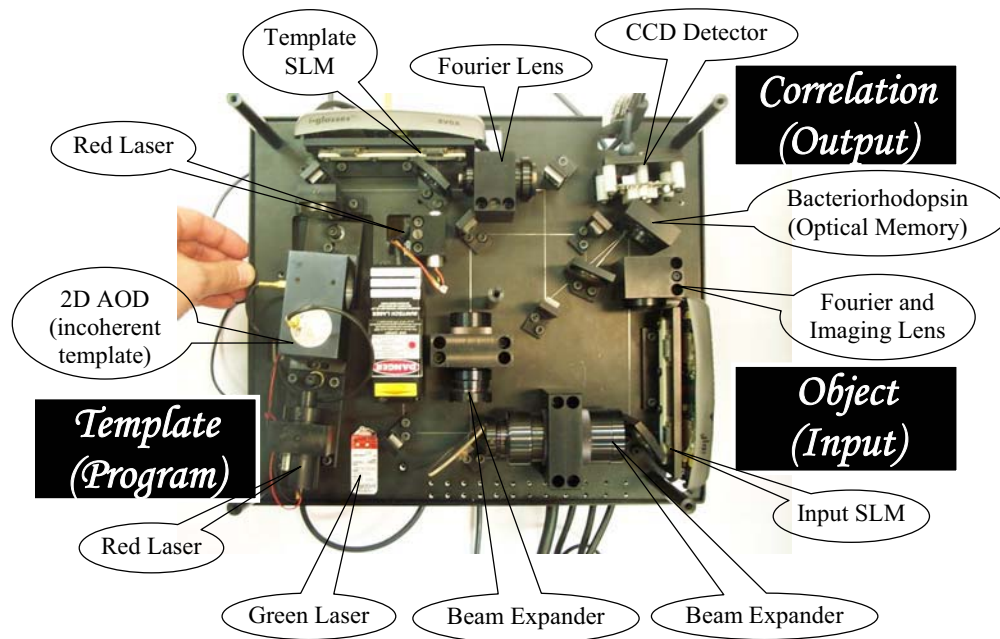
Figure 26

The recent two-wavelength; coherent; and incoherent laptop POAC

We have designed and built a *portable*, *laptop* size optical CNN implementation, which is based on a new type of optical correlator architecture and outperform all the so far published and implemented optical antecedents. This device incorporates both coherent and incoherent correlator solutions, to utilize the actual advantages of the used mode of operation. The strong and fix structure makes it easier to develop new applications and improve the system performance. This device is able to perform one Tera operation per second. Depending on the application, further simplification of the correlator structure is achievable, that can decrease the size and cost of the current prototype system. Incorporating a special CNN as a sensor and post and pre- processing unit will enhance the current CNN implementation performance considerably.

4.4. Discussion and Conclusion

In this chapter, a survey of the POAC evolution had been explored. The BR's different states of the photo cycle have different absorption spectra and it is used to store information with a resolution more than 5000 lines/mm. The classical JTC suffer from that a new hologram pickup is needed for every new input or new template. Thus, in 2000, the t_2-JTC was developed to enhance the correlation capabilities, with a great number of different templates that one could correlate the same input.

In 2001, by alternative scanning mechanisms, the speed of the reconstruction was enhanced further and the size of the JTC correlator had was dramatically decreased. During 2002, the design had enhanced the correlation peaks via an optical post-processor unit. *phase-only* and *amplitude-only* modulators in the template arm of the architecture to implement complex valued templates. Then in 2003, the two-wavelength architecture one could spatially separate the *recording* and the *read out* process and apply them simultaneously, which improved the system performance considerably.

Finally in 2004, the laptop POAC provided *coherent* and/or *incoherent* modes of operations as well as using *Lippmann* and *Bragg* holograms with two different wavelengths. The target applications of the laptop version are: *optical document security; adaptive target tracking* and *collision avoidance*. The parameters of the key elements used in the recent version of POAC are listed in appendix D.

One Step Forward

The next chapter is one major contribution of this work. It describes how the present architecture of the t_2-JTC can be utilized towards the realization of POAC. The aspects of such realization are contributed.

Chapter 5   Aspects of POAC
Implementation

"I proved that essential differences exist between coherent and incoherent correlation. I described and analyzed them to be utilized in optical CNN implementation"

Realizing the Programmable Optical Array/Analogic Computer (POAC), based on Joint Fourier Transform Correlator (JTC), confront primary three questions. First, which is more advantageous to implement POAC, utilizing *coherent* or *incoherent* JTC? Second, How to accomplish the effect of negative values in optics to implement the *templates*? Third, what is the most effective technique to realize the output function? Discussion towards the answers of these essential questions is presented here.

5.1. Understanding Correlation

The term c*orrelator* in JTC compelled its existence for a closer look at the nature of the correlation process for deeper understanding and to introduce a new view point to explicitly provide practical design considerations. First, the mathematical definition of *correlation*, (Equations 22 and 23) and *convolution* (Equation 24) are considered [51].

$$g(\beta) = t(\alpha) \otimes s(\alpha) = \int_{-\infty}^{\infty} t(\alpha - \beta) \cdot s^*(\alpha)\, d\alpha \tag{22}$$

$$g(\beta) = t(\alpha) \otimes s(\alpha) = \int_{-\infty}^{\infty} t(\alpha) \cdot s^*(\alpha - \beta)\, d\alpha \tag{23}$$

$$f(\alpha) = s(\alpha) \bullet t(\alpha) = \int_{-\infty}^{\infty} t(\alpha - \beta) \cdot s(\alpha)\, d\alpha \tag{24}$$

where α is a spatial variable; β is a spatial shift between the signal function *s* and the template *t*; the upper asterisk ***** denotes the complex conjugate; $\otimes$ and $\bullet$ denotes correlation and convolution, respectively.

Second, the present optical t_2-JTC implementation (for both coherent and incoherent processes) realizes *shifting* of the input (signal) image using Equation 23. Here, the reference template $t(\alpha)$ is correlated on signal function $s(\alpha)$ of real values, by shifting *s*. The two dimensional mathematical equation is as follows:

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(x'-x, y'-y)\, t(x', y')\, dx'\, dy' \qquad (25)$$

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(x', y')\, t(x+x', y+y')\, dx'\, dy' \qquad (26)$$

In these equations one obviously see that to perform correlation the signal is mirrored and shifted in Equation 25. While in Equation 26 the template is mirrored and shifted. In the present setup of POAC, described in Chapter 4, Equation 25 is utilized. This is because the image reconstruction is done by every pixel in the template arm. This reconstructs the original image for different reconstruction positions (shifting the reconstructed image) while the template remains stationary. Figure 27 describes this process.

"to perform optical correlation using the JTC, the template should be center-mirrored around its central element"
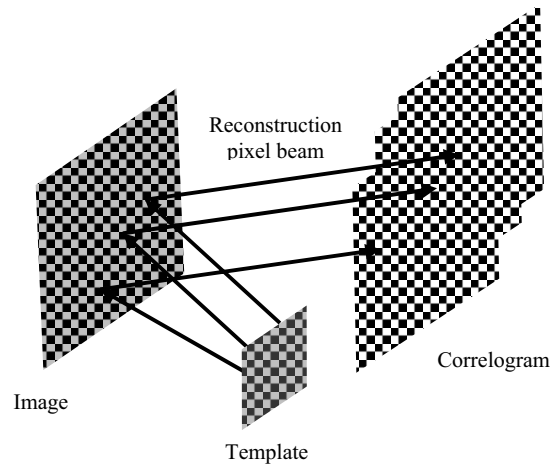
Figure 27

Shifting the image is performed by the different positions of the reconstructing template pixels. It requires to center-mirror the template to reach correlation and not convolution.

Third, because convolution is *commutative* and cross-correlation is not, $h \otimes s \neq s \otimes h$, the phrase "cross-correlation of *t* on *s*" or "*t* scans *s*" is defined.

Fourth, a conclusion is perceived and precautions when applying correlation process are as follows: Using a simple algorithm consisting of: *shifting s(α), multiplying it with t(α), or their mirrors, and summing the results*, Table 3 is obtained. Because the template size is always smaller than the image (signal) size mirroring the template is more economical. Our present POAC is performing correlation according to the combination shaded with black (Shifting the image with no mirroring and mirroring the template without shifting) in the table.

Table 3

*Correlation and convolution applied between two real functions **s** and **t***

			Signal (s)			
			Not Shifted		Shifted	
			Not Mirrored	Mirrored	Not Mirrored	Mirrored
Template (t)	*Not Shifted*	Not Mirrored	N/A	N/A	Convolution	Correlation
		Mirrored	N/A	N/A	**Correlation**	Convolution
	Shifted	Not Mirrored	Convolution	Correlation	Convolution	Correlation
		Mirrored	Correlation	Convolution	Correlation	Convolution

The simulation results presented in Figure 28 show that for an input signal (a) that is convolved over itself (self-convolution) resulting in the output shown in (c). When convolving (a) and its mirror (b); one can obviously see a discriminative peak on (d).

(a)   *(b)*

(c)   *(d)*
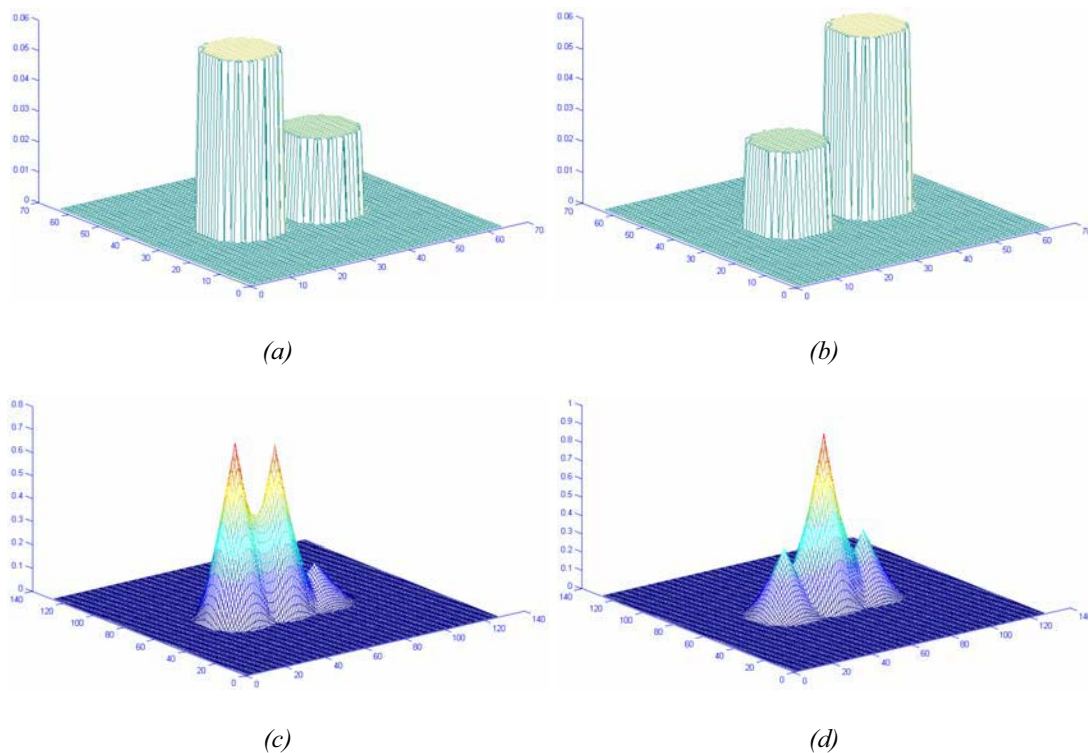
Figure 28

(a) Original signal; (b) mirrored signal; (c) convolution results asymmetric peak; (d) correlation with discriminative high symmetric and sharp peak

Finally, the result of this study is summarized and it becomes obvious that *mirroring*, or *flipping*, one of the two functions (the *image* or the *template*) is necessary to achieve *correlation* not *convolution*. In the two-dimensional case, *center-mirroring* (flipping vertically and horizontally around the center point of the image function) should take place. An experimental result to verify this understanding is shown in Figure 29.

(a)   *(b)*   *(c)*

(d)   *(e)*
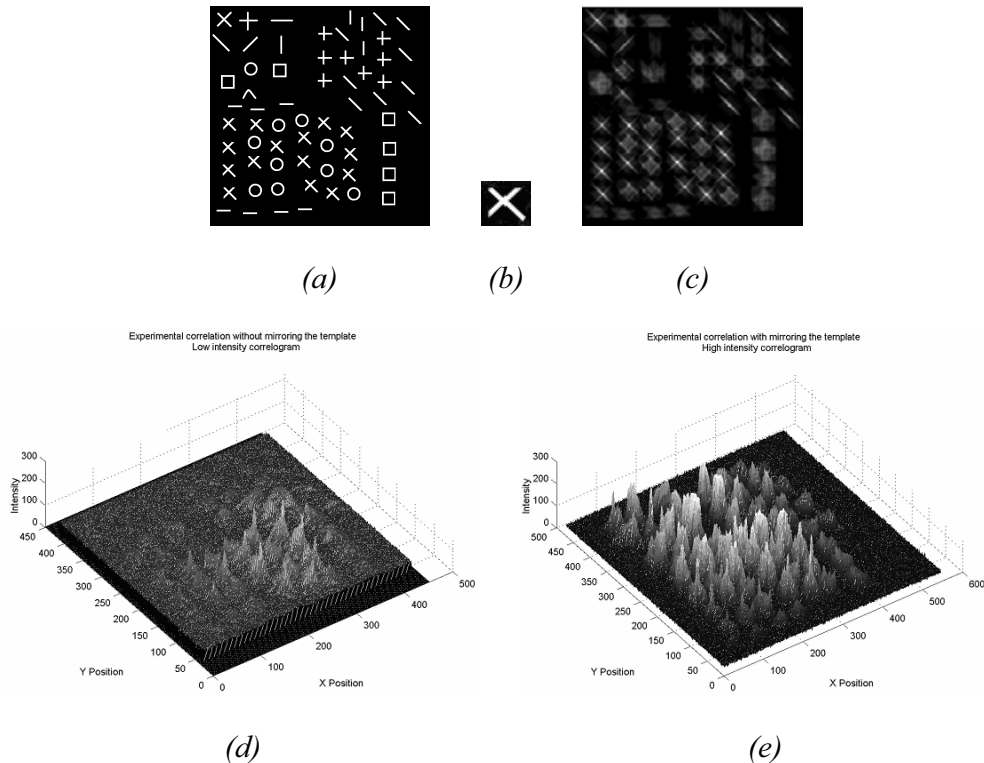
Figure 29

(a) Original image; (b) the template; (c) simulated noiseless correlation between (a) and (b); (d) poor correlation peaks intensity without mirroring the template (e) higher intensity peaks after mirroring the template

One practical explanation of these results is that: due to the optical devices (lenses, prisms, beam splitters, etc.) there are several mirroring effects of both image and template that cannot be traced easily. Using Table 3 as a guide while designing the optical correlator yields proper results. Another outcome of this study is to use the built-in convolution

function of MATLAB but mirroring one of the inputs leads to a correlation matrix with the same size as the signal matrix.

In conclusion, to perform optical correlation using the JTC, the *template* should be *center-mirrored* (it is smaller in size than the signal image function) around its central element. Otherwise, the performed operation would be *convolution*. In the optical setup the template is center-mirrored, and the image is being shifted to achieve correlation peaks at the correlogram plane.

5.2. Coherent versus Incoherent Correlation

The previous study provided the precautions to use correlation. This study answers the question whether one should use coherent or incoherent correlation for the implementation of JTC. For that reason, a simulation algorithm is developed and shown in Figure 30. It considers both coherent and incoherent correlations.

Practically speaking, optical *incoherent* correlation occurs when utilizing an Acousto Optical Deflector (AOD); a Spatial Light Modulator (SLM), or a Vertical Cavity Surface Emitting Laser (VCSEL) Array in a mode that one pixel at a time is to be illuminated. When the first pixel of the template is illuminated, a reconstruction of the image intensity is detected by a CCD camera. Then the next pixel is illuminated while the first is switched off. The previous reconstructed image intensity is overlapped by the current one corresponding to the present pixel. Then the photoelectric charges are integrated, and so forth. In this case correlation occurs on the intensity level of the overlapped reconstructed images.

On the other hand, *coherent* correlation utilizes SLMs to illuminate all of the pixels of the template *simultaneously* resulting in correlation of *complex amplitudes* that preserves the phase information not intensities where phase changes are lost. The correlated waves are *squared*, by the detector, to produce the intensity correlogram. This is why the algorithm in Figure 30 determines the square root of both the image and the template intensity values before performing coherent correlation.
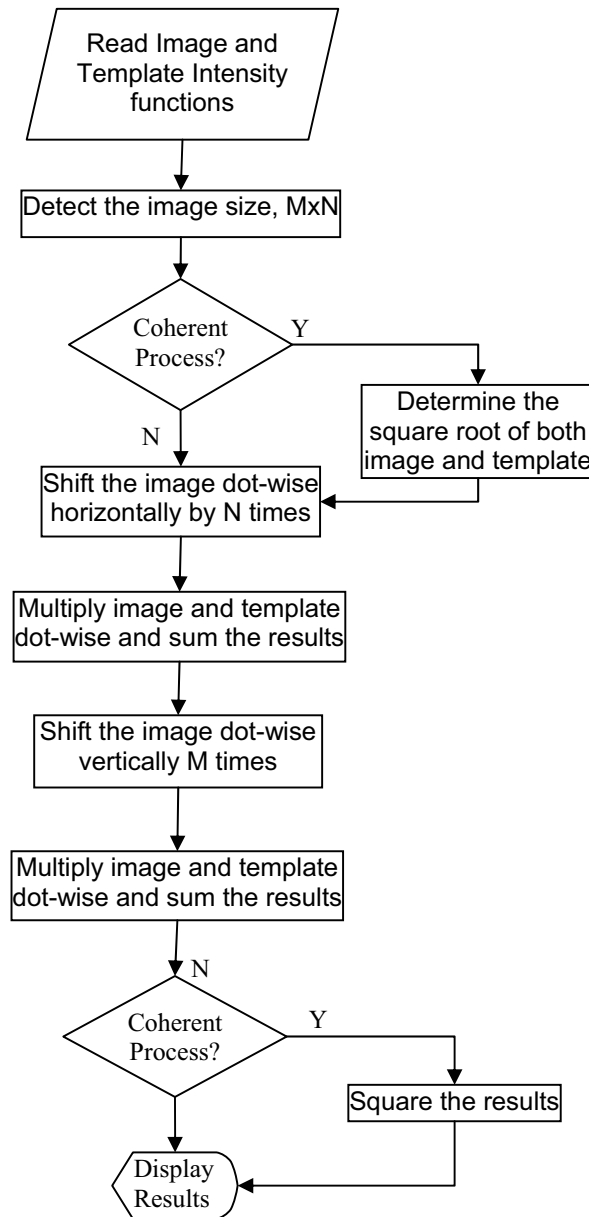
Figure 30

Optical coherent and incoherent correlation algorithm in two-dimension

One-dimensional example is simple and clear enough to show the desired difference between coherent and incoherent correlation. In Figure 31, two functions are correlated via the previous algorithm, noting that both functions are typically the same. Going through the

algorithm, one fundamental result is that the coherent correlation has a higher peak value than the incoherent one. This is due to the existence of phase contents in the coherent case, which means higher information content. Another result is that the contrast **R**: the ratio between the highest peak and its neighbors, is also higher in the coherent case independent of the image and the template functions, if no phase noise presents.

(a)   *(b)*

(c)   *(d)*
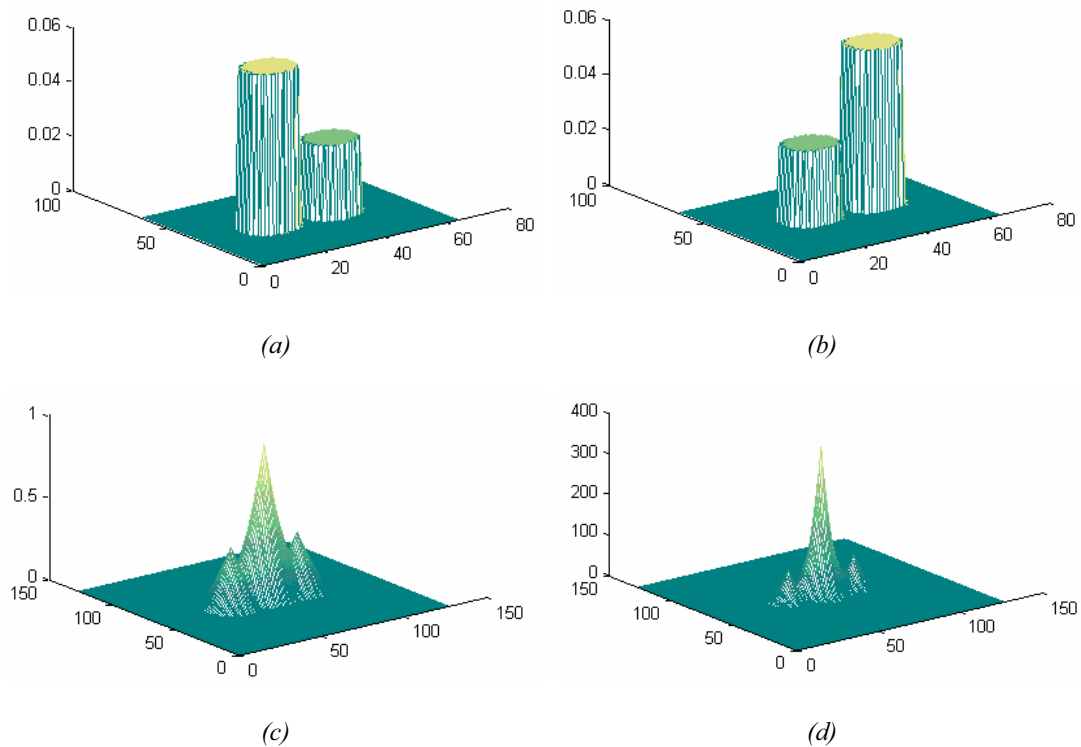
Figure 31

Simulation results for correlating a signal (a) with its mirror (b) incoherently (c) and coherently (d). The coherent correlation give a sharper and higher peak in (d) than the results of the incoherent one in (c) supposing phase noise free environment

When phase noise exists and applied to the input image and/or the template, the previous algorithm is modified to be as in Figure 32. The frequency of the noise is one parameter that dramatically affects the results.

