RL-TR-97-122
In-House Report
October 1997

ADAPTIVE OPTICAL NEURAL NETWORK CLASSIFIER SYSTEMS

Wesley E. Foor, Mark A. Getbehead, and James B. Rosetti

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

RL-TR-97-122 has been reviewed and is approved for publication.

APPROVED: Gary D. Barmore

GARY D. BARMORE, Maj., USAF
Chief, Photonics Division
Surveillance and Photonics Directorate

FOR THE DIRECTOR: Donald W. Hanson

DONALD W. HANSON, Director
Surveillance and Photonics Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 1997

3. REPORT TYPE AND DATES COVERED
FINAL, Oct 93 - Sep 96

4. TITLE AND SUBTITLE
ADAPTIVE OPTICAL NEURAL NETWORK CLASSIFIER SYSTEMS

5. FUNDING NUMBERS

C -	**NA**
PE -	**62702F**
PR -	**4600**
TA -	**P1**
WU -	**13**

6. AUTHOR(S)
Wesley E. Foor, Mark A. Getbehead, James B. Rosetti

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory/OCPA
25 Electronic Pky
Rome NY 13441-4515

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TR-97-122

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory/OCPA
25 Electronic Pky
Rome NY 13441-4515

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
RL-TR-97-122

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Wesley E. Foor, OCPA, (315) 330-3261

12a. DISTRIBUTION AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

We present two adaptive opto-electronic neural network hardware architectures capable of exploiting parallel optics to realize real-time processing and classification of high-dimensional data. The architectures are based on radial basis function neural networks that employ on-line training techniques to offer robustness to noise and optical system imperfections. A binary-input data classifier is presented first and the issues of system imperfections, device characterization, and system noise are addressed. The experimental results from the optical system are compared with data from a computer model of the system in order to identify critical noise sources and to indicate possible areas for system performance improvements. A grayscale-input classifier is then proposed for handling 8 bit input data to broaden the range of applications of the classifier. An optical wavelet transform system intended for use as a multi-resolution image preprocessor for the classifiers is then presented and discussed.

14. SUBJECT TERMS
Optical neural networks, radial basis functions, pattern recognition, optical wavelet transform, multiscale image processing

15. NUMBER OF PAGES
52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UNLIMITED**

Contents

1 INTRODUCTION 1

2 BINARY INPUT NEURAL NETWORK CLASSIFIER 3

2.1 Radial Basis Function Neural Networks 3

2.2 Handwritten Digit Recognition 5

2.3 Optical RBF Neural Network 5

2.3.1 Parallel Optical Distance Computation 6

2.3.2 Parallel Basis Function Evaluation 7

2.4 Noise Characterization and Computer Modeling 8

2.4.1 Optical System Noise Characteristics 9

2.4.2 Optical System Model 12

2.5 Adaptive Optical Radial Basis Function Neural Network 12

2.5.1 Adaptive Postprocessor 13

2.5.2 Influences of Noise on System Performance 13

2.5.3 Adaptive Optical RBF Neural Network Results 17

3 GRAYSCALE INPUT NEURAL NETWORK CLASSIFIER 18

3.1 Optical Image Subtraction 18

3.2 Variable Retarder Characterization 19

3.2.1 Operation 19

3.2.2 Procedure 21

3.2.3 Results 23

4 WAVELET PREPROCESSING **25**

4.1 Conventional Joint Transform Correlator 25

4.2 Multiple-input Joint Transform Correlator 27

4.2.1 Theory 27

4.2.2 Multispectral example using wavelets 29

4.3 Simulation Experiments 30

4.3.1 Implementation in Optics 33

4.4 Discussion 35

5 CONCLUSIONS **36**

REFERENCES 38

List of Figures

2.1 Schematic diagram of RBF neural network for digit recognition. 4

2.2 Dual-rail vector-matrix multiplier used as a parallel optical distance computer. . . . 6

2.3 Euclidean distance vs. template number for a single input image (handwritten 4). a) Actual distance and b) optically computed distance. 7