"Coherent Correlation has higher peak and higher contrast than incoherent correlation for phase-noise free case"
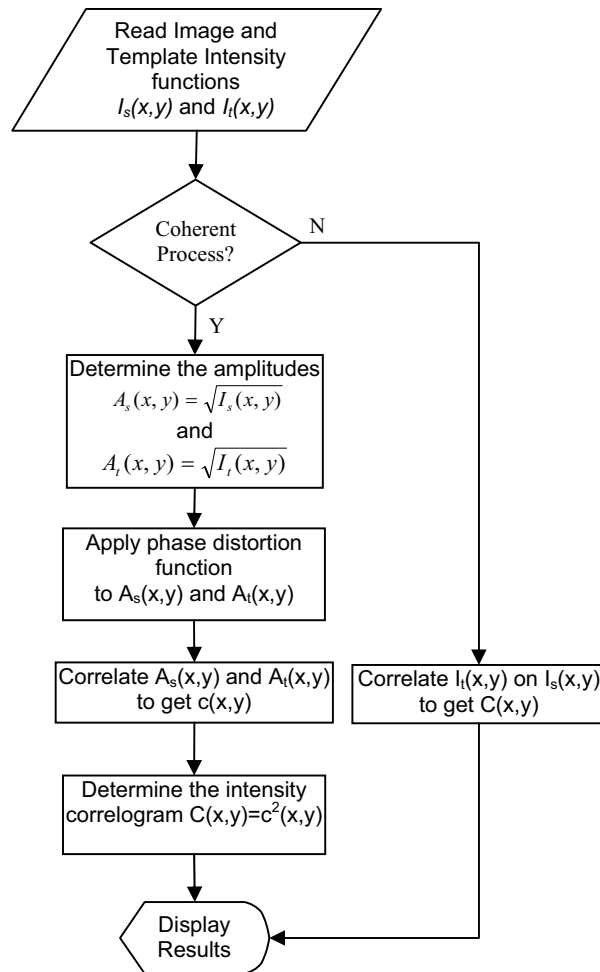
Figure 32

Coherent and incoherent correlation algorithm in the presence of phase-distortion.

The algorithm is applied to two typical one-dimensional functions to simply illustrate the concept. In this example, phase noise of value π is applied to the image function with a spatial frequency equals to the template size. As expected, the incoherent correlation was not affected at all. It remained constant and immune to the introduced phase noise. While the coherent correlation exhibited lower contrast. Figure 33 illustrates the concept of introducing phase noise into the system.
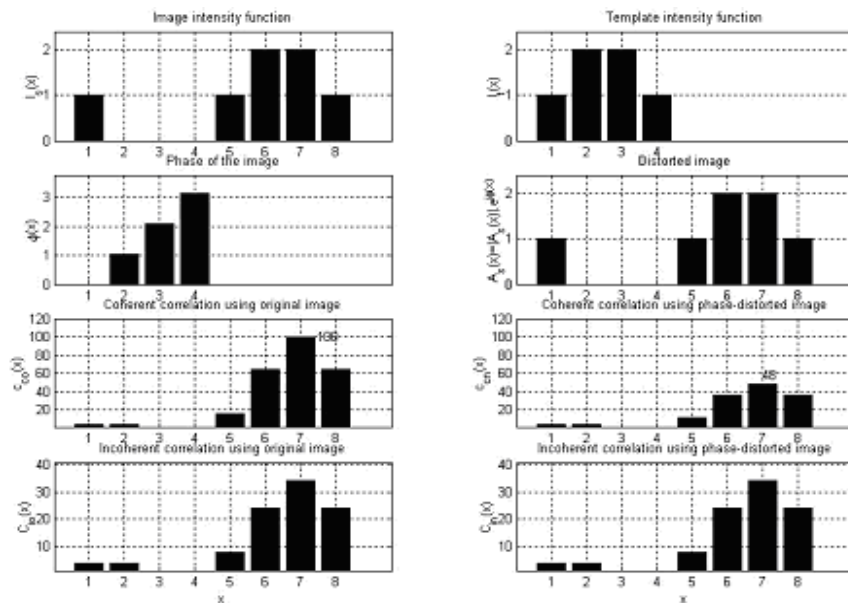
Figure 33

Optical simulation of coherent and incoherent one-dimension correlograms in the presence of phase-distortion of value range [0,π] within a period equivalent to the template size.

Here are some numerical values to explain the phenomena. In Figure 33, the incoherent correlogram, last row, does not seem to be affected at all by the phase change. The contrast remained constant at 1.42. On the other hand, the coherent correlogram highest peak intensity had dropped by about half of its original value. In addition, the contrast was dropped from 1.56 to 1.33 which is less than that of the incoherent correlation. Other simulation runs were performed for several phase values and spatial distribution periods are listed in Table 4.

"The higher the spatial period of the phase noise (low frequency), the higher the coherent correlation peak value and the higher the contrast."

Table 4

Examples of coherent correlation in the presence of phase noise distributed over different periods and spatial frequency

Noise period	Spatial period	Incoherent correlation highest peak value (arbitrary value)	Coherent correlation highest peak value (arbitrary value)	Incoherent correlation contrast	Coherent correlation contrast
0 to π	template size	100	64	1.42	1.33
-π to π	template size	100	4	1.42	1.00
0 to π	2× template size	100	88	1.42	1.50
-π to π	2× template size	100	59	1.42	1.40

In Table 4, one can obviously conclude that for a high spatial period of the phase noise (low frequency), the coherent correlation peak value becomes high as well as the contrast.

From the above studies I conclude the following:

1. The *template* should be center mirrored to achieve correlation not convolution.

2. The incoherent correlation is immune to phase noise but simple to realize and requires photoelectric accumulation to integrate the images needed for correlation. Moreover, the thresholding is hard to implement because the contrast is low

3. Coherent correlation has higher contrast than the incoherent one but sensitive to phase noise

4. The higher the spatial period of the phase noise (low frequency), the higher the coherent correlation peak value and the higher the contrast

5. Simulation results recommended that coherent correlation should be used if effective optical implementation of the CNN templates is required.

One other analytical prove is to consider each pixel in an SLM generates a complex amplitude vector ($A_i=A_i\,e^{-j\varphi_i}$). Where A_i is the absolute value and φ_i is the phase. In the incoherent case, the summation is performed on the square of the absolute value of the vectors representing only the intensities of each one, Equation 27.

$$I_{incoherent} = A_1^2 + A_2^2 + ... + A_i^2 \qquad\qquad (27)$$

In the coherent case, the vectors summation is multiplied by its complex conjugate as in Equation 28.

$$I_{coherent} = (A_1\,e^{-j\varphi_1} + A_2\,e^{-j\varphi_2} + ... + A_i\,e^{-j\varphi_i})(A_1\,e^{-j\varphi_1} + A_2\,e^{-j\varphi_2} + ... + A_i\,e^{-j\varphi_i})^* \qquad (28)$$

where (*) represents the complex conjugate. This vectors' summation is squared resulting in higher intensity than the incoherent case. (a) *(b)*

Figure 34 shows how the coherent correlation is performed

(a)   *(b)*
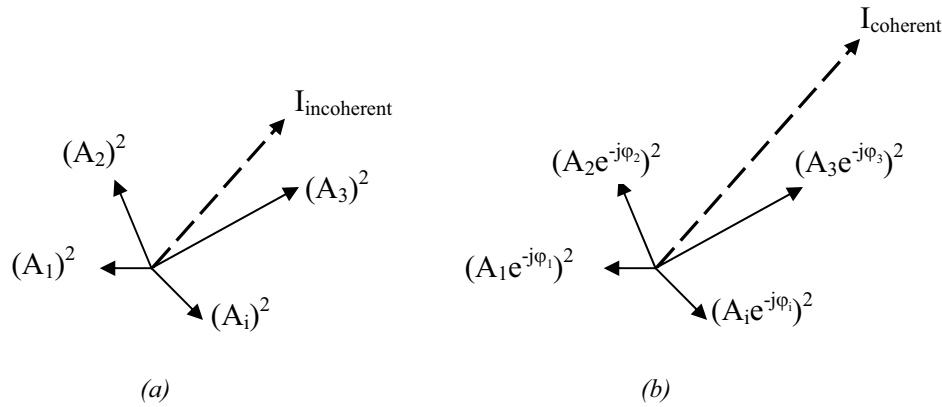
Figure 34

(a) Incoherent correlation includes only the absolute value of the complex amplitude. (b) Coherent correlation includes both amplitude and phase values corresponding to (i) number of pixels. Hence, The resultant peak intensity is much higher in the coherent mode than the incoherent one.

5.3. Optical Representation of Negative Template Elements

Because most of the CNN templates, that are considered to be implemented optically, consist of both *positive* and *negative* real-valued elements, it is essential to provide the same effect for the optical templates to achieve similar results.

Consider the example of Equation 29, where a complex number A is represented into amplitude $|A|$ and phase θ. The method is to apply a phase shift of value π to the real number, Equation 30, leading to *negating* it, if its original phase is considered to be zero. Within the template a reference phase for all of its elements will be considered zero among the elements. Hence, introducing a phase shift π to any of the template elements will result in negating this specific element, with respect to the neighboring elements.

$$A = |A| e^{j\theta} = A\cos(\theta) + iA\sin(\theta) \tag{29}$$

$$A = |A| e^{j(\theta+\pi)} = A(-1) + i(0) = -A, \text{ for } \theta = 0 \tag{30}$$

The complex valued nature of the optical waves and the availability of optical devices (SLM or ESLM) that provide phase modulation can yield the optical implementation of negative numbers. In the following sections, it will be shown how this is optically simulated within the optical CNN templates. Here, B represents the intensity function of the template. The phase $\phi = \pi$ is applied to one or more elements of $b = \sqrt{B}$, the amplitude wave function of the *B-template*, when negative values are required. In Figure 35, the utilized algorithm simulates both coherent and incoherent optical correlations.

"A phase shift π is applied to one or more elements of the template when negative values are required"
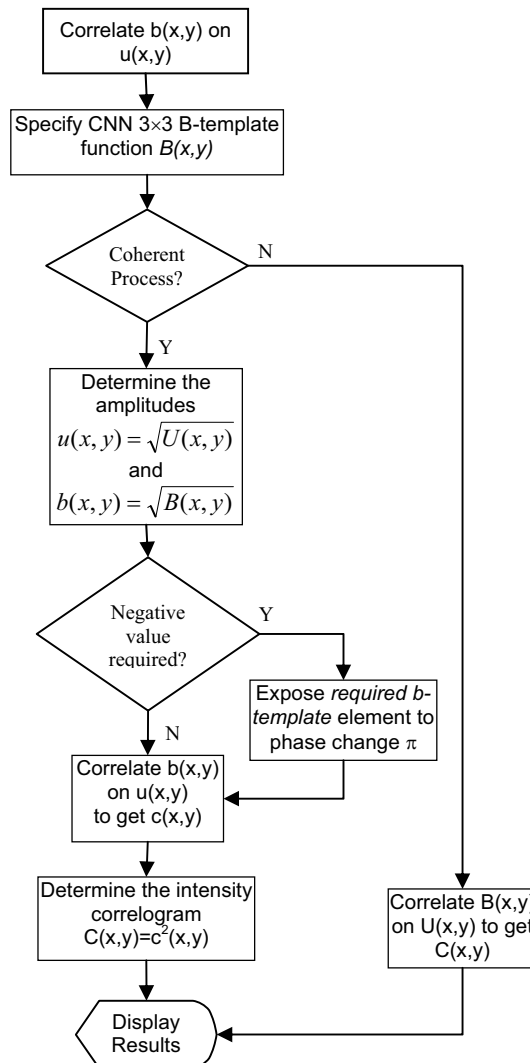
Figure 35

Optical Feedforward CNN B-template simulation using JTC

Note that in the coherent case the square root of the original intensity input for both the image and template is required.

From the above analysis and simulation algorithm I proved that the complex amplitude values of template elements can be optically implemented by applying a phase shift of value π resulting in inversing its arithmetic sign. I had confirmed these results via optical simulation algorithm, *e.g. see Figure 52*. Moreover, I described how to optically implement

the effect of the negative value, via *phase-only* and *amplitude-only* spatial light modulators (SLMs) placed in the template arm of the programmable optical array computer, *see optical setup in section 4.3.3 and Figure 23*.

5.4. The Output Function

At this step of understanding POAC in terms of JTC the correlogram is formed and is in need to apply an output function to obtain the expected results. The filtered correlogram contains peaks that are easy to threshold with a uniform threshold level. To acquire optical adaptive thresholding one needs a *smart pixel array processor* (SPAP) to detect the local maxima of the correlogram intensity distribution. The visual CNN-UM chip is one available candidate, but presently it has low resolution, 128×128, compared to what is typically required, ~1000×1000. This large difference in resolutions makes it impractical to utilize the CNN chip in such operation presently.

$$f(x) = \left(1 - x^2\right)\left(1 - e^{-x^2/2}\right)$$

(31)

The Fourier transform of the correlogram can help provide the segmentation of its frequency contents. It is proposed to eliminate the low frequencies which usually correspond to background noise using a coherent high pass filter, in addition to eliminating the very high frequencies, generated by the edges of the filter window. A *Mexican-hat-like* band-pass filter is used, Equation 31, to achieve this, as simulated in Figure 36. The Mexican-hat function is suitable for such filtering because of its Gaussian-like distribution that has no sharp edges which might generate high frequency noise.

"Mexican-hat-like band-pass filter eliminates the low frequencies which usually correspond to background noise and the very high frequencies corresponds to sharp edges"
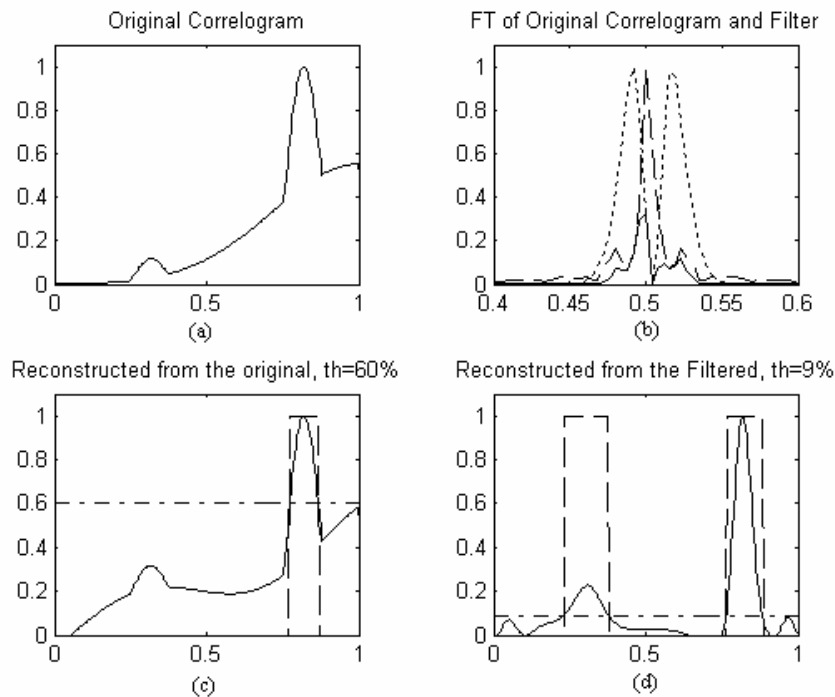
Figure 36

(a) correlogram of a non-uniform illuminated image; (b) FT of the correlogram (the dashed line) filtered with a Mexican hat filter (dotted line); (c) thresholding the original correlogram (one peak is lost); and (d) thresholding the filtered correlogram (both peaks are detected)

Another technique is to use a center surround template based on the above concept. The following example shows how did I proved it with a one-dimensional example followed by a two dimensional experimental results.

Figure 48 shows a model of a practical correlogram contains several kind of noise affects the peak intensity. A one-dimension longitudinal cross section of a practical correlogram might contains: (1) offset constant background noise; (2) A flat high intensity resulted from another high intensity object (not the target object); (3) Sharp and high intensity peak; (4) Low intensity peak superimposed on a background noise; (5) Superimposed peak on a linear background noise; (6) High intensity peak over a flat and high back ground noise; (7) Superimposed peak on a non symmetric background noise; and (8) Very low intensity peak. Note that the correct peak shape has always the same peaky

shape but might vary in its width. Only 6 correct peaks exist on this correlogram. The goal is to *post process* this correlogram to avoid adaptive thresholding and using *level thresholding* to extract the peaks.
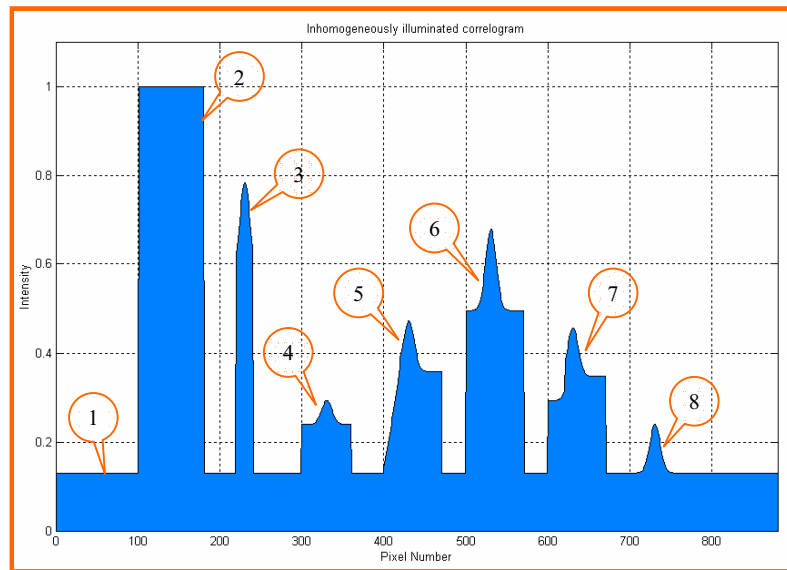
Figure 37

A one-dimension longitudinal cross section of a practical correlogram might contain several correlation peaks varying in size and width as well as high intensity background. The problem is to extract only the 6 correct peaks (3-8 in this example) located on this correlogram

A Mexican hat filter, Figure 49, can be used as a band-pass filter to extract the correlation peaks only. It works as a center-surround template based on the fact that the correlation peak is always symmetric and sharper than any other higher intensities located on the correlogram. I use it here as an intensity-invariant template. This way, only the correlogram peaks will be extracted and no other, even higher in intensity. I proved via simulation that the positive area of this filter must be at maximum one-third of the total function integral (i.e. $A = \frac{1}{3}(A + 2B)$) or $A{=}B$). This critical area is defined by three parameters: (1) positive area width parameter (a) that defines the width if the positive area

at amplitude of zero (y=0); (2) the depth parameter (b) it is a linear function of (a) that defines how deep should be the negative area to satisfy the critical area condition; and (3) the amplitude (c) which in a normalized situation equals to 1 (c=a)
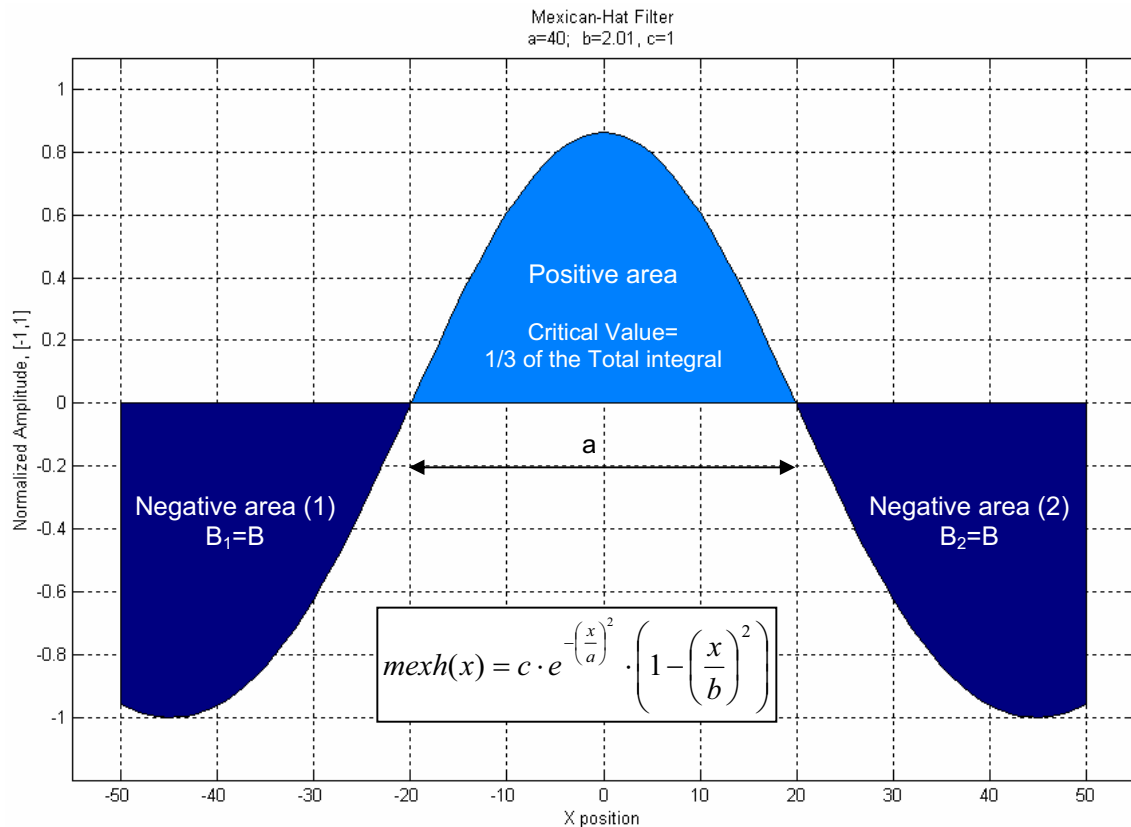
Figure 38

Mexican-hat filter is used to eliminate the unnecessary background intensities and extract the correct correlation peaks

The quadratic interpolation represents the relation between the Mexican-hat parameters (a) and (b) to have the critical positive area necessary for post processing the correlogram for peak(s) detection task, Figure 39.a. At one operating point with specific values of Mexican hat parameters (a) and (b), the template values are determined by sampling. In this case shown here, the operating point is at a=40 and b=1/2.01.

Sampling 4 points results in half the diagonal values for a 7×7 center surround template. In this case the diagonal is {-0.9598, -0.7636, 0.2597, 0.8619, 0.2597, -0.7636, -0.9598}. The resultant template, for this case is:

$$T = \begin{bmatrix} -0.9598 & -0.9598 & -0.9598 & -0.9598 & -0.9598 & -0.9598 & -0.9598 \\ -0.9598 & -0.7636 & -0.7636 & -0.7636 & -0.7636 & -0.7636 & -0.9598 \\ -0.9598 & -0.7636 & 0.2597 & 0.2597 & 0.2597 & -0.7636 & -0.9598 \\ -0.9598 & -0.7636 & 0.2597 & 0.8619 & 0.2597 & -0.7636 & -0.9598 \\ -0.9598 & -0.7636 & 0.2597 & 0.2597 & 0.2597 & -0.7636 & -0.9598 \\ -0.9598 & -0.7636 & -0.7636 & -0.7636 & -0.7636 & -0.7636 & -0.9598 \\ -0.9598 & -0.9598 & -0.9598 & -0.9598 & -0.9598 & -0.9598 & -0.9598 \end{bmatrix}$$

(a)   (b)
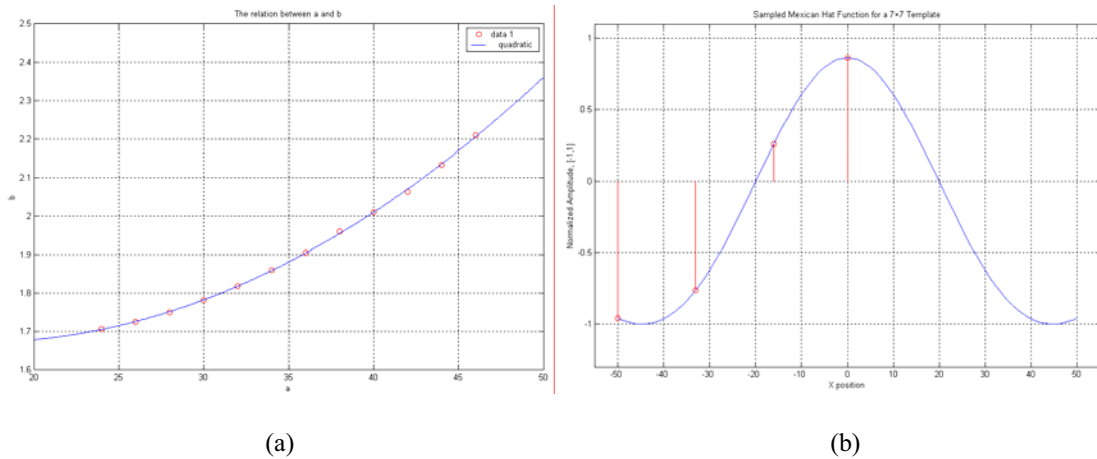
Figure 39

(a) Quadratic interpolation of Mexican hat parameters to satisfy the critical area condition required for correlogram peak(s) detection; (b) Sampled Mexican hat function at resulting in a 7×7 center surround template

When we correlate the template **T** with the correlogram of Figure 37, only the correct peaks will appear above Zero (positive values) that can be thresholded with a constant level. Hence, one avoids the complex and computationally expensive implementation of adaptive thresholding. Figure 40 shows that only the six correct peaks are extracted correctly. Note

that the high intensity flat area (2) is not detected as a peak although its intensity was higher than any other correct peak on the original correlogram.

(a)   (b)
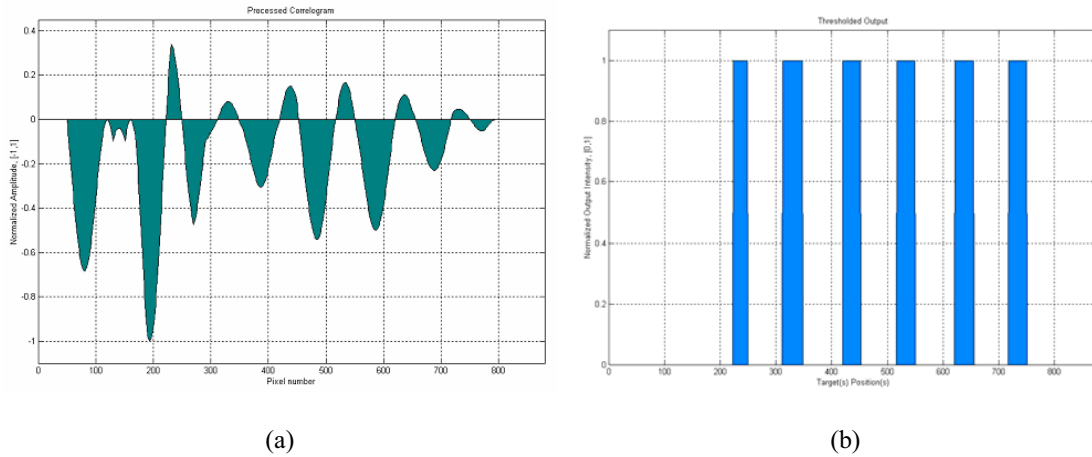
Figure 40

(a) Processed correlogram results in extracting the correct correlation peaks; (b) Processed correlogram after thresholding showing the exact position of the correct correlation peaks

The processed correlogram is shown in Figure 40.a is thresholded at Zero. This extracts the six peaks located on the original correlogram of Figure 37. Note that the flat area (1) and (2) had disappeared and the output represents only the correct results. Figure 40.b, shows the exact position of the correct correlation peaks.

5.5. Experimental Results

I applied this technique on experimental results of the state of the art optical computer *laptop POAC*. Figure 41 shows the input image and its corresponding templates that represent the inputs of POAC in a target tracking experiment. It also shows the output of the correlation process between the input image and one of the templates (T3) both in simulation and experimentation. The figure tells that the experimental result has a background noise that will make thresholding to find the correct target impossible without postprocessing the correlogram.

0°   T1   5°   T2

10°   T3   15°   T4

20°   T5   25°   T6

30°   T7   35°   T8

(a)   (b)

(c)   (d)
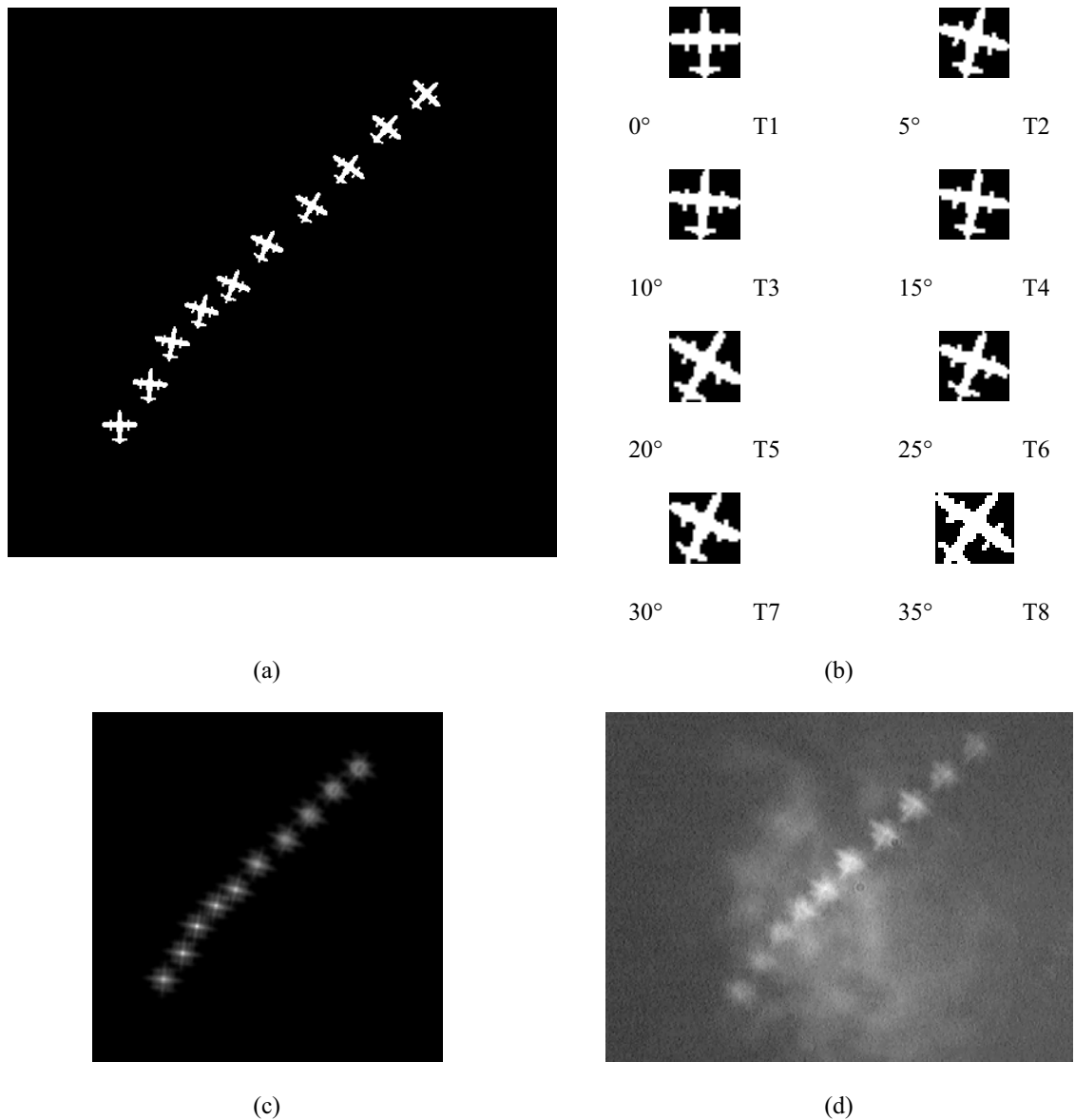
Figure 41

Binary images represent: (a) the original input image (500×500 pixels); and (b) its corresponding series

of rotated templates (32×32). (c) simulation output of correlating the input image and T3 (10° rotation)

(500×500 pixels); and (d) noisy output of POAC correlation of the same input and template (640×480 pixels)

Hence, applying the center surround template, Figure 42.a, to the correlogram as a postprocessing operation yields to the processed correlogram in Figure 43.b.

(a)

(b)

(c)

(d)

(e)
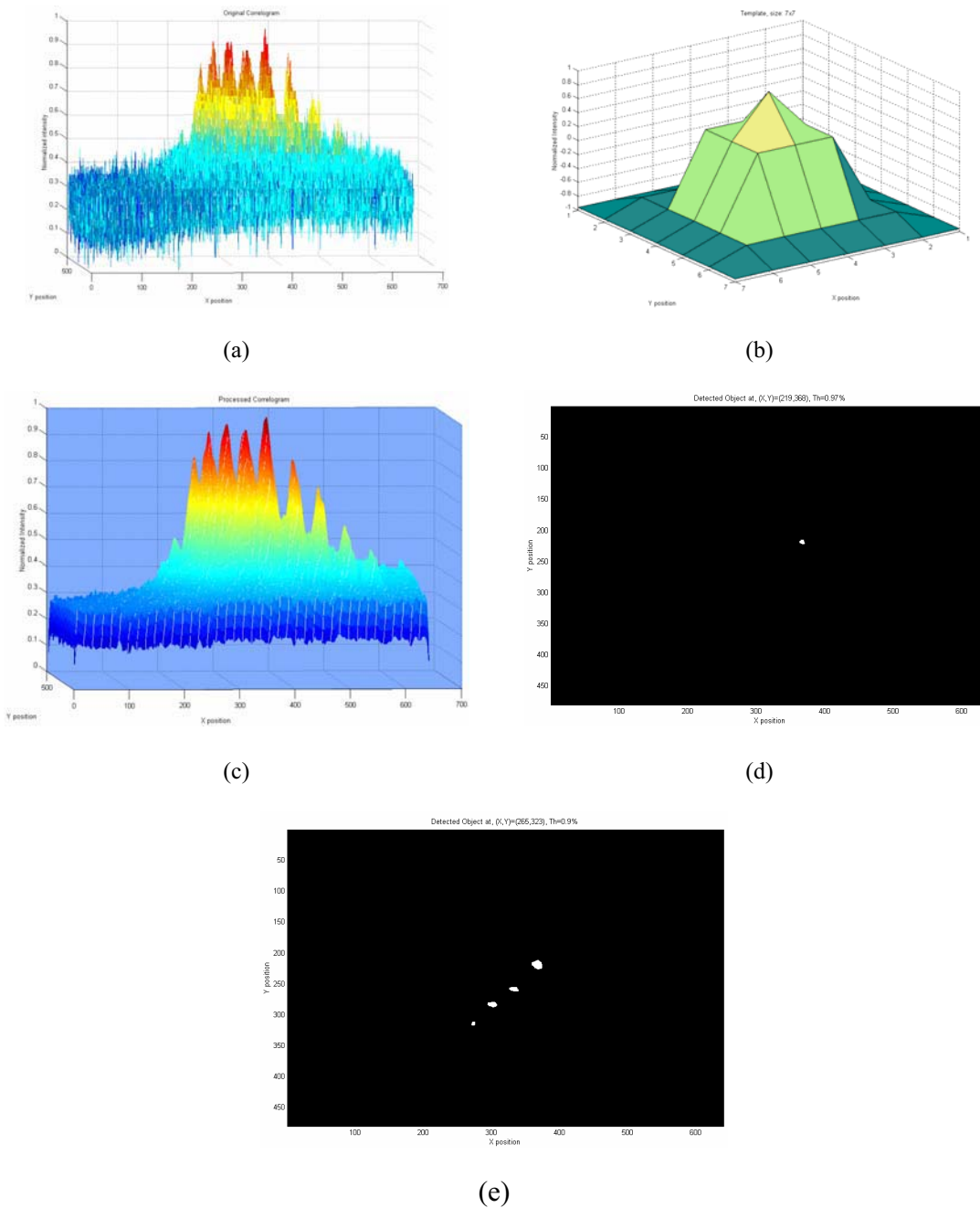
Figure 44

(a) 3D representation of original correlogram resulted from the practical optical correlation; (b) the center surround 7×7 template; (c) 3D representation of the processed correlogram with enhanced peaks; (d) the target is precisely detected with a level thresholding instead of adaptive thresholding; (e) at a lower thresholding level, image classification task can take place.

This figure shows how a practical correlogram can be post processed to provide better detection results as well as simpler thresholding technique. Figure 45.c. shows the post processed correlogram where the background noise had be suppressed and the peaks had been enhanced. Figure 46.d. shows the thresholded post-processed correlogram at a high thresholding level (97% from the normalized one) that leads to detect the exact target object represented by the object template. While in Figure 47.e, the thresholding level is lower (90%) and leads to the detection of similar objects. This is very important when object classification is the task. Furthermore, the size of the detected areas gives an indication about the level of similarity between the targeted object and other objects.

I used digital computer to perform this Mexican-hat filter template developed on Delphi 7.0 running on an AMD-processor 1.67GHz and 512 MB memory. For several template sizes and input correlograms the following data is collected:

Table 5

Practical results for performing the Mexican-hat filter over a digital computer with different sizes

		Correlogram Size	
		256×256	600×400
Template Size	3×3	>15 msec	30 msec
	5×5	15 msec	47 msec

It is obvious that for a small template size (3×3) one can reach the video frame rate (30 frames/sec.) with a standard CCD camera-image size (600×400 pixels). For higher performance, one might need to apply a larger template size. Hence, for the same image size, one cannot fit the processing within the video frame rate. A traditional solution for such problem, I also do propose it, is to use a dedicated digital signal processor (DSP) or a CNN-UM chip when a larger template size, larger correlogram size, or even higher frame rate is needed depending on the application.

5.6. Discussion and Conclusion

In this chapter, three main aspects to design POAC had been addressed and studied. These are:

1. The preferred type of the correlation process

2. The optical representation of negative values of the template elements

3. The application of the postprocessing and thresholding functions

This work is contributing explicit design considerations for these aspects summarized as follows:

1. *"To perform optical correlation using the JTC, the template should be center-mirrored around its central element."*

2. *"Coherent Correlation has higher peak and higher contrast than incoherent correlation for phase-noise free case."*

3. *"The higher the spatial period of the phase noise (low frequency), the higher the coherent correlation peak value and the higher the contrast."*

4. *"A phase shift π is applied to one or more elements of the template when negative values are required."*

5. *"Mexican-hat-like band-pass filter eliminates the low frequencies which usually correspond to background noise and the very high frequencies corresponds to sharp edges"* providing appropriate thresholding.

One Step Forward

In this chapter the aspects of POAC design based on optical JTC setup were considered. The next chapter will show how to exploit these aspects to build the first optical template library for image processing. The library is a part of the present well-known electronic CNN template library that is converted into optics via the concluded aspects introduced in this chapter. Method of such conversion, optical template library, and examples will be presented.

Chapter 6   Optical Template and Algorithm Library for POAC

The previous chapter had shown the main three aspects of POAC design (*coherent correlation*, *optical representation of negative values* and *thresholding* the output) that are used in this chapter to simulate the first optical template library. First, the POAC simulator is introduced. Second, the optical template library is described. Thirds, one example of combining templates into one algorithm is presented. Finally, the results are discussed and major contributions are concluded.

6.1. POAC Simulator

The POAC simulator is designed to perform both *coherent* and *incoherent* correlation as the kernel of the simulation processes. It performs as follows:

1. read-in the preset simulator parameters (input image file name, template values, and the threshold value);
2. read-in input image intensity and amplitude functions;
3. perform coherent and incoherent correlation;
4. threshold the correlograms; and
5. display and save the results

Figure 48 shows a diagram of the simulator algorithm
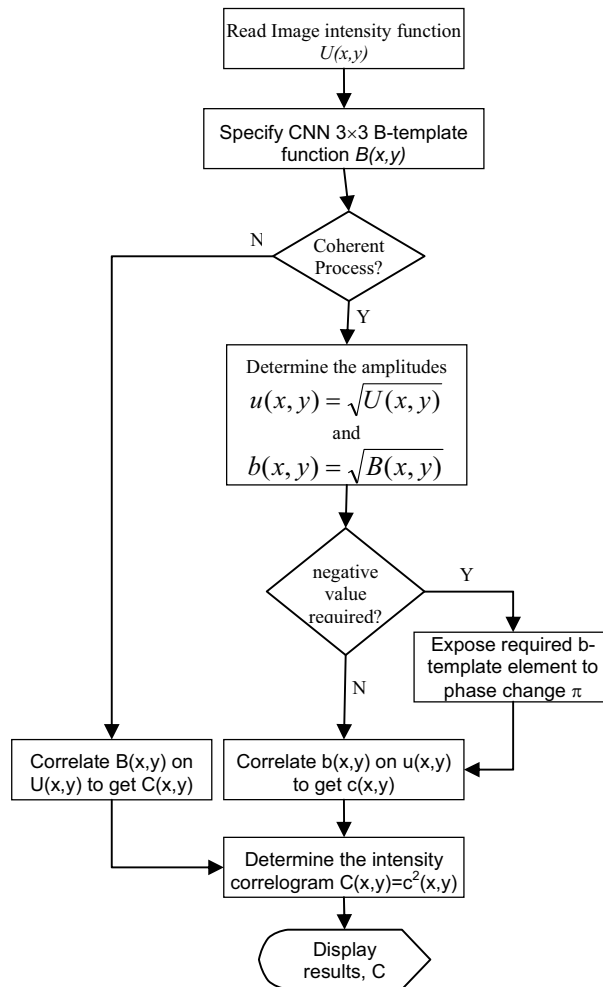
Figure 48

POAC simulator algorithm based on JTC

The practical interpretation of this algorithm is that:

- both the intensity *input image* and the *template* are loaded to two spatial light modulators (SLMs.)
- During the optical wave traveling in space, only the amplitudes are considered and represented by the square root of the intensity values.
- For the negative number representation of the template elements, another SLM with *phase only* modulation changes the required π phase shift.