2.4 Non-adaptive RBF postprocessing chip for a single-output network. 9

2.5 Measured and modeled LCLV transfer functions. 11

2.6 Adaptive RBF postprocessing chip module. 14

2.7 Predicted recognition rate versus imaging axis beam profile. Circles indicate isolated error model, squares indicate multiple error model, solid curves indicate adaptive case and dashed curves indicate non-adaptive case. 14

2.8 Predicted recognition rate versus collimating axis beam profile. Symbols are as defined in Figure 2.7. 15

2.9 Predicted recognition rate versus LCLV readbeam profile. Symbols are as defined in Figure 2.7. 15

2.10 Predicted recognition rate versus leg intensity ratio. Symbols are as defined in Figure 2.7. 16

3.1 Optical image subtractor using phase-shifter. 19

3.2 Optical image subtractor used as a grayscale-input parallel distance computer. . . . 20

3.3 Retarder used as an attenuator. 20

3.4 Variable retarder characterization test setup. 22

3.5 Alignment steps for variable retarder characterization. 23

3.6 Final results of variable retarder characterization. 24

4.1 Schematic diagram of a conventional joint-transform correlator. 26

4.2 Input and output planes of a JTC showing locations of inputs and outputs (a) input plane (b) output plane. 27

4.3 Input plane of multiple-input JTC. 28

4.4 Output plane of multiple-input JTC. 28

4.5 Input plane of multispectral example. 29

4.6 Output plane of multispectral example. 30

4.7 Image used in simulation experiments. 31

4.8 Output plane of multispectral simulation experiment. 31

4.9 Close-up view of right hand side of Figure 4.8. Response on right is for $a_2 = 1.2$, response of left is for $a_1 + a_2$, where $a_1 = 0.7$. 32

4.10 Response on right is for $a_2 = 1.2$, response of left is for $a_1 + a_2$, where $a_1 = 1.0$. . . . 33

4.11 Response when input image was rotated by 90 degrees. Response on right is for $a_2 = 1.2$, response of left is for $a_1 + a_2$, where $a_1 = 1.0$. 34

4.12 Schematic diagram of multiple-input JTC used for multispectral analysis using two wavelet scales. 34

List of Tables

2.1 Performance comparison between optical RBF classifier and software RBF classifier. 8

2.2 Measured error values from optical system. 12

2.3 Performance comparison between optical and modeled optical RBF classifier. 13

2.4 Performance comparison between adaptive optical RBF classifier and the modeled adaptive optical RBF classifier. 17

1

INTRODUCTION

Adaptive neural networks are considered to be promising architectures for real-time pattern recognition. Due to the large network sizes required for real-world problems involving imagery, these adaptive networks will be most useful for solving real-time problems if they are implemented in parallel hardware. The parallel processing capabilities of opto-electronic systems together with the relatively simple computational requirements of artificial neural networks make optics a natural candidate for hardware implementations of neural computing systems. Optical computing systems however, typically suffer from optical device imperfections and system noise that can degrade performance. By employing adaptive on-line training techniques these noise sources can be incorporated into an error-driven learning process to provide improved system performance. Neural networks therefore offer an opportunity to realize parallel optical computing systems that tolerate noise. In this report we describe two adaptive neural architectures capable of exploiting parallel optical hardware for multi-dimensional signal classification.

The radial basis function (RBF) neural network has been successfully used in many multi-dimensional classification applications including 3D object recognition [1, 2], radar signal classification [3], face recognition [4, 5], fingerprint recognition [6], speech recognition [7], and handwritten character recognition [8, 9, 10]. Other applications of RBF neural networks include nonlinear function approximation [11, 12, 1], kernel regression [13], 'neural beamforming' for phased-array antennas [14, 15, 16, 17], equalization of time-dispersive communication channels, and nonlinear modeling and prediction for echo cancellation in the presence of nonlinear distortion [18]. Previous experiments have shown that RBF networks have similar classification performance to backpropagation neural networks while typically incurring shorter training times [8]. Both all-electronic [19] and opto-electronic [20, 10] parallel hardware implementations of RBF networks have been reported.