- The correlation, the kernel of the CNN equation, takes place during the reconstruction process at the correlation plane, in the coherent case
- Finally, a charged coupled device (CCD) camera detects the intensity of the thresholded correlogram.

In this section, I developed an optical feedforward CNN algorithm and simulator based on the present optical implementation of the programmable optical array computer (POAC).

6.2. POAC Optical Template Library

The starting point to create the first POAC template library was to convert the well-established template library of the CNN [52] into optics. Worth to mention that POAC is one optical implementation of CNN, namely feed-forward only CNN class of operations. For practical reasons (present research direction and target applications) the aim was to: (a) prove that optical implementation of CNN template library is possible; and (b) prepare the first optical template library for POAC. Thus, eighteen templates had been selected from the earlier CNN library to become the optical library of interest.

The selection of a template to be optically simulated was based on two concepts: (1) it provides an operation that requires *feed-forward only* class of operation; and (2) being a linear template. These practical considerations were necessary to concentrate the center of attention on well-defined group of templates that would be implemented for the present POAC architecture. Scanning the original CNN template library yield the following templates to construct the first optical template library:

1. Corner Detector
2. Diagonal Line Delete
3. Diagonal Line Detector
4. Horizontal Line Delete
5. Horizontal Line Detector
6. Vertical Line Delete
7. Vertical Line Detector

"Template selection is based on: (1) it belongs to the feed-forward only class; (2) being a linear template"

8. Edge Detector

9. Figure Extractor

10. Southern Element Detector

11. Northern element Detector

12. Eastern Element Detector

13. Western element detector

14. Pattern Matching Finder

15. Left Peeler

16. Right Coroner Detector

17. Junction Extractor of a Skeleton

18. Local Concave Place Detector

Because the goal was not to design new templates but to establish an optical template library, a methodology to convert the original templates had been followed. First, converting CNN template values into optical real values within [0,1]. Then, follow the empirical equation $b_{poac}=|b_{cnn}|/2$, then fine tune the resultant values. Afterwards, follow the simulator algorithm shown in Figure 48 to apply the template operation. The same image library for the original CNN template library had been used for POAC as examples for the sake of comparison.

For appropriate referencing, each template operation is described as follows: (1) optical template operation title; (2) feed-forward *B template* and *threshold z* values; and (3) simulation result example, *input image* and *thresholded* correlograms for both coherent and incoherent (processed output image), all are represented in intensity values.

"Eighteen templates had been selected from the earlier CNN library to become the optical library of interest"

(a) Original image

(b) Corner detection optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram
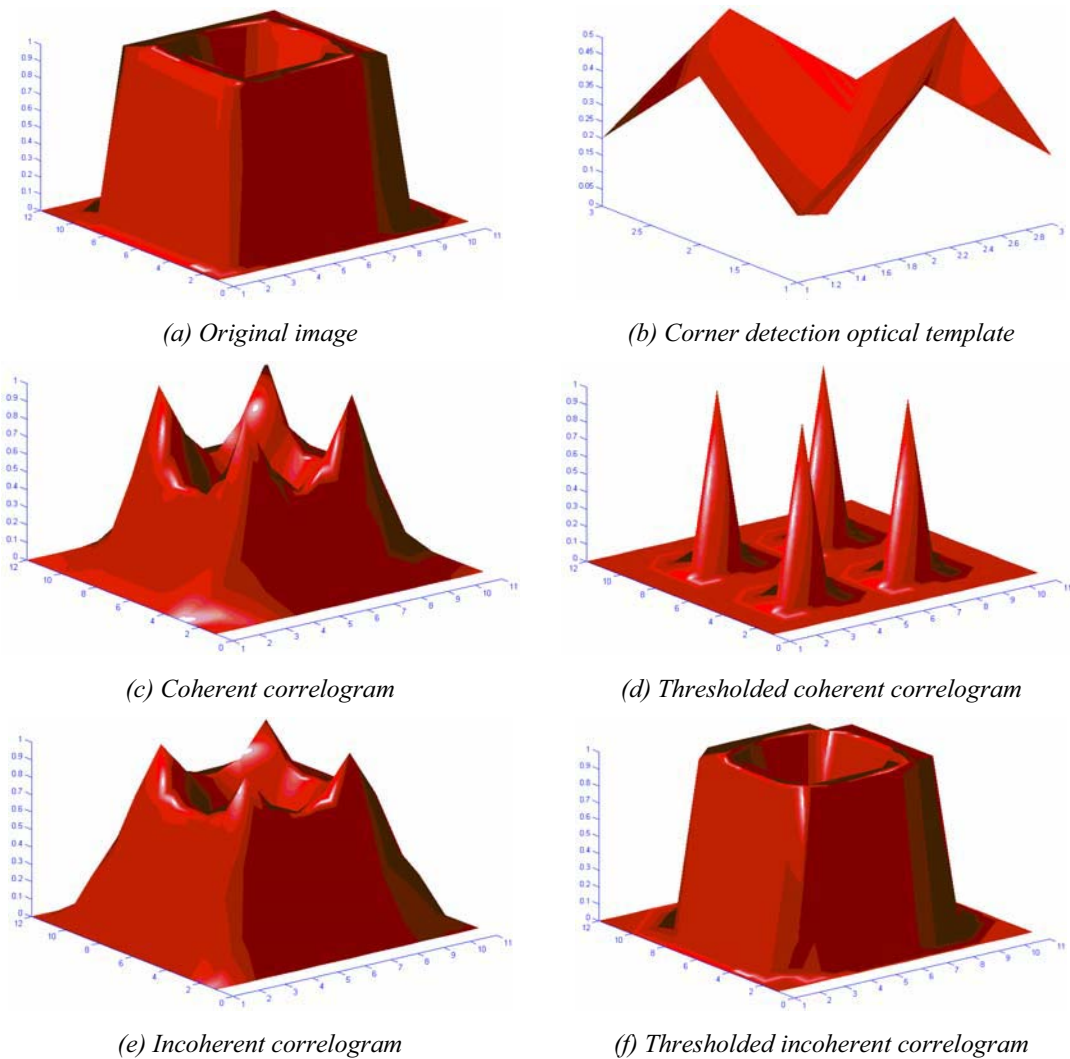
Figure 49

Three dimensional graph showing the input image (a) for a corner detection template operation (b). For coherent correlation (c) the thresholding function works perfectly to detect the corners (d). While for incoherent correlation (e) the thresholding function failed to find the corners at the same level (f). The figure shows that coherent correlation has a higher contrast than the incoherent one.

The present library contents are presented with illustration examples from Figure 50 to Figure 67. In these figures: (a) is the original image; (b) the template; (c) the coherent correlogram; (d) the thresholded coherent correlogram; (e) the incoherent correlogram; (f) the thresholded incoherent correlogram; (g) the original image and its processed version;

and (h) the corresponding template values. The threshold z denotes a percentage of the normalized intensity of the output image. The negative values are simulated by introducing a π phase shift to the desired element, as discussed in the previous chapter.

(a) Original object image

(b) Corner detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.2 & 0.5 & 0.2 \\ 0.5 & 0.0 & 0.5 \\ 0.2 & 0.5 & 0.2 \end{bmatrix}, z = 0.69$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 50
Corner Detector

(a) Original object image

(b) Diagonal line delete optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.1 & 0.0 \\ 0.1 & 1.0 & 0.1 \\ 0.0 & 0.1 & 0.0 \end{bmatrix}, z = 38$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 51
Diagonal Line Delete

(a) Original object image

(b) Diagonal line detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

$$B = \begin{bmatrix} 0.0 & 0.1 & 0.0 \\ 0.1 & -1.0 & 0.1 \\ 0.0 & 0.1 & 0.0 \end{bmatrix}, z = 0.47$$

(g) 2D input image (left) and resultant
output from coherent correlation (right)

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 52
Diagonal Line Detector

(a) Original object image

(b) Horizontal line delete optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.1 & -1.0 & 0.1 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}, z = 0.47$$

(h) Optical feedforward template
(B) and threshold (Z)

Figure 53
Horizontal Line Delete

(a) Original object image

(b) Horizontal line detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.1 & 1.0 & 0.1 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}, z = 0.38$$

(h) Optical feedforward template (B) and threshold (Z)

Figure 54

Horizontal Line Detector

(a) Original object image

(b) Vertical line delete optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.1 & 0.0 \\ 0.0 & -1.0 & 0.0 \\ 0.0 & 0.1 & 0.0 \end{bmatrix}, z = 0.47$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 55
Vertical Line Delete

(a) Original object image

(b) Vertical line detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.1 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.1 & 0.0 \end{bmatrix}, z = 0.38$$

*(h) Optical feedforward template (**B**) and threshold (**Z**)*

Figure 56

Vertical Line Detector

(a) Original object image

(b) Edge detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

$$B = \begin{bmatrix} 0.01 & 0.01 & 0.01 \\ 0.01 & -1.0 & 0.01 \\ 0.01 & 0.01 & 0.01 \end{bmatrix}, z = 0.33$$

(g) 2D input image (left) and resultant
output from coherent correlation (right)

(h) Optical feedforward template
(B) and threshold (Z)

Figure 57
Edge Detector

(a) Original object image

(b) Figure extractor optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

$$B = \begin{bmatrix} 0.03 & 0.03 & 0.03 \\ 0.03 & -1.00 & 0.03 \\ 0.03 & 0.03 & 0.03 \end{bmatrix}, z = 0.32$$

(g) 2D input image (left) and resultant
output from coherent correlation (right)

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 58
Figure Extractor

(a) Original object image

(b) Southern element detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant output from coherent correlation (right)

$$B = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.0 & -1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}, z = 0.28$$

*(h) Optical feedforward template (**B**) and threshold (**Z**)*

Figure 59

Southern Element Detector

(a) Original object image

(b) Northern element detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 0.0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}, z = 0.28$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 60
Northern Element Detector

(a) Original object image

(b) Eastern element detector optical template

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(e)

(f)

(g) 2D input image (left) and resultant output from coherent correlation (right)

$$B = \begin{bmatrix} 0.5 & 0.0 & 0.0 \\ 0.5 & -1.0 & 0.0 \\ 0.5 & 0.0 & 0.0 \end{bmatrix}, z = 0.28$$

*(h) Optical feedforward template (**B**) and threshold (**Z**)*

Figure 61

Eastern Element Detector

(a) Original object image

(b) Western element detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.5 \\ 0.0 & -1.0 & 0.5 \\ 0.0 & 0.0 & 0.5 \end{bmatrix}, z = 0.28$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 62
Western Element Detector

(a) Original object image

(b) Pattern matching finder optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

$$B = \begin{bmatrix} 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 1.0 \end{bmatrix}, z = 0.65$$

(g) 2D input image (left) and resultant
output from coherent correlation (right)

(h) Optical feedforward template
(B) and threshold (Z)

Figure 63
Pattern Matching Finder

(a) Original object image

(b) Left peeler optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}, z = 0.26$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 64
Left Peeler

(a) Original object image

(b) Right corner detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant
output from coherent correlation (right)

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & -1.0 & 0.5 \\ 0.0 & 0.0 & 0.0 \end{bmatrix}, z = 0.53$$

(h) Optical feedforward template
*(**B**) and threshold (**Z**)*

Figure 65
Right Corner Detector

(a) Original object image

(b) Junction extractor optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

(g) 2D input image (left) and resultant output from coherent correlation (right)

$$B = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 1.0 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}, z = 0.47$$

*(h) Optical feedforward template (**B**) and threshold (**Z**)*

Figure 66

Junction Extractor

(a) Original object image

(b) Local concave place detector optical template

(c) Coherent correlogram

(d) Thresholded coherent correlogram

(e) Incoherent correlogram

(f) Thresholded incoherent correlogram

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.2 & 0.2 & 0.2 \\ 0.2 & -1.0 & 0.2 \end{bmatrix}, z = 0.57$$

(g) 2D input image (left) and resultant
output from coherent correlation (right)

(h) Optical feedforward template
(B) and threshold (Z)
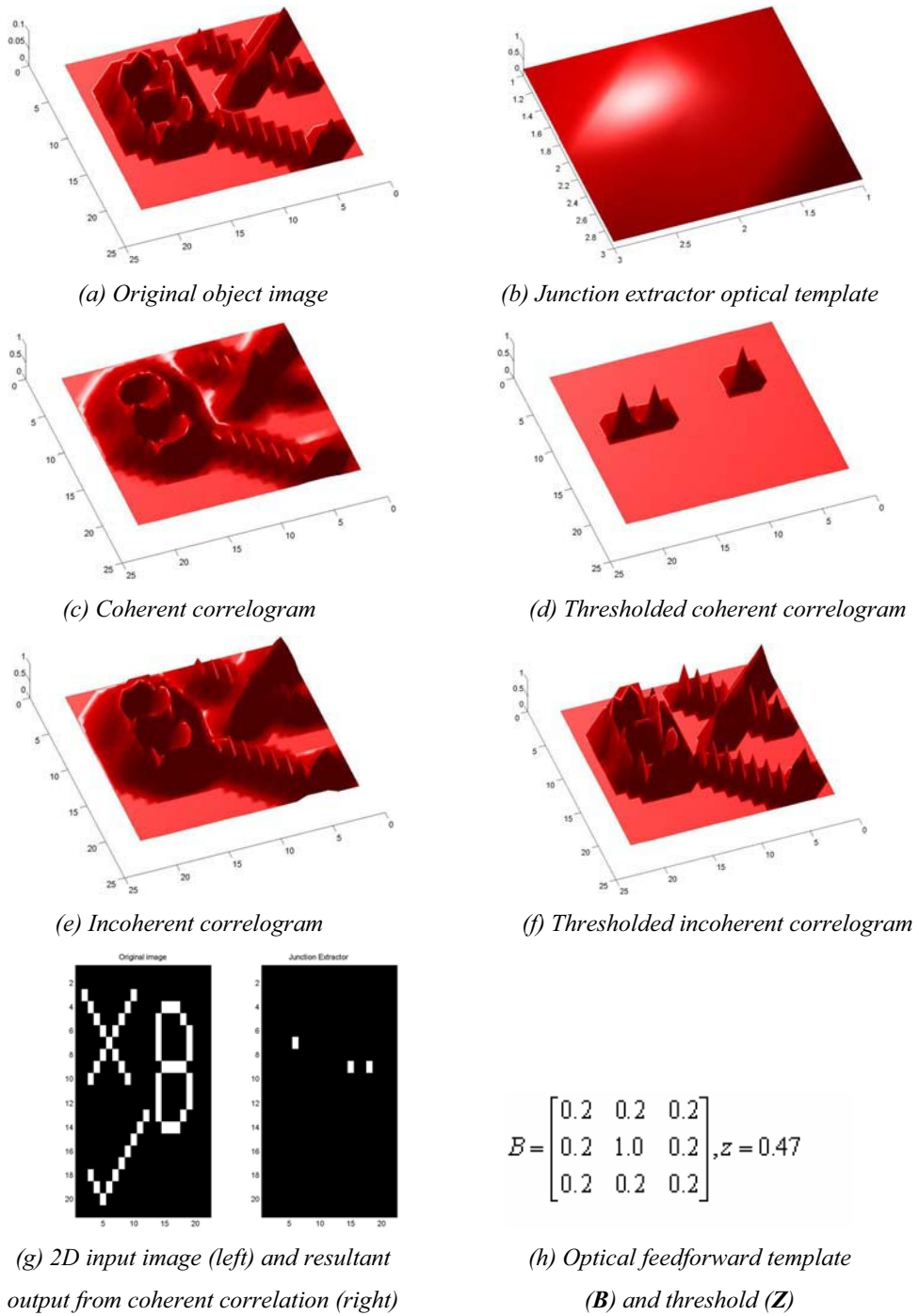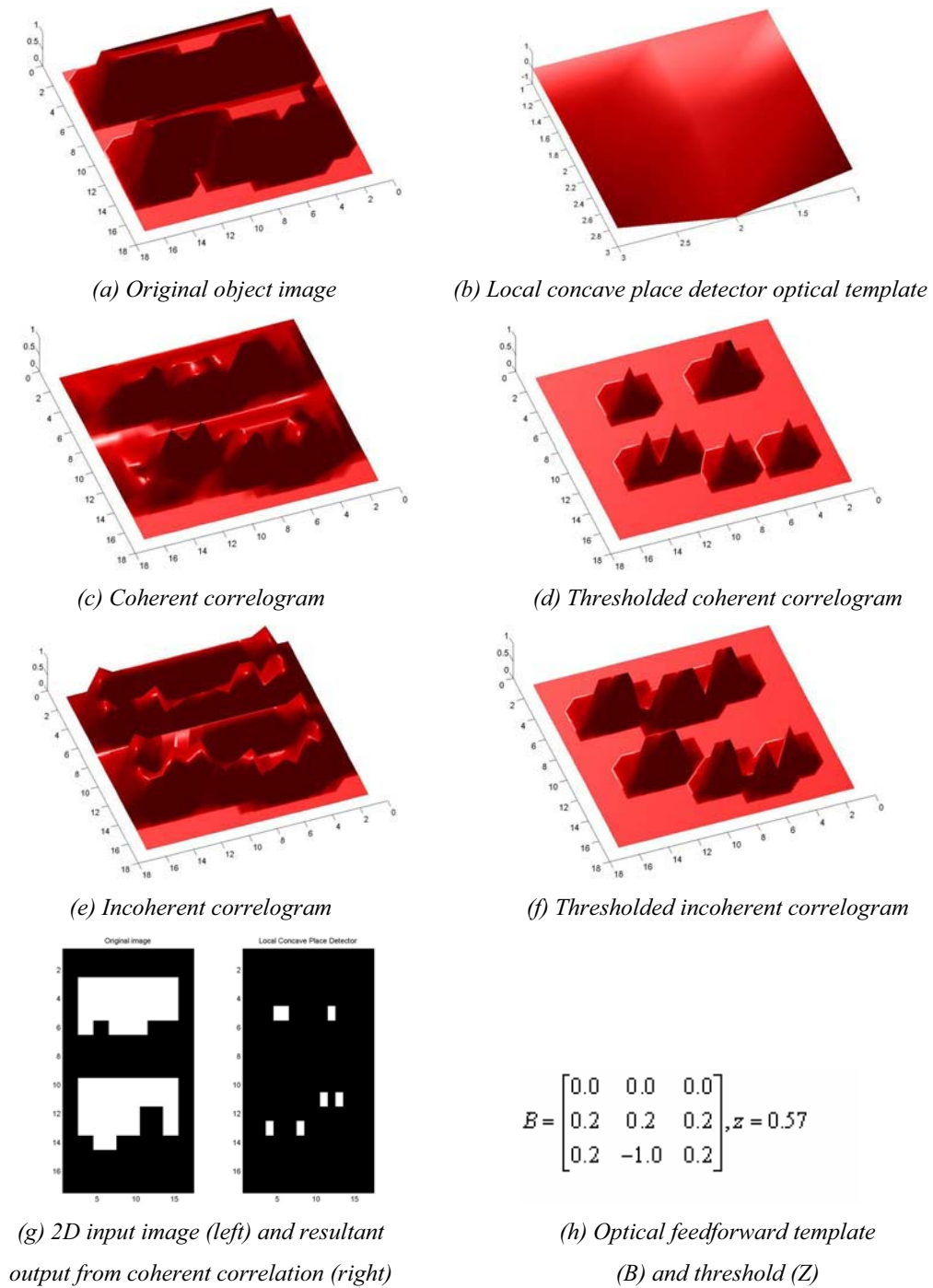
Figure 67
Local Concave Place Detector

From the above library figures one can obviously notice that coherent correlation is the best choice to perform optical template operations. For the same thresholding level, incoherent correlation gives undesired results. The phase change of the negative elements takes place only in the coherent case.

In this section, I designed the first optical template library for POAC that consists of 18 templates. Some of them can be simply modified and extended (e.g. rotating Southern element detector template results in East, West and North element detectors)

6.3. Algorithm Example: Skeletonization

Most of image processes requires more complex operations than those introduced in the present library. Hence, by considering the present library as *micro operations* for POAC, different combinations executed in sequence will result in *macro operations* (algorithms). Although, the present architecture of POAC is not supporting optical *feedback* it would be of certain importance to introduce more complex operations. These operations can be executed by POAC via electronic feedback that will slow down the optical processing. Here, an algorithm uses the optical library to perform one important image processing operation, *skeletonization*.

Skeletons are shape representations which are particularly suitable for describing elongated patterns, such as characters and line drawings [53]. The skeleton of a shape is the locus of the symmetric points of the local symmetries of the shape. Skeletons can be computed from grey-scale images or binary images. The present example concerns binary images.

Skeletonization covers a wide range of applications including real-time human motion analysis [54], shape extraction, micro-organisms detection and even cartoon characters generation for media and advertising purposes. In addition, it is being utilized as preprocess for image and/or pattern recognition algorithms.

An algorithm for skeletonizing binary images was introduced by the previous work of the CNN template library. This is optically simulated here using the present optical simulator. The template element's values are modified to match the nature of optics, e.g. negation, adjusting the thresholding levels, refining the values, etc.

"The skeletonization algorithm consists of 8 templates applies successively for several numbers of iterations"

The algorithm consists of 8 templates applies successively for several numbers of iterations, depend on the object size, until a stable output is achieved. Template B_{i+1} is 45° counterclockwise rotation of B_i, where i=1,2,3,4,5,6,7 and 8. The utilized templates are as follows:

$$B_1 = \begin{bmatrix} -0.11 & -0.10 & 0.11 \\ -0.22 & -1.0 & 0.22 \\ -0.11 & 0.10 & 0.11 \end{bmatrix} \qquad B_2 = \begin{bmatrix} -0.10 & 0.11 & 0.22 \\ -0.11 & -1.0 & .11 \\ -.22 & -.11 & .10 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 0.11 & 0.22 & 0.11 \\ -0.10 & -1.0 & 0.10 \\ -0.11 & -0.22 & -0.11 \end{bmatrix} \qquad B_4 = \begin{bmatrix} 0.22 & 0.11 & 0.10 \\ 0.11 & -1.0 & -0.11 \\ -0.10 & -0.11 & -0.22 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} 0.11 & 0.10 & -0.11 \\ 0.22 & -1.0 & -0.22 \\ 0.11 & -0.10 & -0.11 \end{bmatrix} \qquad B_6 = \begin{bmatrix} 0.10 & -0.11 & -0.22 \\ 0.11 & -1.0 & -0.11 \\ 0.22 & -0.11 & -0.10 \end{bmatrix}$$

$$B_7 = \begin{bmatrix} -0.11 & -0.22 & -0.11 \\ 0.10 & -1.0 & -0.10 \\ 0.11 & 0.22 & 0.11 \end{bmatrix} \qquad B_8 = \begin{bmatrix} -0.22 & -0.11 & -0.10 \\ -0.11 & -1.0 & 0.11 \\ 0.10 & 0.11 & 0.22 \end{bmatrix}$$

z=0.25, for each negative element of each template, a π phase shift is applied.

```
                          ┌──────────────────────┐
                         /  Read image intensity  /
                        /    function (U)        /
                       └──────────────────────┘
                                  │
                       ┌──────────────────────┐
                       │ Determine the square  │
                       │ root of (U) to        │
                       │ get the amplitudes (u)│
                       └──────────────────────┘
                                  │
                         /  Read the first template  /
                        /   intensity function (B)  /
                       └──────────────────────┘
                                  │
                       ┌──────────────────────┐
                       │ Determine the square  │
                       │ root of (B) to get    │
                       │ the amplitude (b)     │
                       └──────────────────────┘
```

Read image intensity function (U)

Determine the square root of (U) to get the amplitudes (u)

Read the first template intensity function (B)

Determine the square root of (B) to get the amplitude (b)

Read Next template intensity function B

Apply phase difference of value (180) to the negative elements of (b)

Correlate $A_t(x,y)$ on $A_s(x,y)$ to get $C(x,y)$

Threshold

Square the result $sqr(C(x,y))$

Display results

Repeat for all of the 8 templates

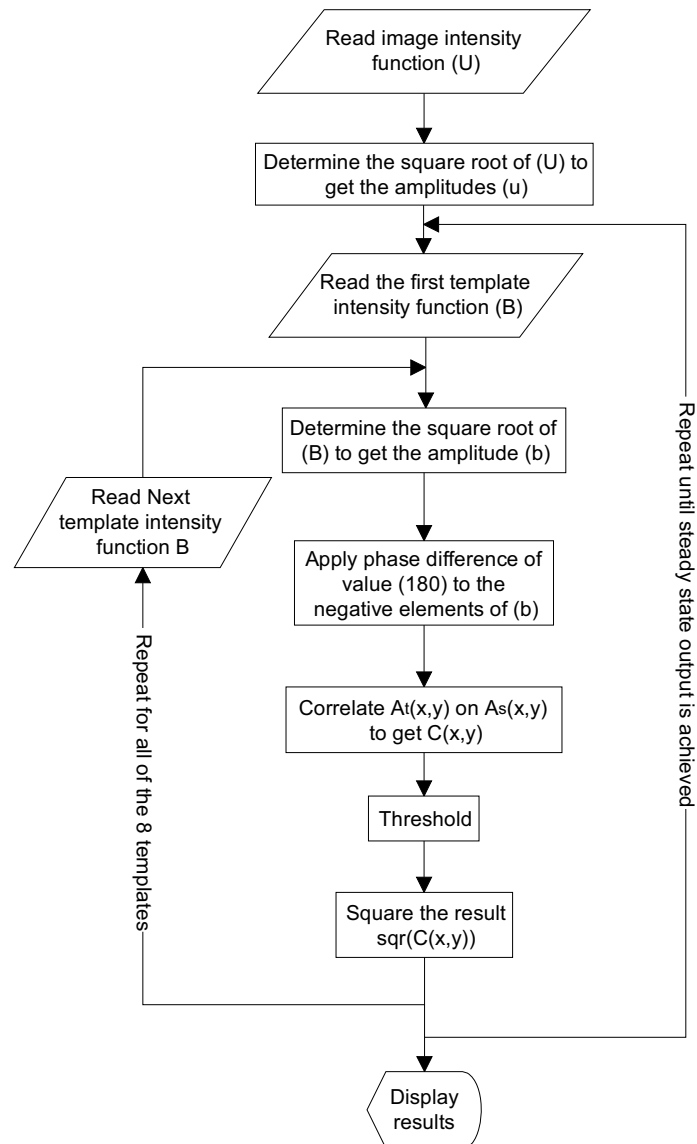Repeat until steady state output is achieved

Figure 68

Optical Skeletonization algorithm using POAC simulator

The algorithm was run over several binary images of objects to find their skeletons. Sample results are shown in Figure 69
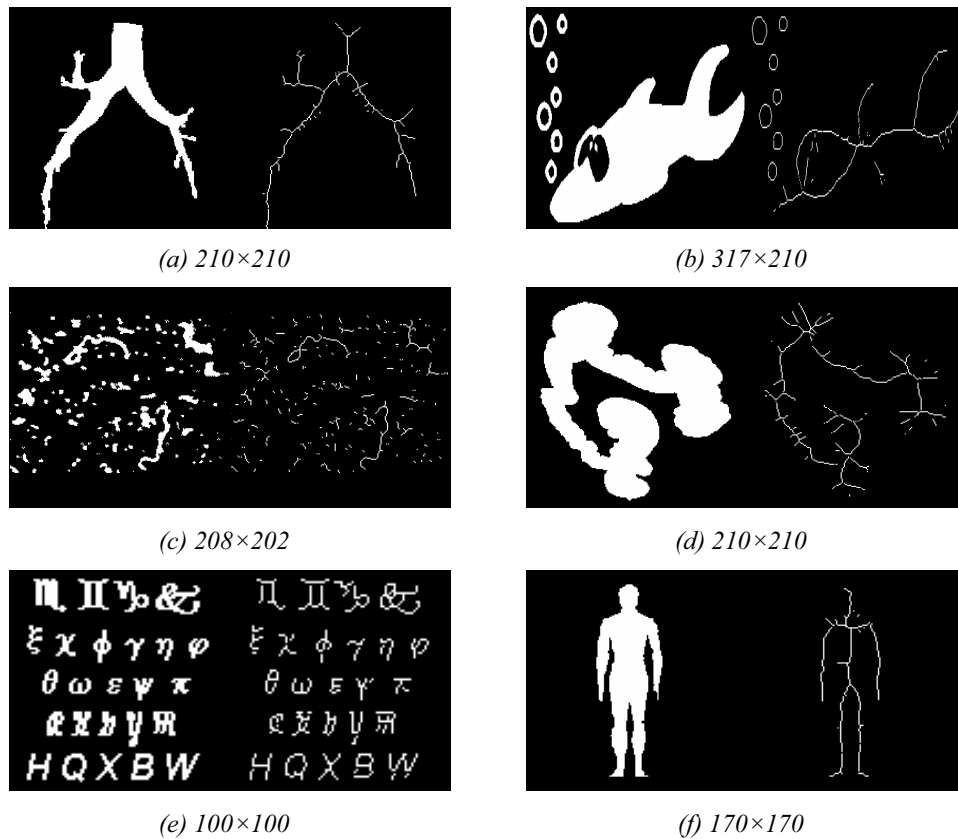
(a) 210×210

(b) 317×210

(c) 208×202

(d) 210×210

(e) 100×100

(f) 170×170

Figure 69

Sample skeletons generated by the present skeletonization algorithm. In (a) to (f): Left figure shows the original image and the right figure shows its processed skeleton. Image size is indicated below each figure in pixels.

The skeletonization algorithm is simulated using the present POAC simulator. The contrast of the correlogram required a thresholding level of only 25% of the maximum intensity, for all of templates included in the algorithm. This high contrast reflects how robust is the thresholding between the skeleton and it background. Hence, promising realization results are foreseen. As a model, realization of such algorithm can be utilized of the present POAC version. The feedback, better called *put-back*, could be done electronically. This would be relatively slow process, but it is important to prove the effectiveness of the simulation.

In this section, I made available, via simulation, macro algorithm to provide the design of feedback or cascaded templates to form complex optical skeletonization task.

6.4. Classified Optical Operations

The simulation results of 18 optical feedforward CNN template operations promised the possibility to realize such operations utilizing the present optical correlator setup. However, an electronically addressed spatial light modulator (ESLM) might be used to realize those templates require coherent correlation. The results are also encouraging to realize more optical operations that might be useful in several applications. Here, a compilation of several templates and operations that would be required for optical information processing. They are grouped to be realized in the future for POAC.

The categories are as follows:

1. Binary Image Operations, Table 6
2. Derivative-Based Operations, Table 7
3. Filtering - noise reduction and other enhancement filters, Table 8
4. Feature Detectors - edges and others features, Table 9
5. Geometric Operations - image rotation, translation and scaling, Table 10
6. Histogram based operation, Table 11
7. Image Analysis - labeling image pixels, Table 12
8. Image Enhancement, Table 13
9. Image Processing Functions, Table 14
10. Image Synthesis - noise image data, Table 15
11. Image Transforms - Fourier, Hough and other transforms, Table 16
12. Linear 2-D Filter Design, Table 17
13. Linear Filtering, Table 18
14. Mathematics-based Operations (Binary and Arithmetic), Table 19
15. Morphology - pixel shape based analysis, Table 20
16. Neighborhood and Block Processing, Table 21
17. Pixel Values and Statistics, Table 22
18. Region-Based Processing, Table 23
19. Smoothing Operations (using low pass filters), Table 24

The details of the classified optical processes are listed in Appendix A. These categories require more studies to prove its ability to be optically realized.

6.5. Discussion and Conclusion

This chapter demonstrated how the first optical template library for POAC had been developed. However, it is important to mention that this templates library is not limited to the present 18 templates. Several image processing operations had been categorized and listed in Appendix A as raw material of extending the library contents.

During the development of this optical library, I was sure to use the same sample images used for the earlier electronic CNN template library for comparison reasons. The achieved results are typical to those of the original template library; however the optical methods utilized here are totally different.

So far, only feedforward optical CNN template operations (B templates) can be realized. Using optically addressable SLM (OASLM) it seems possible to realize feedback A templates as well. The task of my study concerned only with feedforward operations.

One Step Forward

The next chapter will conclude the results of my Ph.D. studies. The thesis I had achieved and some of the results are already embedded in the present POAC architectures.

Chapter 7   Discussion, Conclusions and Future Work

This Ph.D. project had investigated the powerful advantages of optical computing (OC). The vivid improvements of OC in both speed and quality of information processing had motivated this work to build the first set of optical programs for the present Optical Cellular Neural Networks (CNN) architectures. This chapter summarizes the conclusions and thesis of this work as well as the planned future work that show how to utilize and extend the achieved results.

7.1. Thesis One

I showed the essential differences between optical coherent and incoherent correlations. I described, analyzed and proposed with proof how they can be utilized for optical CNN implementation.

I have shown that the optical CNN implementation performs better in coherent mode of the modified Joint Fourier Transform Correlator (t2-JTC) than in the incoherent one. I have studied the differences between the two modes of operation and how to implement templates in these architectures. Finally I worked out fast post-processing procedures to enhance the discrimination capability.

Published in JCSC, ECCTD03

I have defined models and developed algorithms by which optical coherent and incoherent correlation is being performed and found the differences between these modes of operations.

The *time-multiplexed* incoherent correlation requires photoelectric accumulation to integrate the images needed for correlation. I showed via numerical simulation that the

incoherent correlation is immune to phase noise and is simple to be realized. However, I pointed out that coherent correlation has higher contrast than the incoherent one. (*see section 5.2.*) That is why in coherent mode the thresholding can be more effectively implemented. (*see section 5.2.*)

I showed via simulation and experimentation that the need of center-mirroring is dependent on the actual architecture of the optical correlator to achieve correlation. I gave a prescription table for how to use it in different types of optical implementations with special regards to our t2-JTC. (see section 5.1.)

These results have been successfully verified by experiments in different architectures.

I designed fast post-processing Mexican-hat functions with selected optimum parameters to yield correlograms that can be simply thresholded.

My simulation and experimental results show that the complexity of adaptive thresholding is avoided (*see section 5.4.*). The size of the template and its values are crucial to properly eliminate the background of the peaks. My functions improved the signal to noise ratio (SNR) considerably.

7.2. Thesis Two

I proved that the optical implementation of negative template values is practically possible by applying a phase shift of value π to such values.

It was an unsolved crucial question to optically implement negative template values. I had defined the problem and proposed a practical solution to achieve negative value effect in optics.

The nature of light does not provide negative values, if one is dealing with intensity only. According to my solution to perform negative template value, one has to shift the phase of this template element (pixel) by value of π (see section 5.3.). I verified the proposed technique by digital simulation. I proposed that cascaded phase-only and amplitude-only spatial light modulators (SLMs) have to be placed in the template arm of

the programmable optical array computer (POAC) and designed its architecture. The first SLM has to be imaged onto the second one and commonly Fourier transformed by a lens, (see section 4.3.3.). This solution requires the coherent mode of operation.

Published in JCSC, ECCTD03

7.3. Thesis Three

I developed the first optical image processing template library of operations and algorithms to be implemented by the programmable optical analogic array computer (POAC).

My specific task in this Ph.D. research work was to find, design and test via simulation the availability of an optical template library for POAC. (see chapter 6.) Hence, find optical programs for the "Programmable Optical Analogic Array Computer" (POAC). So far, eighteen optical feed-forward templates had been designed, as well as, one complex algorithm.

Published in ECCTD03, IEEE MWSCS, IEEE CNNA2004

Starting from the well-established CNN template library and based on the state of the art POAC architecture, I developed a simulator for feedforward optical CNN (see section 6.1.) I designed the first optical template library for POAC that consists of 18 templates; they are utilizing my results in thesis 2. Some of them can be simply modified and extended (e.g. rotating Southern element detector template results in East, West and North element detectors) (see section 6.2.). I developed, via simulation, a macro algorithm to design cascaded templates to solve complex optical skeletonization task (see section 6.3.).

It has been shown that for proper implementation of the library, coherent mode of operation and the architecture designed for negative template values (see Thesis 2) is required.

7.4. Application of the Results

The main application field of the results achieved in this work is to be utilized as micro operations for POAC. The high capabilities of POAC are in need for programs. The templates and their library that I had developed in this work are leading to the standardization and can be used to commercialize the system.

The most promising application areas seems to be the image understanding and processing tasks where the use of an optical computer is reasonable and fast. Optical computer implementations are superior to their digital signal processing counterparts especially in those tasks, where great numbers of correlations have to be completed with large templates on high-resolution images

VanderLugt type, optical filtering based correlators are fast, due to their inherent massive parallelism. However, because of the time consuming process of matched filter design, they are not applicable when the task requires fast (re-) programming. As there is no need of any matched filter calculation in the advanced t2-JTC architecture, data gained from the previous image frames become applicable.

Other applications are suggested, like stereo image processing, motion estimation for image compression or diverse correlation based identification algorithms, where not only fast programmable optical correlation is necessary, but local parallel pre- and post-processing (CNN) is also required.

In addition to what is mentioned above, the following are also recommended applications:

- Multiple target recognition and tracking including collision avoidance;

- Security (document security and face recognition);

- Database mining (mainly optical searching for fingerprints or image archives databases);

- Traditional image processing and pattern recognition tasks;

- Industrial quality control and recognition of assembly parts;

- Complex-valued functions operations with large capacity (e.g. 1000×1000 complex numbers);

- Microscopic recognition, tracking and guiding of laser-tweezer-like manipulation.

7.5. Future Work

At this point of my research, it becomes visible to me that the required technology to realize a part of the results to serve the goals of implementing POAC is presently not available. Specifically speaking, implementing the optical template library. Recent technology is offering high quality SLMs that can work in *mostly-phase-only* mode of operation. In the near future, I do expect that these devices will offer pure *phase-only* mode, hence implementing the optical template library will be in reach of hands. On-shelf optical templates (*micro-programs*) and standard optical algorithms will be available. One can also foresee complete applications with standard general purpose optical template modules. In this framework, I suggest two research programs that their goals would be as follows:

Research Program I: Realizing the optical template library for POAC

1. Find, test and measure the best SLM that is able to work in phase-only mode and embed it in POAC

2. Implement the present optical template and measure their results over a set of image database

3. Use appropriate optimization method(s) to *fin-tune* the experimental results

4. Utilize genetic programming, preferably over GRID computers, to find solutions for the image processing tasks listed in Appendix A

Research Program II: Optical implementation of complex-valued arrays correlation

1. Based on the results on Program I (points 1 and 2) it will be possible to implement complex numbers for both the input image and the template in the present POAC setup

2. Perform the existing template operations over the complex-valued input pixel arrays

3. Study the results of *single-input-single-template* (SIST); multi-*input-single-template (MIST)*; *single-input-multi-template (SIMT);* and *multi-input- multi-template (MIMT)* fully-parallel complex-valued modes of optical operations.

Beyond developing an optical template library for POAC, but still within the frame of optical computing, I also started some experiments, *Appendix B*, to work with diffractive optical elements (DOEs) to be utilized as diffractive optical processors (DOPs). The original theories of these experiments as well as some earlier results are available [55, 56]. From this recent knowledge and the state of the art available technologies I propose the following research program:

Research Program III: Design and implementation of CGH for optical image processing

- Study the earlier methods dedicated to design and develop DOEs and computer generated holograms CGHs, starting from the basic ones (Lohmann and Lee methods) and ending with the most recent optimization algorithms (iterative Fourier transform algorithms IFTA)

- Fabricate and measure the quality of each method for testing samples (standard testing optical maps and charts are available)

- Calculate and simulate the diffractive window distribution (hologram mask) taking into account of constraints to get target image in the output signal window (SW) with high quality as possible

- Define specific image processing tasks and focus the research efforts to develop them

- Test, measure and refine the design

- General image-processing operations (such as *skeletonizing*). Single-step or multiple-step algorithms. Genetic programs (including iterations) running on computer networks could find the appropriate DOEs.

Finally, I hope that my studies had contributed some results that had, or will, broaden the boarders of science in this field.