In chapter 2 of this report we begin by experimentally demonstrating an adaptive optical RBF classifier that facilitates on-line learning to offer robustness to noise and optical system imperfections. The binary-input system performance is evaluated in the application of handwritten digit classification. The issues of system imperfections, device characterization, and system noise are discussed. The experimental results from our optical system are compared with data from a computer model of the system in order to identify critical noise sources and to indicate possible areas for system performance improvements. In chapter 3 we propose a grayscale-input data classifier for use with non-binary (e.g., 8 bit or analog) signals. The proposed implementation replaces the first layer hardware of the binary-input system with optical image subtraction optics that permits a true Euclidean distance measure to be computed. In chapter 4 we describe our ongoing work in the development of a multiscale optical wavelet transform system intended for use as an image preprocessor for the classifiers. It should be noted that Dr. Samuel P. Kozaitis from Florida Institute of Technology (FIT) has played a key role in our optical correlator and optical wavelet research and development. Conclusions and future directions of this work are then presented in chapter 5.

2

BINARY INPUT NEURAL NETWORK CLASSIFIER

In this chapter we present a binary-input data optical radial basis function neural network classifier. Many classifier applications either use binary data directly or may be converted to a binary data format with acceptable degradation including: optical character recognition, fingerprint identification, (edge-enhanced) template matching [21], and certain disparity computations. More details of this binary-input system design and its application to handwritten digit recognition can be found in earlier publications [10, 22].

2.1 Radial Basis Function Neural Networks

A single-output multilayer neural network can be regarded as a continuous input-output mapping, $\Re^N \rightarrow \Re$. The network should behave well in the presence of noise and correctly generalize when a previously unseen input pattern is presented. To obtain these characteristics we may impose smoothing constraints on the input-output mapping as derived from regularization techniques and approximation theory.[1] Approximation theory attempts to provide an optimal solution to approximating a continuous multivariate function $f(\mathbf{x})$, with an *approximating function* $\hat{f}(\mathbf{w},\mathbf{x})$, where $\mathbf{x}$ is a N dimensional input vector and $\mathbf{w}$ is a parameter vector used to minimize the approximation error. The set of M input vectors comprise the training set on which the desired input-output mapping, $\{\mathbf{x}_i \rightarrow f(\mathbf{x}_i); i = 1, \ldots, M\}$, is defined.

The RBF approximation scheme arises when certain symmetry assumptions are made about the smoothing constraints utilized in the regularized solution and corresponds to an approximating function of the form

$$\hat{f}(\mathbf{w},\mathbf{x}) = \sum_{i=1}^{M} a_i \exp\left(\frac{-|\mathbf{x} - \mathbf{t}^i|^2}{\sigma_i^2}\right), \tag{2.1}$$

where $\{\mathbf{t}^i\}$ is a set of center locations, $\{\sigma_i\}$ are the corresponding center widths, and $\{a_i\}$ are a set of weighting factors. These three sets define the parameters of $\mathbf{w}$.