Appendices

Appendix A   Classified Optical Operations

Table 6

Binary Image Operations

No.	Operation	Description
1	Applylut	Perform neighborhood operations using lookup tables
2	Bwarea	Compute area of objects in binary image
3	Bweuler	Compute Euler number of binary image
4	Bwfill	Fill background regions in binary image
5	Bwlabel	Label connected components in binary image
6	Bwmorph	Perform morphological operations on binary image
7	Bwperim	Determine perimeter of objects in binary image
8	Bwselect	Select objects in binary image
9	Dilate	Perform dilation on binary image
10	Erode	Perform erosion on binary image
11	Makelu	Construct lookup table for use with *applylut*

Table 7

Derivative-Based Operations

No.	Operation	Description
1	GF	Gradient filters
2	BDF	Basic derivative filters
3	PGF	Prewitt gradient filters
4	SGF	Sobel gradient filters
5	AGF	Alternative gradient filters
6	GGF	Gaussian gradient filters
7	BSDF	Basic second derivative filter
8	FDL	Frequency domain Laplacian
9	GSDF	Gaussian second derivative filter
10	ALF	Alternative Laplacian filters
11	SDGD	Second-Derivative-in-the-Gradient-Direction

Table 8

Digital Filters - noise reduction and other enhancement filters

No.	Operation	Description
1	Mean Filter	noise reduction NR using mean of neighborhood
2	Median Filter	NR using median of neighborhood
3	Gaussian Smoothing	NR using convolution with a Gaussian smoothing kernel
4	Conservative Smoothing	NR using maximum and minimum of neighborhood
5	Crimmins Speckle Removal	more complex NR operator
6	Frequency Filters	high and low pass image filters, etc
7	Laplacian/Laplacian of Gaussian Filter	edge detection filter
8	Unsharp Filter	edge enhancement filter

Table 9

Feature Detectors - edges and others features

No.	Operation	Description
1	Roberts Cross Edge Detector	2×2 gradient edge detector
2	Sobel Edge Detector	3×3 gradient edge detector
3	Canny Edge Detector	non maximal suppression of local gradient magnitude
4	Compass Edge Detector	3×3 gradient edge detectors
5	Zero Crossing Detector	edge detector using the Laplacian of Gaussian operator
6	Line Detector	line (as distinct from edge) feature detector

Table 10

Geometric Operations - image rotation, translation and scaling

No.	Operation	Description
1	Scale	change image content size
2	Rotate	change image content orientation
3	Reflect	flip over image contents
4	Translate	change image content position
5	Affine Transformation	general image content linear geometric transformation
6	Imcrop	Crop image
7	Imresize	Resize image
8	Imrotate	Rotate image
9	interp2	2D data interpolation
10	copyimage	copy or crop image
11	flipimage	flip image top to bottom
12	mirrorimage	reverse image left to right

13	resize	resize image
14	resizeex	resize image with pixel interpolation
15	rotate	rotate image any angle
16	rotate90	rotate image 90 degrees
17	setimagearea	set region of interest in image, all subsequent operations affect only the region of interest

Table 11

Histogram-based Operations

No.	Operation	Description
1	ConStr	Contrast stretching
2	Equal	Equalization
3	MedF	median filtering
4	MaxF	maximum filtering
5	MinF	minimum filtering

Table 12

Image Analysis - labeling image pixels

No.	Operation	Description
1	Edge	Find edges in intensity image
2	Qtdecomp	Perform quadtree decomposition
3	Qtgetblk	Get block values in quadtree decomposition
4	Qtsetblk	Set block values in quadtree decomposition
5	calcavglevel	calculate average brightness
6	calchisto	calculate histogram
7	Getpixelcolor	read a pixel value
8	isgrayscaleimage	Determine if image is grayscale
9	pixelcount	count pixels in brightness range

10	Setpixelcolor	set pixel value
11	Intensity Histogram	the image intensity distribution
12	Classification	mapping from pixel values to feature or object classes
13	Connected Components Labeling	grouping pixels with the same class label into regions

Table 13

Image Enhancement

No.	Operation	Description
1	Histeq	Enhance contrast using histogram equalization
2	Imadjust	Adjust image intensity values or colormap
3	Imnoise	Add noise to an image
4	medfilt2	Perform 2-D median filtering
5	ordfilt2	Perform 2-D order-statistic filtering
6	wiener2	Perform 2-D adaptive noise-removal filtering

Table 14

Image Processing Functions

No.	Operation	Description
1	blur	smoothing filter
2	blurthresh	smoothing filter with threshold
3	brightenmidrange	raise brightness of mid levels
4	changebright	increase or decrease brightness
5	Divide	divide by factor
6	dilate	darken by enlarging dark areas
7	erode	brighten by enlarging bright areas
8	exchangelevel	change range to new value
9	expandcontrast	increase contrast

10	gammabrighten	adjust gamma brightness
11	histobrighten	histogram brightening
12	histoequalize	histogram equalization
13	kodalith	create high contrast image
14	limitlevel	set maximum brightness level
15	matrixconv	matrix convolution, 3 x 3 kernel
16	matrixconvex	matrix convolution, 63 x 63 kernel
17	medianfilter	median filter, to despeckle an image
18	multiply	multiply by factor
19	negative	negative image
20	outline	edge detection filter
21	pixellize	pixellation, to create mosaic effect
22	removenoise	median filter, to despeckle an image
23	sharpen	sharpening filter
24	sharpengentle	gently sharpen
25	threshold	set minimum brightness level
26	usetable	assign brightness based on table
27	addimage	add two images
28	andimage	AND two images
29	cover	overlay two images, transparent range of colors
30	coverclear	overlay two images, transparent color
31	multiplyimage	multiply two images
32	orimage	OR two images
33	subimage	subtract two images
34	wtaverage	weighted average of two images
35	xorimage	XOR two images
36	zeroimage	set all pixel values to a value

Table 15

Image Synthesis - noise image data

No.	Operation	Description
1	Noise Generation	various types of image noise

Table 16

Image Transforms - Fourier, Hough and other transforms

No.	Operation	Description
1	Distance Transform	maps binary images to distance from background
2	Fourier Transform	maps image into spatial frequency domain
3	Hough Transform	maps image into votes for specified shapes
4	dct2	Compute 2-D discrete cosine transform
5	Dctmtx	Compute discrete cosine transform matrix
6	fft2	Compute 2-D fast Fourier transform.
7	Fftn	Compute N-D fast Fourier transform
8	Fftshift	Reverse quadrants of output of FFT
9	idct2	Compute 2-D inverse discrete cosine transform
10	ifft2	Compute 2-D inverse fast Fourier transform.
11	Ifftn	Compute N-D inverse fast Fourier transform.
12	Iradon	Compute inverse Radon transform
13	Phantom	Generate a head phantom image
14	Radon	Compute Radon transform

Table 17

Linear 2-D Filter Design

No.	Operation	Description
1	Freqspace	Determine 2-D frequency response spacing
2	freqz2	Compute 2-D frequency response
3	fsamp2	Design 2-D FIR filter using frequency sampling
4	ftrans2	Design 2-D FIR filter using frequency transformation
5	fwind1	Design 2-D FIR filter using 1-D window method
6	fwind2	Design 2-D FIR filter using 2-D window method

Table 18

Linear Filtering

No.	Operation	Description
1	conv2	Perform 2-D convolution
2	convmtx2	Compute 2-D convolution matrix
3	Convn	Perform N-D convolution
4	filter2	Perform 2-D filtering.
5	Fspecial	Create predefined filters

Table 19

Mathematics-based Operations (Binary and Arithmetic)

No.	Operation	Description
1	NOT	Logical NOT pointwise inversion of a binary image
2	AND	Logical AND pointwise logical ANDing/NANDing of two binary images
3	OR/NOR	Logical OR/NOR pointwise logical ORing/NORing of two binary images
4	XOR/XNOR	Logical XOR pointwise logical XORing/XNORing of two binary images
5	SUB	Subtracting specific part of the image from the whole image, defined as: Logical (A AND (NOT(B)))
6	ADD	$c = a + b$ pointwise addition: image + image (or constant)
7	SUB	$c = a - b$ pointwise subtraction: image (or constant)
8	MUL	$c = a * b$ pointwise multiplication: images * image (or constant)
9	DIV	$c = a / b$ pointwise division: images / image (or constant)
10	LOG	$c = \log(a)$
11	EXP	$c = \exp(a)$
12	SQRT	$c = \mathrm{sqrt}(a)$
13	TRIG.	$c = \sin/\cos/\tan(a)$
14	INVERT	$c = (2^B - 1) - a$
15	Blending	pointwise linear combination of two images
16	Bitshift Operators	pointwise scaling of an image

Table 20

Morphology-based Operations - pixel shape based analysis

No.	Operation	Description
1	BolConv	Boolean Convolution
2	Prop	Propagation
3	Dilation	grow image regions
4	Erosion	shrink image regions
5	Opening	structured removal of image region boundary pixels
6	Closing	structured filling in of image region boundary pixels
7	Hit and Miss Transform	image pattern matching and marking
8	Thinning	structured erosion using image pattern matching
9	Thickening	structured dilation using image pattern matching
10	Skeletonization/Medial Axis Transform	finding skeletons of binary regions

Table 21

Neighborhood and Block Processing

No.	Operation	Description
1	Bestblk	Choose block size for block processing
2	Blkproc	Implement distinct block processing for image
3	col2im	Rearrange matrix columns into blocks
4	Colfilt	Perform neighborhood operations using columnwise functions
5	im2col	Rearrange image blocks into columns
6	Nlfilter	Perform general sliding-neighborhood operations

Table 22

Pixel Values and Statistics

No.	Operation	Description
1	corr2	Compute 2-D correlation coefficient
2	Imcontour	Create contour plot of image data
3	Imfeature	Compute feature measurements for image regions
4	Imhist	Display histogram of image data
5	Impixel	Determine pixel color values
6	Improfile	Compute pixel-value cross-sections along line segments
7	mean2	Compute mean of matrix elements
8	Pixval	Display information about image pixels
9	std2	Compute standard deviation of matrix elements

Table 23

Region-Based Processing

No.	Operation	Description
1	Roicolor	Select region of interest, based on color
2	Roifill	Smoothly interpolate within arbitrary region
3	roifilt2	Filter a region of interest
4	Roipoly	Select polygonal region of interest

Table 24

Smoothing Operations (using low pass filters)

No.	Operation	Description
1	ULF	Uniform linear filter
2	TLF	Triangle linear filter
3	GLF	Gaussian linear filter
4	MNLF	Median non-linear filter
5	KNLF	Kuwahara non-linear filter
6		Other filters utilizing the frequency domain

Appendix B   Related Results to the Main Topic: Towards Diffractive Optical Processors (DOPs)

In the past 30 years, optics has changed its philosophy and culture giving birth to photonics for transmitting and processing light. The diffractive advances in microelectronics field for photonics kept pace with the progress of holography for signal processing. Extension of knowledge in technologies for diffractive CAD/CAM and improvements in production efficiency made it possible rapidly to create effective components with quality level close to industrial needs.

Recently, progress in diffractive optical elements (DOEs) design has accelerated [57]. The challenge is now to integrate DOE components in optical, electronic or mechanical systems. It is therefore the next task of be undertaken. One further step, which seems to be achieved in the next 10 years, is diffractive optical processors (DOPs). These devices will be able to process light waves by a single very high resolution printed transparent computer generated holograms (CGHs). Recent attempts had been reported in [55].

B.1. Design Characteristics of Diffractive Optical Elements (DOEs)

Several design and encoding methods have been proposed in the literature over the past 30 years, back to when Lohmann [58,59] first introduced the concept of computer-generated holography/holograms (CGH). Simple FFT-based calculations and detour-phase encoding techniques were investigated.

Along with the increasing in computing power, more sophisticated iterative design methods and encoding techniques have been reported [60]. Furthermore, since Lesem and Hirch proposed the kinoform encoding method over phase substrate [61,62] along with the development of new microlithographic fabrication technologies, several new applications have been proposed.

DOEs can be designed by *direct* or *iterative* techniques. Direct techniques are limited in number and more specific for small range of applications, e.g. array illuminators and beam shaping. They exhibit either cell or pixel-oriented encoding strategies. However, they belong to two main categories: *Dammann gratings* [63] and Talbot Array illuminators.

On the other hand, the *iterative* design algorithms imply optimization techniques. They can be grouped within four major categories [64,65,66]:

1. *Unidirectional algorithms*: these algorithms perform a single-pixel change at each iteration and analyze their effect on the reconstruction. Direct Binary Search (DBS), Simulated Annealing (SA) and iterative Discrete On-Axis (IDO) algorithms belong to this category.

2. *Bidirectional algorithms (IFTA)*: perform one or more global transformations at each iteration, using forward and backward propagators. They project the result on to several different constraints defined by the type of application, fabrication technologies, operation, etc. These include Gerchberg-Saxton-based algorithms [67] and the Ferwerda and Yang-Gu algorithms [68,69]. They are referred to as Iterative Fourier Transformation Algorithms (IFTA).

3. *Genetic algorithms*: they are based on evolutionary programming

4. *Global optimization*: perform global optimization that is related to encoding techniques such as error diffusion.

B.2. Practicing with Lohmann Encoding Method

When the expansion of the technology of diffractive optics truly begun in the mid-1960s, as a result of Lohmann's pioneering work on digital simulation of optical holograms [70,71,72], computers and computer-controlled pen-and-ink plotters were already available. They permitted the generation of almost-arbitrary black and white patterns, but the resolution was poor and therefore a photo-reduction step was necessary to achieve a reasonable angular spread of the signal window. It was soon realized that four times higher diffraction efficiencies could be achieved by bleaching the photographic emulsion [73], which leads to a phase-only binary transparency. This traditional patterning technique is

still perfectly applicable; use of modern laser printers makes the process simple and reasonably accurate [74].

Electron beam lithography can produce nanostructures [75], below 50nm. Multilevel profiles can be generated in electron-sensitive resist by controlling the electron dose per unit area, which again eliminates mask-alignment errors [76]. Furthermore, the process known as proportional etching can be applied to transfer the resist pattern into the substrate [77]. However, direct-write electron beam lithography is an expensive fabrication method because of the high cost of both equipment and its maintenance.

Because most of the manufacturers want to avoid the sophisticated machinery to fabricate diffractive optics, we chose the traditional method (microlithography of 2μm resolution), to create our detour-phase holograms and to verify the basic concept. This resolution is high enough for our target applications.

A computer generated hologram (CGH) offers a potential technology by which one may specify and manufacture reference optical diffraction elements (ODEs). Because measurements are made in the *far-field*, the ODE may be considered to be a *Fraunhofer diffracting element*. Brown and Lohmann et. al. [70] developed a method called *detour phase* to encode the amplitude and phase of a Fraunhofer diffraction pattern in the late 1960's. Subsequent work has shown that improvements to the detour phase method can be found by using iterative error reduction methods [78]. High quality binary holograms needed to produce high quality images require high resolution writing techniques.

```
┌─────────────────┐
│ Load the 2D image │
│ intensity matrix, I │
└─────────────────┘
          │
          ▼
┌─────────────────┐
│  Set the window  │
│ width b and cell │
│     size a       │
└─────────────────┘
          │
          ▼
┌─────────────────┐
│ Fourier Transform │
│  the complex-    │
│ valued I matrix, C │
└─────────────────┘
          │
          ▼
┌─────────────────┐
│  Quantize the   │          Module 1
│  Amplitude A as a │
│   height of the  │
│     rectangle,   │
│  d=round(A*a/2)  │
└─────────────────┘
          │
          ▼
┌─────────────────┐
│ Quantize the Phase │     Module 2
│       P,         │
│  shift=round((P/2π) │
└─────────────────┘
          │
          ▼
┌─────────────────┐
│  Combine d and   │
│ shift to get a binary │
│  encoded one     │
│  period matrix, X │
└─────────────────┘
          │
          ▼
┌─────────────────┐
│ Duplicate the X  │──────────────┐
│ matrix to enhance │              │
│  the frequency   │              │
│  content, X'     │              │
└─────────────────┘              │
          │                      ▼
          ▼              ┌─────────────────┐
┌─────────────────┐      │ Inverse Fourier  │
│ Save and print X │      │ Transform X and  │
│  and X' in the   │      │ X' to examine the │
│ required format for │   │ predicted results, Y │
│   testing and    │      └─────────────────┘
│  manufacturing   │
└─────────────────┘
```

Figure 70

The utilized algorithm to develop Lohmann detour-phase binary encoding diffractive elements from 2D

image matrix

Utilizing Lohmann detour phase encoding technique, the algorithm shown in Figure 70, a 1.7 GHz Pentium PC is used to calculate the Fourier transform of the required pattern. The input image is typically a 20×20 matrix of zeros with an imbedded 10x10 array of ones and zeros representing the desired image. A 200×200 two-dimensional discrete-Fourier-

transform is calculated and converted to a detour phase pattern written as a 16-bit bitmap file. I found that this system is able to produce the required output files in less than 1 second for a 20×20 input image matrix. The results had been tested for one period and for 6 periods of Lohmann binary encoded image. The optical setup used to test the simulation and testing results of an asymmetric object (letter "F") is shown in Figure 71 and Figure 72.
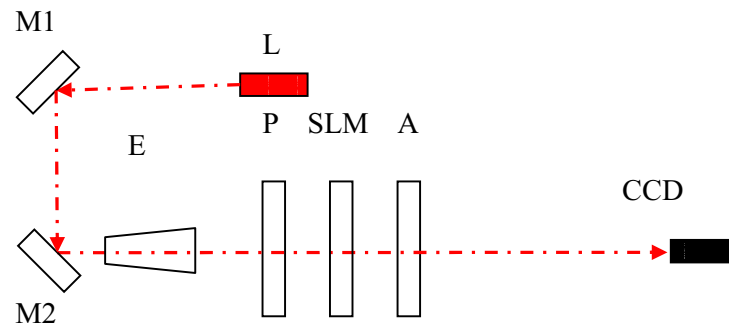
Figure 71

The utilized optical setup to test the Lohmann holograms. L=Laser source; E=Beam expander; P=Polarizer;

SLM=LC2002 SLM; A=Analyzer; CCD=CCD detector; f=focal length
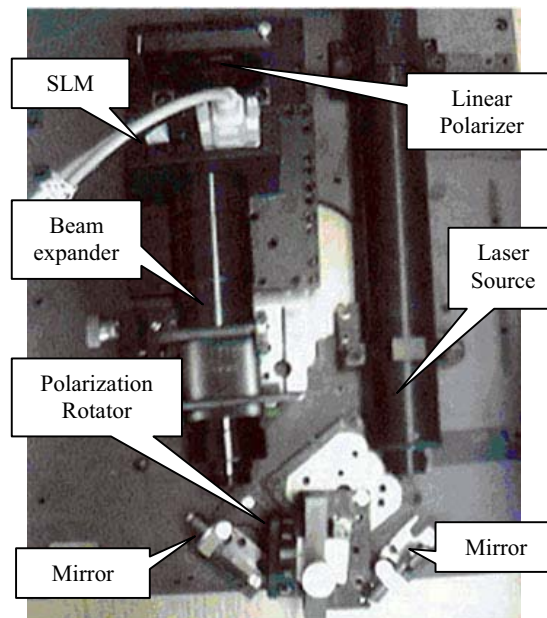
Figure 72

Optical setup for the utilized test-bed including a laser source, mirrors, polarization rotator, beam expander,

SLM and linear polarizer

We selected the ratio between the aperture width to the cell size to be 1/5 while the down-scale size to be 1/10. For a final aperture width of 2μm, the designed aperture parameter should be 2μm×5=10μm which the minimum width for the present technology limitations. Hence, the cell size is 10μm×15=150μm and with the down-scale it is 30μm. We have final phase freedom size of 28μm. Figure 73(a) shows the fabricated DOE while Figure 73(b) shows its reconstructed image.

In conclusion, the available 2μm microlithography technology allows the manufacturing of optical diffractive elements that are encoded using the detour phase technique in a suitable resolution and accuracy for several applications. This work had simulated, developed and optically tested some patterns utilizing a PC and SLM-based optical test-bed. The results are promising to start manufacturing the patterns for the targeted applications.

(a)   *(b)*
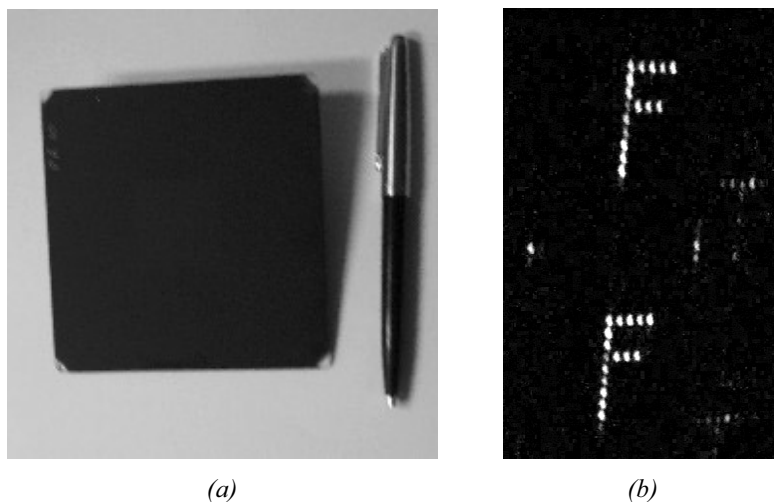
Figure 73

(a) The Fabricated DOE utilizing 1μm technology with an effective size of 30×30mm. (b) Reconstruction of letter F utilizing the Lohmann Detour-phase encoded DOE

B.3. Practicing with Direct Binary Search Technique

One more advanced technique, direct binary search (DBS), had been used to generate and optimize a CGH. The algorithm performs a single-pixel change (flipping from black to white or vice versa of a binary image) at each iteration and analyzes their effect on the

reconstruction. The simulation and experimental results for 128×128 binary image are shown in Figure 74.

(a)   *(b)*   *(c)*

(d)
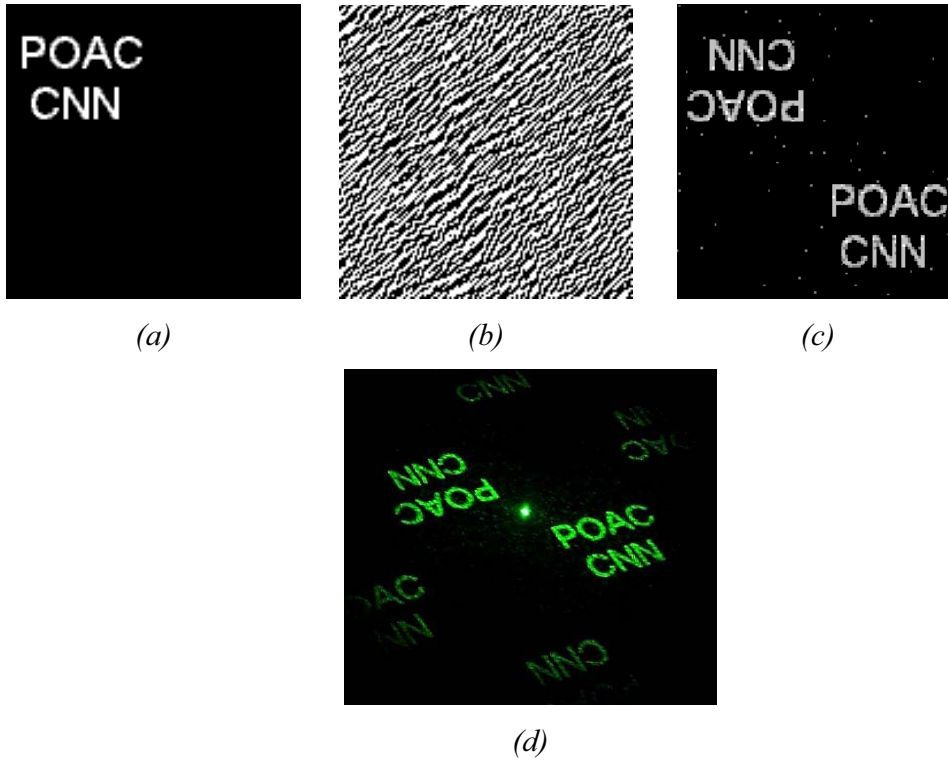
Figure 74

(a) Target 128×128 binary image (b) optimized binary hologram using DBS technique; (c) the simulated reconstruction of the twin image; and (d) the implemented reconstruction of the twin image using green laser

The implementation had utilized the same lithography technique described above. The resolution is 3µm while the hologram size is 15mm×15mm.

B.4. Discussion and Conclusion

This appendix demonstrated practical experiments to develop DOEs as a primary step in the proposed future research project described in chapter 7. Moreover, the present experiments are being connected with and used in one other running project for optical document security application.

Appendix C   Technical and Engineering Work: Acousto Optical Deflector (AOD) Control System

One early model of POAC is the semi-incoherent correlator, section 4.3.2, exploits an acousto optical deflector (AOD). Two AODs are used to display the input image to the optical setup. One is for horizontal shifting of the laser beam while the other is for the vertical shifting. The frequency range of the programmable clock oscillator (PCO) that is driving the ADO is 70-95 MHz that provides high speed to display the input image. The calibration cycle is 60ms while the output power is 2 W/50 ohm (0-10V). The input image size is up to 128×128 pixels. Figure 75 shows the real working system components while Figure 76 shows the front control panel of the system.

I had developed this system to be interfaced and controlled via personal computer software that communicates with through the Local Printer Terminal (LPT). It controls both PCOs as well as uploading the input image to be displayed to the associated microcontroller memory. Communication time had been optimized to reach the maximum frequency required for displaying a stable image, although it is still incoherent.
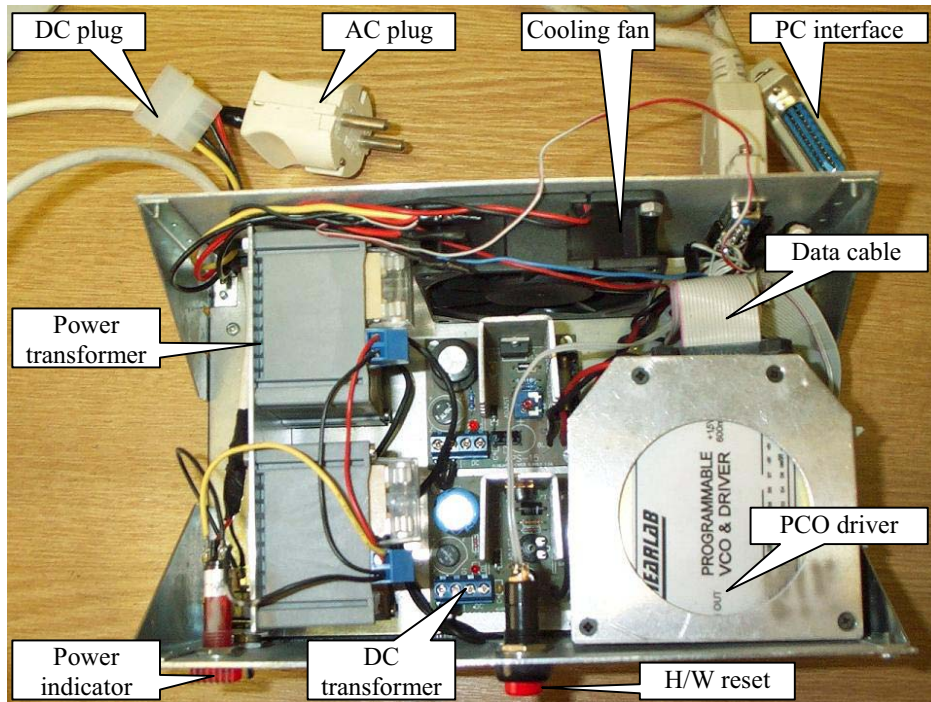
Figure 75

The AODs driving real working system components at a frequency range of 70-95MHz

The system starts up by powering on all of the peripherals before running the software controller. About 25 μsec are required to have a stable output. The image is to be chosen from the available library and uploaded to the microcontroller memory. Then the operating frequency and its raster are set. The image memory should be maintained by refreshing its contents by frequent uploads. If the system halts, the hardware reset button should be pressed and the software will smartly adapt the settings and the output once again.
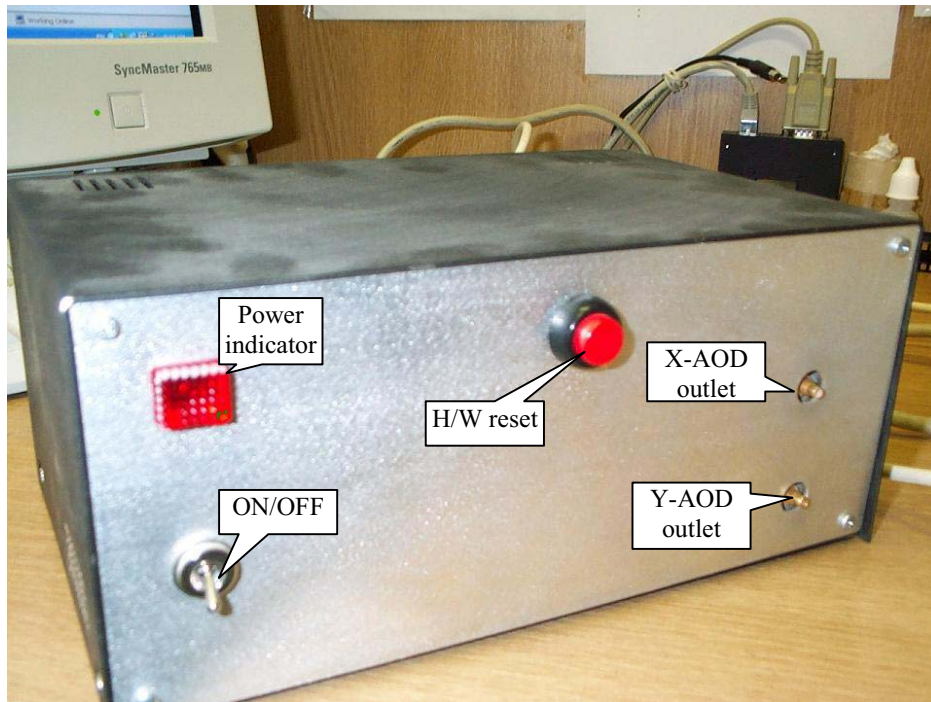
Figure 76

The AODs driving system control panel showing the X- and Y- channels to be connected to the AODs

This AOD driving system had been utilized in the semi-incoherent POAC model introduced in section 4.3.2. Presently, the experience of building this system is being utilized to develop similar one to by our group to control the incoherent mode of operation of the state of the art Laptop-POAC version.

Appendix D   Technical Specifications of Laptop POAC

Laser source:

Wavelength (nm)	Power (mW)	Beam Size (mm)	Beam Shape
650	35	1.0	Gaussian
532	20	1.0	Gaussian

Beam Expander: ×5 and ×10

Bacteriorhodopsin: 5000 line/mm; Decay time < 1 sec

Acousto optical deflector (AOD):
a.   frequency 1 μsec/pixel
b.   up to 64×64 pixels (grayscale)

Spatial light modulator (SLM):
a.   Resolution: 800×600 pixels
b.   Speed: 360 frame/sec.

Charged coupled device (CCD) detector:
a.   Resolution: 640×480 pixels at Frame rate 15Hz;
b.   Resolution: 320×240 pixels at Frame rate 30Hz;

Table 25

Laptop POAC Specifications

Property	Present Model
Image size [pixel]	500×500 (4 bit/pixel)
Image frame rate [Hz]	15
Template size [pixel]	32×32 or 64×64
Template frame rate [Hz]	1000 or 250
Equivalent processing power [Tera Ops]	0.3
Dimensions of POAC [cm]	29×34×18
Time of a template operation (32×32) [msec]	1
Cascadability	Yes
Power Dissipation	150 W
Possibility of complex valued Operations	No
Post processing by Xenon	No
Post processing by coherent optics	No
Erosion/Dilation (32×32) 1 bit	33 msec

REFERENCE

[1]   T. Roska and L. O. Chua, "*The CNN Universal Machine: An Analogic Array Computer*", IEEE Transactions on Circuits and Systems-II, vol. 40, pp. 163-173, March 1993

[2]   Szabolcs Tőkés, László Orzó, Csaba Rekeczky, Ákos Zarándy, Tamás Roska, "*Dennis Gabor as the Initiator of Optical Computing: Importance and Prospects of Optical Computing and an Optical Implementation of the CNN-UM Computer*", Dennis Gabor Memorial Conference and Symposium on Holography and High Resolution Measurement Techniques including Information Processing and Precision Measurement Techniques, 5-6th June, pp. 173-187, 2000

[3]   Alastair D. McAulay, "*Optical computer Architecture*", John Wiley & Sons, 1991

[4]   Szabolcs Tőkés, "*Optical Implementation of the CNN-UM Computer and its Realization with Joint Fourier Transform Correlator (JTC)*", Ph.D. Dissertation, MTA-SZTAKI, Budapest, 2000

[5]   L. O. Chua and L. Yang, "*Cellular Neural Networks: Theory and Applications*", IEEE Transactions on Circuits and Systems, Vol. 35, pp. 1257-1290, 1988

[6]   Tamás Roska, "*Towards a Visual Microprocessor*", John Wiley and Sons, 2001

[7]   Leon O. Chua and Tamás Roska, "*Cellular Neural Networks: Foundation and Primer*", Cambridge University Press, 2000

[8]   J. M.Cruz, L. O. Chua and T. Roska, "*A Fast, Complex and Efficient Test Implementation of the CNN Universal Machine*", Proc. of the third IEEE Int. Workshop on Cellular Neural Networks and their Application", CNNA-94, pp. 61-66, Rome, December 1994

[9]   Federico Bizzarri, Lorenzo Repetto, Marco Storace, Mauro Parodi, "*Towards a DSP implementation of an MRF-based cellular circuit for image segmentation and edge detection*", ECCTD'03, vol. 3, pp. 93-96, 2003

[10] Mario Salerno, Fausto Sargeni, Vincenzo Bonaiuto, Maurizio Bonifazi, "*Improved Architecture For SVCNN Systems*", ECCTD'03, vol. 3, pp. 101-104, 2003

[11] T. Hidvégi, I. Süto, P. Keresztes, P. Szolgay, "*Interfacing the CASTLE emulated digital array processor to the ALADDIN system*", ECCTD'03, vol. 3, pp. 265-268, 2003

[12] Suleyman Malki and Lambert Spaanenburg, "*CNN Image Processing on a Xilinx Virtex-II 6000*", ECCTD'03, vol. 3, pp. 261-264, 2003

[13] Ari Paasio, Jonne Poikonen, "*An Implementation Oriented Coding System for Nested CNNs*", ECCTD'03, vol. 3, pp. 269-272, 2003

[14] Tatsuro Shimada, Yosuke Takagi, Nobuaki Takahashi, Tsuyoshi Otake, Mamoru Tanaka, "*Data Mining for Time Sequence Prediction by Cellular Neural Network*", ECCTD'03, vol. 2, pp. 169-172, 2003

[15] Takanori Funahashi, Yoji Suzuki, Nobuaki Takahashi,Osamu Kakita, Tsuyoshi Otake, Mamoru Tanaka, "*Sparse analysis of cellular covariance structure for Data Mining*", ECCTD'03, vol. 2, pp. 173-176, 2003

[16] Zoltán Fodróczi, András Radványi, György Takács, "*Acoustic Source Localization Using Microphone Arrays Via CNN Algorithms*", ECCTD'03, vol. 2, pp. 177-180, 2003

[17] Zoltán Nagy, Péter Szolgay, "*Numerical solution of a class of PDEs by using emulated digital CNN-UM on FPGAs*", ECCTD'03, vol. 2, pp. 181-184, 2003

[18] Radu P. Matei, "*Oriented Low-Pass CNN Filters Based On Gaussian 1-D Prototype*", ECCTD'03, vol. 2, pp. 185-188, 2003

[19] Sz. Tőkés; L. Orzó; Gy. Váró and T. Roska, "*Bacteriorhodopsin as an Analog Holographic Memory for Joint Fourier Implementation of CNN Computers*", Research Report, MTA SZTAKI, DNS-3-2000, April 2000

[20] L. O. Chua, and T. Roska, "*The CNN Paradigm*", IEEE Trans. on Circuits and Systems, Vol. 40, pp.147-156, March 1993

[21] J. M. Cruz and L.O. Chua, "*A 16x16 Cellular Neural Network Universal Chip: the First Complete Single-Chip Dynamic Computer Array with Distributed Memory and with Gray-scale Input-output*", Analog Integrated Circuits and Signal Processing, Vol. 15, No. 3, pp. 227-238, 1998

[22] S. Espejo, R. Carmona, R. Domingúez-Castro, and A. Rodrigúez-Vázquez, "*CNN Universal Chip in CMOS Technology*", International Journal of Circuit Theory and Applications, Special Issue on CNN II: Part I, Vol. 24, pp. 93-111, 1996.

[23] Shuqun Zhang and Mohammad A. Karim, "*Cellular Neural Network Implementation Using a Phase-Only Joint Transform Correlator*", Opt. Com., Vol. 162, p.p. 31-36, 1999

[24] L. O. Chua, T. Roska, and P. L. Venetianer, "*The CNN is as Universal as the Turing Machine*", IEEE Trans. on Circuits and Systems, Vol. 40, pp.289-291, April 1993

[25] S. Espejo, R. Carmona, R. Domingúez-Castro, and A. Rodrigúez-Vázquez, "*CNN Universal Chip in CMOS Technology*", International Journal of Circuit Theory and Applications, Vol. 24, pp. 93-111, 1996

[26] T. Roska, P. Szolgay, A. Zarándy, P. Venetiáner, A. Radványi, and T. Szirányi, "*On a CNN Chip-Prototyping System*", in Proceedings of 3rd International Workshop on Cellular Neural Networks and Their Applications CNNA'94, pp. 375-381, Rome, 1994.

[27] S. Espejo, R. Domínguez-Castro, G. Liñán, Á. Rodríguez-Vázquez, "*A 64×64 CNN Universal Chip with Analog and Digital I/O*", in Proc. of 5th Int. Conf. on Electronics, Circuits and Systems ICECS'98, pp. 203-206, Lisbon, September 1998.

[28] R. Carmona, S. Espejo, R. Dominguez-Castro, A. Rodríguez-Vázquez, T. Roska, T.Kozek, and L. O. Chua, "*A 0.5 μm CMOS CNN Analog Random Acces Memory Chip for Massive Image Processing*", in Proceedings of 5th International Workshop on Cellular Neural Networks and Their Applications CNNA'98, pp. 271-281, London, 1998.

[29] Alireza Moini, "Vision chips or seeing silicon", technical report, The Centre for High Performance Integrated Technologies and Systems, Australia, 1997 (http://www.eleceng.adelaide.edu.au/Personal/moini/)

[30] Sherif S. Sherif, Stefan K. Griebel, Albert Au, Dennis Hui, Ted H. Szymanski and H. Scott Hinton, "*Field-Programmable Smart-Pixel Arrays: Design, VLSI*

Implementation, and Applications”, Applied Optics, Vol. 38, Issue 5, pp. 838-846, 1999

[31] J. F. Cantin and F. R. Beyette Jr., “*Design and Evaluation of a Field Programmable Smart Pixel Array with Programmable Photoreceivers*”, Optical Computing Conference, Lake Tahoe NV, pp. 102-104, January 2001

[32] T. H. Szymanski and H. S. Hinton, “*Architecture of Field Programmable Smart Pixel Arrays*”, in Proceedings of the International Conference on Optical Computing 94, Institute of Physics, Bristol, UK, Vol. 139, pp. 497–500, 1995

[33] J. Jahns, M. Gruber, S. Sinzinger, "*Planar Integrated Freespace Optics*", Joint German-Chinese Symp. Opto- & Microelectronic Devices and Circuits, Stuttgart, March 2002

[34] A. K. Ghosh, M. B. Lapis, D. Aossey, “*Planar Integration of Joint Transform Correlators*,” El. Lett., vol. 27, 871,1991

[35] W. Eckert, J. Jahns, S. Sinzinger, V. Arriz´on, “*Design and Fabrication of a Compact Planar-Integrated Optical Correlator*”, Proc. 10th Ann. Meeting of IEEE LEOS, San Francisco, paper MO5, 1997

[36] J. D. Joannopoulos, R. D. Meade and J. N. Winn, “*Photonic Crystals: Molding the Flow of Light*”, Princeton, 1995

[37] Steven G. Johnson and J. D. Joannopoulos, “*Introduction to Photonic Crystals: Bloch's Theorem, Band Diagrams, and Gaps (But No Defects)*”, Technical Report, MIT, February 2003

[38] Ovidiu Toader et. al., “*Photonic Band Gap Architectures for Holographic Lithography*”, Phys. Rev. Lett., Vol. 92, No. 4, pp. 043905-1, 2004

[39] M. Kamekura, Y. Seno, and H. Tomioka, “*Detection and Expression of a Gene Encoding a New Bacteriorhodopsin from an Extreme Halophile Strain HT (JCM 9743) which does not Possess Bacteriorhodopsin Activity*”, Extremophiles, Vol. 2, No. 1, pp. 33-40, January 1998.

[40] H. Luecke, H. T. Richter, J. K. Lanyi , “*Proton Transfer Pathways in Bacteriorhodopsin at 2.3 Angstrom Resolution*”, Science, Vol. 280, pp. 1934-1937, January 1998

[41] F. Gai, K. C. Hasson, J. C. McDonald, P. A. Anfinrud, "*Chemical Dynamics in Proteins: the Photoisomerization of Retinal in Bacteriorhodopsin*", Science, Vol. 279, pp. 1886-1891, March 1998

[42] Sz. Tőkés, L.R. Orzó, G. Váró, A. Dér, P. Ormos and T. Roska, "*Programmable Analogic Cellular Optical Computer using Bacteriorhodopsin as Analog Rewritable Image Memory*", NATO Book: "Bioelectronic Applications of Photochromic Pigments" (Eds.: L. Keszthelyi, J. Stuart and A. Der) IOS Press, Amsterdam, Netherlands, pp. 54-73, 2000

[43] C. S. Weaver and J. W. Goodman, "A Technique for Optically Convolving Two Functions", Applied Optics, vol. 5(7), pp. 1248-1249, July 1966

[44] James E. Rau, "Detection of Differences in Real Distribution", Optical Society of America, vol. 56(11), pp. 1490-1494, November 1966

[45] Sz. Tőkés, L. Orzó, Cs. Rekeczky, T. Roska and Á. Zarándy, "*An optical CNN implementation with stored programmability*", Proc. IEEE International Symposium on Circuits and Systems, Geneva, pp. II-136 – II-139, May 2000

[46] Sz. Tőkés, L. Orzó, L. Török, A. Ayoub, and T. Roska, "*An Advanced Joint Fourier Transform Correlator (JTC)*", European Optical Society, Topical Meetings Digest Series, vol. 30, pp. 128-129, Budapest, Hungary, October 2001

[47] Ahmed Ayoub, Szabolcs Tőkés, and László Orzó, "*A Study of Correlation Process of a Modified JTC for Optical CNN*", Proceedings of ECCTD'03 conference, vol. 3, pp.105, 2003

[48] Ahmed Ayoub, Szabolcs Tőkés, and László Orzó, "*Simulation of Optical CNN Template Library Based on t_2-JTC*", Proceedings of ECCTD'03 conference, vol. 3, pp. 257, 2003

[49] Szabolcs Tőkés, László Orzó and Ahmed Ayoub, "*Two-wavelength POAC (Programmable Opto-Electronic Analogic Computer) Using Bacteriorhodopsin as Dynamic Holographic Material*", Proceedings of ECCTD'03 conference, vol. 3, pp.97, 2003

[50] Szabolcs Tőkés, László Orzó, Ahmed Ayoub and Tamás Roska, "*Laptop POAC: a Compact Optical Implementation of CNN-UM*", Accepted by the CNNA2004, Budapest, 2004

[51] Ronald N. Bracewell, "*Two-Dimensional Imaging*", Prentice-Hall, 1995

[52] Ahmed Ayoub, Szabolcs Tőkés, and László Orzó, "*A Study and Simulation Results of Optical CNN Templates and Algorithms with a New Design for Implementation*", DNS-9-2002 Report, MTA-SZTAKI, Hungary, 2002

[53] "*CNN Software Library (Templates and Algorithms)*", version 1.1, MTA-SZTAKI, Budapest, 2001, http://lab.analogic.sztaki.hu

[54] Ju Jia Zou, et. al., "*Shape Skeletonization By Identifying Discrete Local Symmetries*", Pattern Recognition Society 34 1895-1905, 2001

[55] H. Fujiyoshi and A. Lipton, "*Real-time Human Motion Analysis by Image Skeletonization*, Proc. of the Workshop on Application of Computer Vision, October, 1998

[56] Victor A. Soifer, "*Methods for Computer Design of Diffractive Optical Elements*", John Wiley and Sons, 2002

[57] Bernard Kress and Patrick Meyrueis, "*Digital Diffractive Optics: An Introduction to Planar Diffractive Optics and Related Technology*", John Wiley and Sons, 2000

[58] Bernard and Kress and Patrick Meyrueis, "*Optical Diffractive Optics: An Introduction to Planar Diffractive Optics and Related Technology*", John Wiley and Sons, 2001

[59] A.W. Lohmann and D. Paris, "*Binary Fraunhofer Holograms, Generated by Computer*", Applied Optics, Vol. 6, No. 10, pp. 1739-1748, 1967

[60] B.R. Brown and A.W. Lohmann, "*Computer-Generated Binary Holograms*", IBM J. Res. and Dev. Vol. 13, pp. 160-168, 1969

[61] P.F. McKee et. al. , "*New Applications of Optics from Modern Computer Design Methods*", British Telecom Tech. J., Vol. 11, pp. 2159-2169, 1993

[62] B. Lesem and P.M. Hirch, "*The Kinoform: a New Wavefront Reconstruction Device*", IBM J. Res. and Dev., Vol. 13, pp. 160-168, 1969