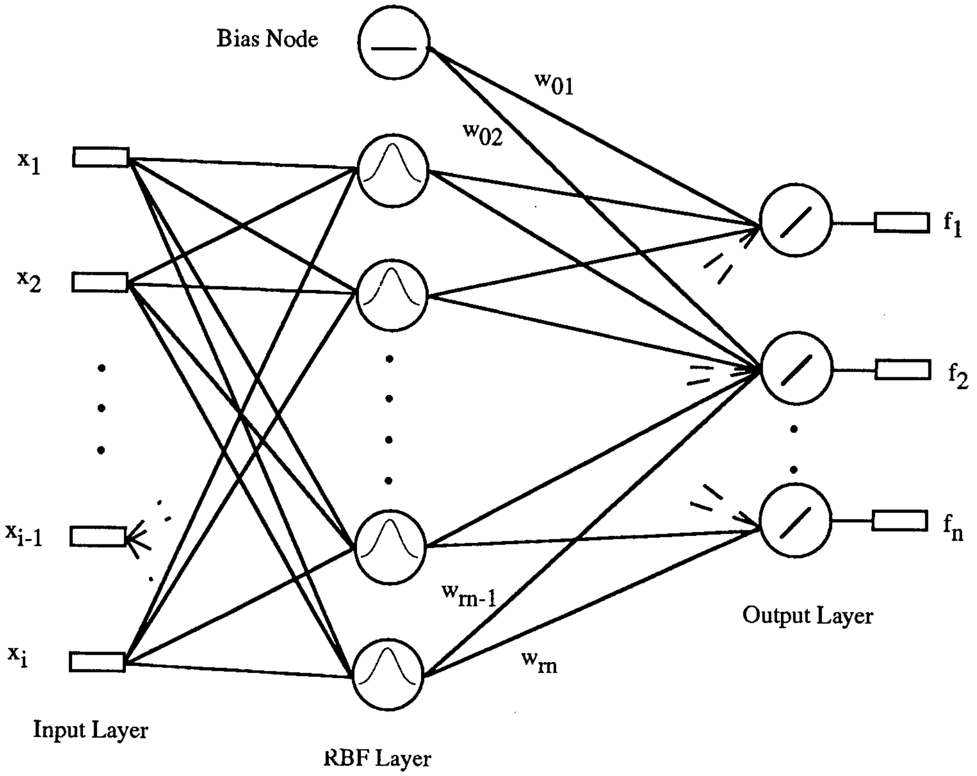

Figure 2.1: Schematic diagram of RBF neural network for digit recognition.

The approximation function given above can be represented in the form of a one-layer neural network as shown in Figure 2.1. The hidden layer RBF node response is given by

$$y_i = \exp\left(\frac{-|\mathbf{x}-\mathbf{t}^i|^2}{\sigma_i^2}\right), \tag{2.2}$$

where $\mathbf{t}^i$ is the neuron center and σ_i is the neuron width. In contrast to conventional networks, which use inner products, the first layer requires the calculation of a Euclidean distance. In addition, the neuron response is not sigmoidal but corresponds to a $\exp\left(-x^2/\sigma^2\right)$ function. For a multiple-output RBF network, the k^{th} output node response to an input vector $\mathbf{x}$ is given by

$$f_k(\mathbf{x}) = \sum_{i=1}^{M} a_{ik} \exp\left(\frac{-|\mathbf{x}-\mathbf{t}^i|^2}{\sigma_i^2}\right), \tag{2.3}$$

where a_{ik} is the connection weight between the i^{th} RBF node and the k^{th} output node. A class-based clustering algorithm was applied to the training set to improve the first layer hardware efficiency and to eliminate redundant center locations[10].

One can train parameters $\{\sigma_i\}$ and $\{a_{ik}\}$ by minimizing the squared error at each output node, using a gradient descent technique. The squared error of the k^{th} output node in the presence of the single input vector $\mathbf{x}^l$ is defined as

$$(E_k^l)^2 = \left[\hat{f}_k(\mathbf{w},\mathbf{x}^l) - f_k(\mathbf{x}^l)\right]^2.$$

The gradient descent update equation for the p^{th} RBF width is

$$\Delta\left(\frac{1}{\sigma_p^2}\right) = -\alpha_\sigma (E_k^l)^2 \left(|\mathbf{x}^l - \mathbf{t}^p|^2\right) a_p \exp\left(\frac{-|\mathbf{x}^l - \mathbf{t}^p|^2}{\sigma_p^2}\right), \tag{2.4}$$

where α_σ is the acceleration constant for the width update. The update equation for the interconnection weight between the p^{th} RBF node and the k^{th} output node is

$$\Delta\left(a_{pk}\right) = \alpha_a (E_k^l)^2 \exp\left(\frac{-|\mathbf{x}^l - \mathbf{t}^p|^2}{\sigma_p^2}\right), \tag{2.5}$$

where α_a is the acceleration constant for the weight update. The iterative update equations are applied after each training input vector is presented.

2.2 Handwritten Digit Recognition

Handwritten digit recognition is used to evaluate the performance of our RBF neural network classifier. RBF classifiers have been reported with shorter training times and superior classification performance than either backpropagation networks or k nearest-neighbor classifiers for this problem.[8] Handwritten digit recognition can be viewed as a multidimensional classification problem that has an output class for each of the ten digits. The unsupervised data preprocessing technique applied to each input and digit database is described elsewhere[10]. The training set consists of 600 handwritten digits and the testing set of 300 digits.