[63] D.C. Chu, J.R. Fienup and J. W. Goodman ,"*Multiemulation On-Axis Computer Generated Hologram*", Applied Optics, Vol. 12, No. 7, pp. 1386-1388, 1973

[64] M. Goel and D.L. Naylor, "*Analysis of Design Strategies for Dammann Gratings*", SPIE Proc., Vol. 1689, pp. 35-45, 1996

[65] J.R. Fienup, "*Iterative Method Applied to Image Reconstruction and to Computer Generated Holograms*", Optical. Eng., Vol. 19, No. 3, pp. 297-305, 1980

[66] B.K. Jennison et. al., "*Iterative Approaches to Computer-Generated Holography*", Optical. Eng., Vol. 28, No. 6, pp. 629-637, 1989

[67] F. Wyrowski, "*Design Theory of Diffractive Elements in the Paraxial Domain*", J. Opt. Soc. Am. A, Vol. 10, No. 7, pp. 1553-1561, 1993

[68] R.W. Gerchberg and W.O. Saxton, "*A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures*", Optik. Vol. 35, No. 2, pp. 237-246, 1972

[69] X. Tan et. al., "Diffractive phase elements for beam shaping: a new design method", Appl. Opt., Vol. 34, No. 8, pp. 1314-1320, 1995

[70] G. Yang et. al., "*Gerchberg-Saxton and Yang-Gu Algorithms for Phase Retrieval in a Nounitary Transform System: a Comparison*", Appl. Opt., Vol. 33, No. 2, pp. 209-218, 1994

[71] B. R. Brown and A. W. Lohmann, "*Computer-Generated Binary Holograms*", IBM J. Res. Develop., 1969

[72] A. W. Lohmann and D. P. Paris, "*Binary Fraunhofer Holograms, Generated by Computer*", Applied Optics, vol. 6, pp. 1739–1748, 1967

[73] S. H. Lee, "*Topics in Applied Physics: Optical Information Processing Fundamentals*", vol. 48, Springer-Verlag, 1981

[74] Biedermann, K., "*Silver Halide Photographic Materials*", in Holographic Recording Materials, H. M. Smith, Ed. (Springer Berlin), 1977

[75] A. J. Lee and D. P. Casasent, "*Computer-Generated Hologram Recording Using a Laser Printer*", Applied Optics, vol. 26, pp. 136-138, 1987

[76] P. D. Fischer and S. Y. Chou, "*Sub-50 nm High Aspect Ratio Silicon Pillars, Ridges and Trenches Fabricated Using Ultrahigh Resolution Electron Beam Lithography and Reactive Ion Etching*", Applied Physics Letters, vol. 62, pp. 1414-1417, 1993

[77] M. Ekberg et. al., "*Proximity-Compensated Blazed Transmission Grating Manufacture with Direct-Writing Electron-Beam Lithography*", Applied Optics, vol. 33, pp. 103-107, 1994

[78] W. Däscher et. al., "*Fabrication of Monolithic Diffractive Optical Elements by the Use of E-Beam Direct Write on Analog Resist and a Single Chemically Assisted Ion-Beam-Etching Step*", Applied Optics, vol. 34, pp. 2534-2539, 1995

[79] Jari Turunen and Frank Wyrowski, "*Diffractive Optics for Industrial and Commercial Applications*", Akademie Verlag GmbH, 1997

INDEX

W

Z

TEMPLATE LIBRARY FOR PROGRAMMABLE OPTICAL ANALOGIC ARRAY COMPUTER (POAC)

PROGRAMMING OF OPTICAL CELLULAR NEURAL NETWORK (CNN)

Theses of the Ph.D. Dissertation

Ahmed Ayoub

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

Scientific advisor
Szabolcs Tőkés, Ph.D.

Budapest, 2004

"Everything should be made as simple as possible, but not simpler."

Albert Einstein

C O N T E N T S

Introduction

It seems that with the fast progress of digital technology the analog solutions are losing field. However, some questions remained – e.g. where real-time processing of sensor-array inputs, like in artificial retina, is required -, that could not be solved with sequential digital computing in the foreseeable future.

The proven limitations in pure digital computing motivated an intensive research on the field of neural/nonlinear parallel processing and the fabrication of the first hardware prototypes. However, contrary to all expectations, this did not lead to a breakthrough of a new technology. The reason for that was not necessarily the well-known bottleneck, the limited precision of all analog hardware implementations, but other factors. The spread and utilization of these novel computing tools in industrial applications were even more limited by the imperfection of the architectures, to mention the most important: quick reprogramming of these new computing structures was not possible, restricting their use to very specific tasks. Furthermore, the approach assuming a fully connected processing net turned out to be a major obstacle from the realization point of view, since the implementation complexity increases exponentially with the number of processors.

Computing architectures based on the cellular neural/nonlinear network (CNN) paradigm offer an adequate solution for the problems above. CNNs are regular, single or multi-layer, parallel processing structures with analog nonlinear computing units (base cells). The state values of the individual processors are continuous in time and their connectivity is local in space. The program of these networks is completely determined by the pattern of the local interactions, the so-called template, and the local logic and arithmetic (analog) instructions. The time-evolution of the analog transient, driven by the template operator and the processor dynamics, represents the computation in CNN. Results can be defined both in equilibrium and/or non-equilibrium states of the network. Completing the base cells of CNN with local sensors, local data memories, arithmetical and logical units, furthermore with global program memories and control units results in the CNN universal machine (CNN-UM) architecture. The CNN-UM is an analogic (analog and logic) supercomputer, it is universal both in Turing sense as a nonlinear operator, therefore it can be used as a general architectural framework when designing CNN processors. Up to the present there have been various physical implementations of this architecture: mixed-signal VLSI, emulated digital VLSI and optical.

The Programmable Optical Array/Analogic Computer (POAC) is an optical implementation of CNN. It is based on coherent and/or incoherent modified Joint Fourier Transform Correlator (t_2-JTC). Its high capabilities, specifically full parallelism and large array size, provided by the physics of light makes it a candidate for achieving high frame rate of processing.

In this dissertation the focus is put on understanding the present architectures of POAC and find new methods to enhance their functions. Moreover, the effort is concentrated on finding and modeling the first optical template library for several optical processing tasks.

Computer simulations are made for different CNN optical computer architectures, some of them had been investigated experimentally. The proposed optical computing system is to accomplish image-processing algorithms, including image classification and recognition tasks.

Methods Used in the Experiments

In the course of my work, theorems and methods from the field of digital computing hardware and software architectures, image processing techniques, methodologies related to neurological modeling and optical computing system architectures were explored and utilized. In addition, I studied in details the present optical correlator implementations and their theoretical background.

I studied the present CNN template library and template design strategies. Based on these studies, I deduced an empirical equation to transform the traditional feed-forward CNN template values into optical ones. Then, I proved via a new coherent optical technique that negative values can be implemented in optics. Finally, I created through simulation the first optical CNN template library.

I utilized several software tools to design and develop the optical CNN simulator. The first of it was the CNN simulator (A*lladin*), developed by the Analogic and Neural Computing Systems Laboratory of MTA-SZTAKI, as a fundamental tool to study the CNN template library tasks. The second was the MATLAB (release 12) that I used to create the optical CNN simulator. The third was a custom program that I created to design computer generated holograms (CGHs) suitable for fabrication via photolithographic methods. The fourth was the custom software, MaskEditor v.1.6, developed for internal use only to handle the design process of CGHs fabrication.

In addition to the software tools, I also had built an optical hardware setup to design, create and experiment with CGHs that were necessary during my studies with optics. Spatial light modulators (SLMs) had been tested to work in phase-only or amplitude-only mode of operation through my experiments with CGH photolithographic fabrication.

New Scientific Results

Thesis 1: I showed the essential differences between optical coherent and incoherent correlations. I described, analyzed and proposed with proof how they can be utilized for optical CNN implementation.

I have shown that the optical CNN implementation performs better in coherent mode of the modified Joint Fourier Transform Correlator (t_2-JTC) than in the incoherent one. I have studied the differences between the two modes of operation and how to implement templates in these architectures. Finally I worked out fast post-processing procedures to enhance the discrimination capability.
Published in JCSC, ECCTD03

1.1. I have defined models and developed algorithms by which optical coherent and incoherent correlation is being performed and found the differences between these modes of operations.

The *time-multiplexed* incoherent correlation requires photoelectric accumulation to integrate the images needed for correlation. I showed via numerical simulation that the incoherent correlation is immune to phase noise and is simple to be realized. However, I pointed out that coherent correlation has higher contrast than the incoherent one. (*see section 5.2.*) That is why in coherent mode the thresholding can be more effectively implemented. (*see section 5.2.*)

1.2. I showed via simulation and experimentation that the need of *center-mirroring* is dependent on the actual architecture of the optical correlator to achieve correlation. I gave a prescription table for how to use it in different types of optical implementations with special regards to our t_2-JTC. (*see section 5.1.*)

These results have been successfully verified by experiments in different architectures.

1.3. I designed fast post-processing *Mexican-hat functions* with selected optimum parameters to yield correlograms that can be simply thresholded.

My simulation and experimental results show that the complexity of adaptive thresholding is avoided (*see section 5.4.*). The size of the template and its values are crucial to properly eliminate the background of the peaks. My functions improved the signal to noise ratio (SNR) considerably.

Thesis 2: I proved that the optical implementation of negative template values is practically possible by applying a phase shift of value π to such values.

It was an unsolved crucial question to optically implement negative template values. I had defined the problem and proposed a practical solution to achieve negative value effect in optics.

The nature of light does not provide negative values, if one is dealing with intensity only. According to my solution to perform negative template value, one has to shift the phase of this template element (pixel) by value of π (*see section 5.3.*). I verified the proposed technique by digital simulation. I proposed that cascaded *phase-only* and *amplitude-only* spatial light modulators (SLMs) have to be placed in the template arm of the programmable optical array computer (POAC) and designed its architecture. The first SLM has to be imaged onto the second one and commonly Fourier transformed by a lens, (*see section 4.3.3.*). This solution requires the coherent mode of operation.
Published in JCSC, ECCTD03

Thesis 3: I developed the first optical image processing template library of operations and algorithms to be implemented by the programmable optical analogic array computer (POAC).

My specific task in this Ph.D. research work was to find, design and test via simulation the availability of an optical template library for POAC. (*see chapter 6.*) Hence, find *optical programs* for the "Programmable Optical Analogic Array Computer" (POAC). So far, eighteen optical feed-forward templates had been designed, as well as, one complex algorithm.
Published in ECCTD03, IEEE MWSCS, IEEE CNNA2004

Starting from the well-established CNN template library and based on the state of the art POAC architecture, I developed a simulator for feedforward optical CNN (*see section 6.1.*) I designed the first optical template library for POAC that consists of 18 templates; they are utilizing my results in *thesis 2*. Some of them can be simply modified and extended (e.g. rotating Southern element detector template results in East, West and North element detectors) (*see section 6.2.*). I developed, via simulation, a macro algorithm to design cascaded templates to solve complex optical skeletonization task (*see section 6.3.*).
It has been shown that for proper implementation of the library, coherent mode of operation and the architecture designed for negative template values (see Thesis 2) is required.

New Practical Results

- The main task of my Ph.D. studies was to find, design and test via simulation the fundamental optical operations for the POAC architectures having been in development. So, my studies' results had been utilized in the design and implementation of the state of the art optical computer (Laptop-POAC) (*see section 4.3.5.*).

- I developed practical guidelines to perform optical correlation (both for incoherent and coherent ones).

- Fast post-processing functions have been developed for better thresholding.

- I showed via simulation how a negative template value can be implemented in optics and proposed its practical implementation.

- I made available the first optical template CNN library, 18 templates, for the Programmable Optical Analogic Array Computer (POAC)

- I created one optical algorithm that, if implemented, it will take about 5 milliseconds to perform skeletonization over an input image, greater than a video frame rate.

Application of the Results

The main application field of the results achieved in this work is to be utilized as *micro operations* for POAC. The high capabilities of POAC are in need for programs. The templates and their library that I had developed in this work are leading to the standardization and can be used to commercialize the system.

The most promising application areas seems to be the *image understanding* and processing tasks where the use of an optical computer is reasonable and fast. Optical computer implementations are superior to their digital signal processing counterparts especially in those tasks, where great numbers of correlations have to be completed with large templates on high-resolution images

VanderLugt type, optical filtering based correlators are fast, due to their inherent massive parallelism. However, because of the time consuming process of matched filter design, they are not applicable when the task requires fast (re-) programming. As there is no need of any matched filter calculation in the advanced t_2-JTC architecture, data gained from the previous image frames become applicable.

Other applications are suggested, like *stereo image processing*, *motion estimation for image compression* or *diverse correlation based identification algorithms*, where not only fast programmable optical correlation is necessary, but local parallel pre- and post-processing (CNN) is also required.

In addition to what is mentioned above, the following are also recommended applications:

- Multiple target recognition and tracking including collision avoidance;
- Security (document security and face recognition);
- Database mining (mainly optical searching for fingerprints or image archives databases);
- Traditional image processing and pattern recognition tasks;
- Industrial quality control and recognition of assembly parts;
- Complex-valued functions operations with large capacity (e.g. 1000×1000 complex numbers);
- Microscopic recognition, tracking and guiding of laser-tweezer-like manipulation.

Acknowledgments

The Author's Publications

In print

  Szabolcs Tőkés, László Orzó, **Ahmed Ayoub**, and Tamás Roska, "*POAC (Programmable Optical Array/Analogic Computer) Applied for Target Recognition and Tracking*", Accepted, proceedings of SPIE European Symposium on Optics/Photonics in Security and Defense, London, October 2004

2004

  Ahmed Ayoub, Szabolcs Tőkés, László Orzó and Tamás Roska, "*Evolution of the Programmable Optical Array Computer (POAC)*", in proceedings of IEEE CNNA2004, Hungary, pp. 64-69, 2004

  Szabolcs Tőkés, László Orzó, **Ahmed Ayoub** and Tamás Roska, "*Laptop POAC: A Compact Optical Implementation of CNN-UM* ", in proceedings of IEEE CNNA2004, Hungary, pp. 70-75 , 2004

  Ahmed Ayoub, Szabolcs Tőkés and László Orzó, "*Review of the Programmable Optical Array Computer*", Technical Report, DNS-2-2004, MTA-SZTAKI, Hungary, 2004

  Ahmed Ayoub and Szabolcs Tőkés, "*Literature Collection of the Programmable Optical Array/Analogic Computer (POAC)*", Technical Report, MTA-SZTAKI, July 2004.

2003

  Szabolcs Tőkés, László Orzó, **Ahmed Ayoub**, and Tamás Roska, "*Flexibly Programmable Opto-electronic Analogic CNN Computer (POAC) Implementation Applying an Efficient, Unconventional Optical Correlator Architecture*", Journal of Circuits, Systems, and Computers, Vol. 12, No. 6, pp. 739-767, December 2003

  Szabolcs Tőkés, László Orzó and **Ahmed Ayoub**, "*Two-wavelength POAC (Programmable Opto-electronic Analogic Computer) using Bacteriorhodopsin as dynamic holographic material*", in proceedings of ECCTD'03 conference, vol. 3, p.p.97, 2003

  Ahmed Ayoub, Szabolcs Tőkés, and László Orzó, "*A Study of Correlation Process of a Modified JTC for Optical CNN*", in proceedings of ECCTD'03 conference, vol. 3, p.p.105, 2003

  Ahmed Ayoub, Szabolcs Tőkés, and László Orzó, "*Simulation of Optical CNN Template Library Based on t_2-JTC*", in proceedings of ECCTD'03 conference, vol. 3, p.p.257, 2003

  Ahmed Ayoub and Szabolcs Tőkés, "*The First Version of the Optical Programmable Array/Analogic Computer (POAC) Template Library*", in proceedings of The 46th IEEE Midwest Symposium on Circuits & Systems, Cairo, Egypt, December 2003

2002

  Ahmed Ayoub, Szabolcs Tőkés, and László Orzó, "*A Study and Simulation Results of Optical CNN Templates and Algorithms with a New Design for Implementation*", Technical Report, DNS-9-2002, MTA-SZTAKI, Hungary, 2002

2001

  Szabolcs Tőkés, László Orzó, Levente Török, **Ahmed Ayoub**, and Tamás Roska, "*An Advanced Joint Fourier Transform Correlator (JTC)*", Diffractive optics tropical meeting, Budapest-Hungary, 2001

2000

  Ahmed Ayoub and Medhat M. El Messeiry, "*A Study of The Biothermal Effects that Lead to Pressure Ulcer with a New Design for Prevention*", November 2000, The International Congress on Environment Health, Issues in Primary Health Care, Toxicology, Biology & Social Issues., Cairo-Egypt, November 2000

The Author's Awards Related to Topic of Dissertation

- *Best Doctoral Research Work Prize for Second-Year Students*, Hungarian Academy of Sciences, Computer and Automation Research Institute, Budapest, November 2003

- *CNN Young Researcher Contest, Analogic CNN Algorithm Design*, 7th IEEE International Workshop on Cellular Neural Networks and Their Applications, Frankfurt, July 2002

Selected Publications on Topics Related to Dissertation

L. O. Chua and L. Yang, "*Cellular Neural Networks: Theory and Applications*", IEEE Transactions on Circuits and Systems, Vol. 35, pp. 1257-1290, 1988

T. Roska and L. O. Chua, "*The CNN Universal Machine: An Analogic Array Computer*", IEEE Transactions on Circuits and Systems-II, vol. 40, pp. 163-173, March 1993

Tamás Roska, "*Towards a Visual Microprocessor*", John Wiley and Sons, 2001

Leon O. Chua and Tamás Roska, "*Cellular Neural Networks: Foundation and Primer*", Cambridge University Press, 2000

L. O. Chua, and T. Roska, "*The CNN Paradigm*", IEEE Trans. on Circuits and Systems, Vol. 40, pp.147-156, March 1993

"*CNN Software Library (Templates and Algorithms)*", version 1.1, MTA-SZTAKI, Budapest, 2001, http://lab.analogic.sztaki.hu

Szabolcs Tőkés, "*Optical Implementation of the CNN-UM Computer and its Realization with Joint Fourier Transform Correlator (JTC)*", Ph.D. Dissertation, MTA-SZTAKI, Budapest, 2000

C. S. Weaver and J. W. Goodman, "*A Technique for Optically Convolving Two Functions*", Applied Optics, vol. 5(7), pp. 1248-1249, July 1966

James E. Rau, "*Detection of Differences in Real Distribution*", Optical Society of America, vol. 56(11), pp. 1490-1494, November 1966

Mohammed S. Alam, "*Selected Papers on Optical Pattern Recognition Using Joint Fourier Transform Correlation*", SPIE Milestone series, Vol. MS 157, 1999

Ronald N. Bracewell, "*Two-Dimensional Imaging*", Prentice-Hall, 1995

Jari Turunen and Frank Wyrowski, "Diffractive Optics for Industrial and Commercial Applications", Akademie Verlag, Berlin, 1997

A.W. Lohmann and D. Paris, "*Binary Fraunhofer Holograms, Generated by Computer*", Applied Optics, Vol. 6, No. 10, pp. 1739-1748, 1967

Bernard Kress and Patrick Meyrueis, "*Digital Diffractive Optics: An Introduction to Planar Diffractive Optics and Related Technology*", John Wiley and Sons, 2000

Victor A. Soifer, "*Methods for Computer Design of Diffractive Optical Elements*", John Wiley and Sons, 2002

CURRICULUM VITA

AHMED AYOUB

ayoub@sztaki.hu

MAY 2004

Ahmed E. Ayoub was born in Egypt where he received his B.Sc. degree in Computer Engineering 1994 and B.Sc. in Physics 1997. He got his M.Sc. degree in biophysics engineering from Cairo University in 2000. During his Ph.D. studying period, he had been awarded two prizes. The first was for the cellular neural networks (CNNs) young researcher contest for analogic CNN algorithm design during the 7th IEEE international workshop on CNNs and their Applications (Frankfurt 2002). The second was the best doctoral research work prize for second-year students from the Hungarian academy of sciences in November 2003. Presently, he works and studies at the Analogic and Neur85+al Computing Laboratory, Computer and Automation Research Institute of the Hungarian Academy of Sciences. His interests of research are: optical computing, optical CNNs implementations and applications and Diffractive optical processing (DOPs).

PERSONAL INFORMATION

Name: Ahmed El Sayed Ayoub
Date of Birth: December 1, 1972
Nationality: Egyptian

Qualifications

- **M.Sc. in Biophysics Engineering**, Engineering Mathematics and Physics Department, Faculty of Engineering, Cairo University, Egypt, 2000
- **B.Sc. in Physics**, Department of Physics, Banha Faculty of Sciences, Zagazig University, Egypt, 1997
- **B.Sc. in Computer Engineering**, Department of Industrial Electronics, Banha Institute of Technology, Egypt, 1994

Contact Address

Hungarian Academy of Sciences
Computer and Automation Research Institute
Analogical and Neural Computing Systems Laboratory
1111 Kende u. 13-17, 413
Budapest, Hungary

Tel.: (361) 279 6135
Fax.: (361) 209 5264
Email: ayoub@sztaki.hu
Internet: http://virgo.analogic.sztaki.hu/~ayoub/

RESEARCH INTERESTS

- Cellular Neural Networks (CNNs) Applications
- Optical Computing
- Optical CNNs
- Bio-Inspired Photonic Applications

AWARDS
- Best Doctoral Research Work Prize for Second-Year Students, Hungarian Academy of Sciences, Computer and Automation Research Institute , November 2003
- CNN Young Researcher Contest, Analogic CNN Algorithm Design, 7th IEEE International Workshop on Cellular Neural Networks and their Applications, Frankfurt-Germany, July 2002 (3rd Place)