To illustrate the capabilities of the RBF neural network in the application of handwritten digit recognition, we train and test a software version of the network described above. The training set inputs are presented sequentially to the network. The error at each output node is calculated and the update equations are applied to minimize iteratively the network errors. The mean squared error (MSE) measure is used to evaluate the system performance as the network trains. The MSE is given by

$$MSE = 1/P \sum_{l=1}^{P} \left(\sum_{j=1}^{K} (E_j^l)^2\right),$$

where $P = 600$ is the number of training set samples and $K = 10$ is the number of output nodes. The best results obtained were 100.0% recognition of the training set and 97.7% correct recognition of the testing set data. These recognition rates serve as a baseline metric for comparing the performance of the software RBF neural network with the optical hardware neural network that we present in the next section.

2.3 Optical RBF Neural Network

Here we describe an optoelectronic implementation of a parallel RBF neural network classifier. The hardware implementation of the RBF classifier is composed of two subsystems; the first is a parallel Euclidean distance computer, which we implement in optics. This subsystem is spatially

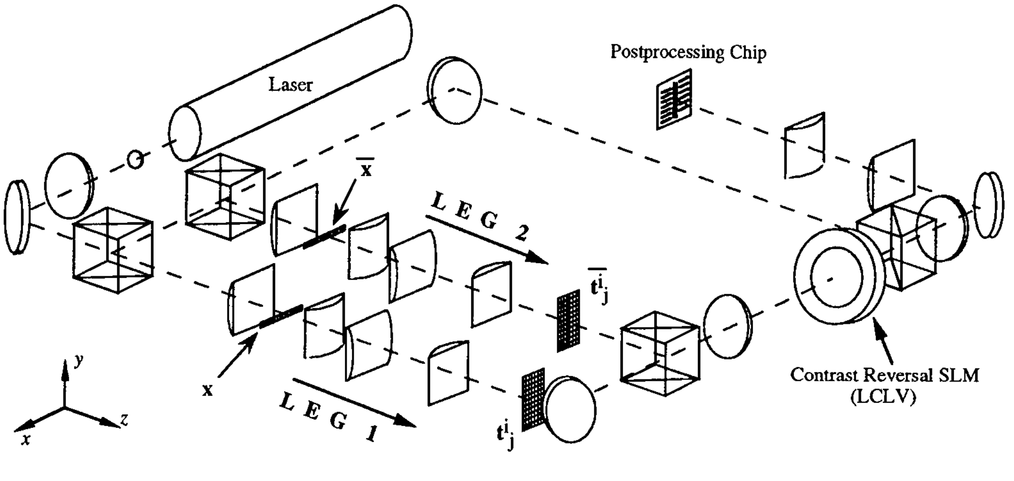

Figure 2.2: Dual-rail vector-matrix multiplier used as a parallel optical distance computer.

multiplexed and makes use of two-dimensional SLM's to represent the center locations. The second subsystem evaluates the basis functions and performs the interconnect weighting between the RBF layer and the output layer. An electronic hardware design is proposed for implementing the post-processing subsystem. In our system the optically computed distances are captured with a CCD camera and the postprocessor is simulated in software.

2.3.1 Parallel Optical Distance Computation

The first layer of the RBF neural network computes the Euclidean distance between the input and each of the centers. The Euclidean distances $\{d^i\}$, between vector $\mathbf{x}$ and the centers $\{\mathbf{t}^i\}$, can be written as

$$d^i = |\mathbf{x} - \mathbf{t}^i|^2 = \sum_{j=1}^{N} (x_j - t^i_j)^2 = \sum_{j=1}^{N} d^i_j \, , \tag{2.6}$$

where

$$\overline{d^i_j} = x_j t^i_j + \overline{x_j}\,\overline{t^i_j} \, , \tag{2.7}$$

for the case of binary vectors. The overbar indicates a bitwise complement. We can implement this distance computation in parallel hardware by using the optical system shown in Figure 2.2. This system is spatially multiplexed, in contrast with previously demonstrated time-multiplexed optical disk-based systems.[20, 23] The light in *LEG 1* of the system illuminates the input SLM, labeled $\mathbf{x}$ and is then collimated in the y direction and imaged in the x direction onto the centers SLM, labeled t^i_j. The result that appears immediately behind the centers SLM is the product term $\{x_j t^i_j \; ; \; i = 1, \ldots, M; j = 1, \ldots, N\}$, which is required in the distance computation. *LEG 2* of the system forms the products $\{\overline{x_j}\overline{t^i_j}\}$ in a similar fashion. The results of these vector-matrix multipliers are simultaneously imaged onto a contrast-reversing liquid-crystal light valve (LCLV) SLM, where they are superimposed to form the terms $\{\overline{d^i_j}\}$. The result of the contrast reversal yields the $\{d^i_j\}$ terms and the final integration operation $\sum_{j=1}^{N} d^i_j$, is performed with a cylindrical lens. For real-time operation, electronically addressed one-dimensional SLM's would be used for the $\mathbf{x}$ and $\overline{\mathbf{x}}$ inputs.

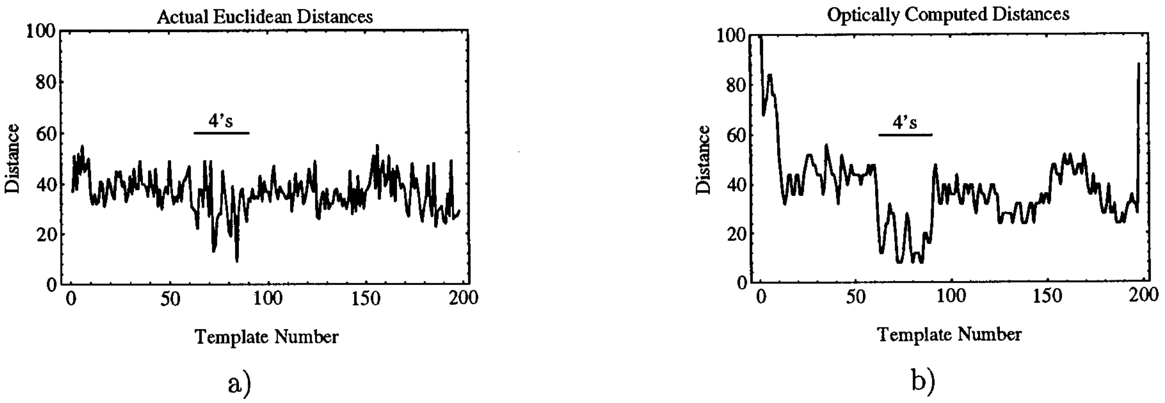

Figure 2.3: Euclidean distance vs. template number for a single input image (handwritten 4). a) Actual distance and b) optically computed distance.

Figure 2.3 shows the actual (a) and optically computed (b) distances between an input vector representing the digit 4 and the 198 center locations (templates). Although large errors are present in the optically computed distances, we observe that the distances are generally smallest for the appropriate template numbers. The accuracy of the optical distance computation can be quantified using the formula

$$\Delta D_{rms} = \frac{\left[1/M \sum_{i=1}^{M} \left(d_i^{Opt} - d_i^{Act}\right)^2\right]^{1/2}}{1/M \sum_{i=1}^{M} d_i^{Act}}, \tag{2.8}$$

where d_i^{Opt} are the Euclidean distances between the 300 testing inputs and the 198 centers calculated from the optical system and d_i^{Act} are the actual Euclidean distances. The average rms distance error over the testing set for our optical system is measured to be 29%. This large 29% distance computation error will lead to a large classification error in our system. The errors present in the optical distance computation are due to device imperfections and system noise, which are discussed in the next section.

2.3.2 Parallel Basis Function Evaluation

We now consider the second subsystem of the RBF classifier, which performs the basis function evaluation and output interconnect weighting. The block diagram in figure 2.4 shows a possible electronic implementation of the subsystem for the single output case.[20] Each array of 198 modules provides a single network output; therefore, we require ten such arrays for our application. Note that all of the proposed operations are local, with exception of the global sum, and are compactly achievable in analog VLSI circuitry. It is also possible, however, that digital techniques would be desirable in order to increase processing speed and/or accuracy within the postprocessor. In this nonadaptive postprocessor the network center widths and interconnection weights can be trained off line in software and then downloaded to the chip during operation. In our experiments the electronic postprocessor chip is emulated in software.

A software RBF neural network was trained by the use of the preprocessed training data. The center widths and interconnect weights are taken from the trained network and loaded into the

Class	Optical	Software
	(Correct out of 30)	
0	0	29
1	0	30
2	0	28
3	4	29
4	30	29
5	28	28
6	5	30
7	26	30
8	0	30
9	0	30
Total	93	293
Overall recognition	31.0%	97.7%

Table 2.1: Performance comparison between optical RBF classifier and software RBF classifier.

software postprocessor emulator. Then, using the optically computed distances of the test set, we evaluated the optical neural network classifier performance. The overall recognition rate for the RBF classifier is 31.0%. This very poor performance does not agree with the 97.7% testing recognition rate measured from the RBF network software simulations presented in Section 2.2. A classwise comparison between the performances of the RBF network that uses the optically computed distances and the perfectly computed distances is shown in Table 2.1. It is evident that the distance errors caused by the optical system imperfections have a catastrophic effect upon the overall system performance. Although it may be possible to use a detailed model of the optical system to compute the center widths and interconnect weights more accurately, the measurement of system parameters would be difficult to make for highly integrated, compact systems. More importantly, temporal changes in the system can greatly effect the system operation. Therefore it is highly desirable to perform on-line adaptation of the network parameters.

2.4 Noise Characterization and Computer Modeling

We now present a noise analysis of the optical RBF system. This analysis will help to identify the limiting noise sources that lead to the large decrease in performance of the optical system as compared to the software RBF neural network. To understand the effects of each noise term on the overall system performance better, we have developed a detailed computer model of the optical system that uses parameters taken from the actual optical hardware. The noise terms that are incorporated in the model include optical system imperfections (illumination nonuniformities, input and centers SLM finite contrast, and optical system point-spread function), LCLV response (intensity transfer function and spatial resolution), and detector response (analog-to-digital quantization error and detector random noise). The following subsections present the mathematical expressions used to represent each of these noise terms in our computer model. Each noise term is characterized

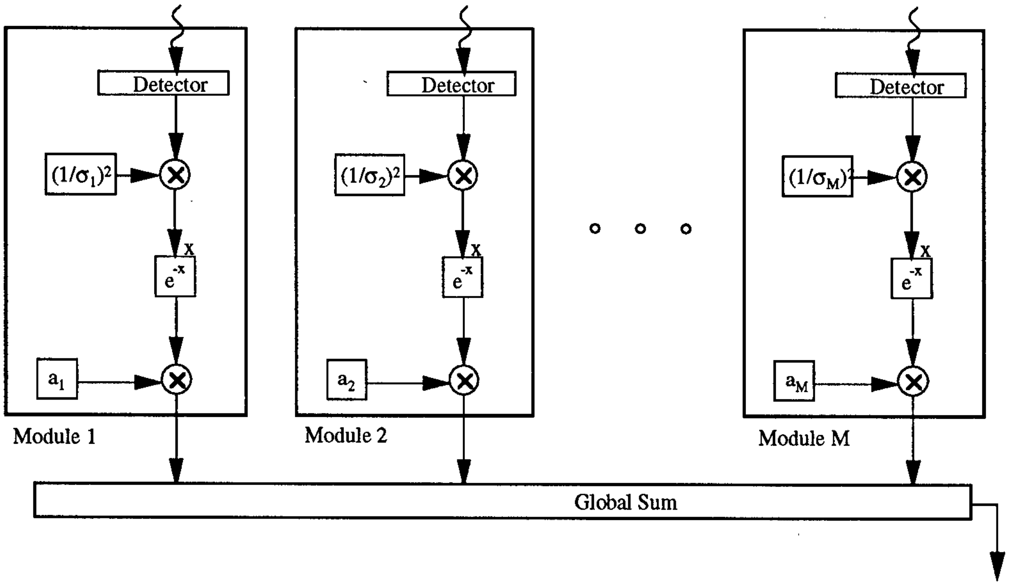

Figure 2.4: Non-adaptive RBF postprocessing chip for a single-output network.

in terms of its effect on the optical distance computation as given in Equations (2.6) and (2.7).

2.4.1 Optical System Noise Characteristics

Illumination Characteristics

There are two different Gaussian laser beam profiles present in each vector-matrix multiplier. This is due to the cylindrical optics that perform collimation along one axis and imaging in the other. The resulting anamorphic beam profile can be measured directly behind the plane of the centers mask SLM. The effect upon the distances can be expressed as

$$\hat{\overline{d^i_j}} = \left[G_1(j,i)x_j t^i_j\right] + \left[G_1(j,i)\overline{x_j}\overline{t^i_j}\right],$$

where

$$G_1(j,i) = \exp\left(-\left(\frac{(j-j_0)}{\sqrt{2}\sigma_1}\right)^2 - \left(\frac{(i-i_0)}{\sqrt{2}\sigma_2}\right)^2\right),$$

$\hat{\overline{d^i_j}}$ is the optically generated estimate of $\overline{d^i_j}$, and (j_0, i_0) represents the midpoint of the centers SLM array. The imaging axis profile σ_1 is along the length of the one-dimensional input SLM. Collimating axis beam profile σ_2 is proportional to the beam expansion in the vector-matrix multiplier and is orthogonal to σ_1. In the LCLV read-out portion of the system there is another Gaussian beam profile (circular) which corrupts the distance computation according to

$$\hat{d^i} = \sum_{j=1}^{N} G_2(j,i)d^i_j,$$

where

$$G_2(j,i) = \exp\left(-\left(\frac{(j-j_0)+(i-i_0)}{\sqrt{2}\sigma_3}\right)^2\right),$$

and $\hat{d^i}$ is the optically generated estimate of d^i. The values shown for the σ parameters in Table 2.2 have been normalized with respect to the SLM pixel dimensions.

SLM Characteristics

In our system we made use of film negatives to act as SLM's for both the center locations and the input vectors. By employing film masks we can achieve very high contrast (> 100:1) and uniformity over the entire mask. While it may be realistic to use these static SLM's for the center locations in a dedicated real-time optical processor, an electronically addressable SLM would generally be required for the input vector SLM's. Electronically addressed SLM's typically have contrast ratios in the range of from 10:1 to 100:1. The effect of finite contrast SLM's on the distance computation can be represented by

$$\hat{\overline{d^i_j}} = \left[x_j + \left(\frac{1-x_j}{C_r}\right)\right]\left[t^i_j + \left(\frac{1-t^i_j}{C_r}\right)\right] + \left[\overline{x_j} + \left(\frac{1-\overline{x_j}}{C_r}\right)\right]\left[\overline{t^i_j} + \left(\frac{1-\overline{t^i_j}}{C_r}\right)\right],$$

where C_r is the contrast ratio.

Liquid-Crystal Light Valve Characteristics

The device used in our system to reverse the contrast is a Hughes LCLV (made before 1987). In our experiment the contrast, uniformity, and resolution were critical characteristics for the LCLV device. The contrast is determined by the transfer function of the device. The intensity transfer function of our contrast-reversing LCLV is shown in Figure 2.5 along with the response curve used in the computer model. The effect of the LCLV response on the optical distance computation can be represented as

$$\hat{d^i} = \sum_{j=1}^{N} TF(\overline{d^i_j}),$$

where TF is the transfer function shown in Figure 2.5.

We measured our LCLV resolution to be 8.0 lines/mm by using the Air Force resolution target. The pixel pitch in our centers masks is 63.5 μm which requires a minimum resolution of 15.8 lines/mm at a 1.0× magnification. The effective LCLV aperture over which uniform contrast reversal could be achieved was 15.0mm in width, which allows us to image the centers masks onto the LCLV at an increased magnification. The factor that limits this magnification is nonuniformity in the active area of the LCLV.[24] The LCLV resolution limit and optical system point-spread function are modeled together as a Gaussian blur that affects the optical distance computation according to

$$\hat{\overline{d^i_j}} = h(i,j) * (x_j t^i_j + \overline{x_j t^i_j}),$$

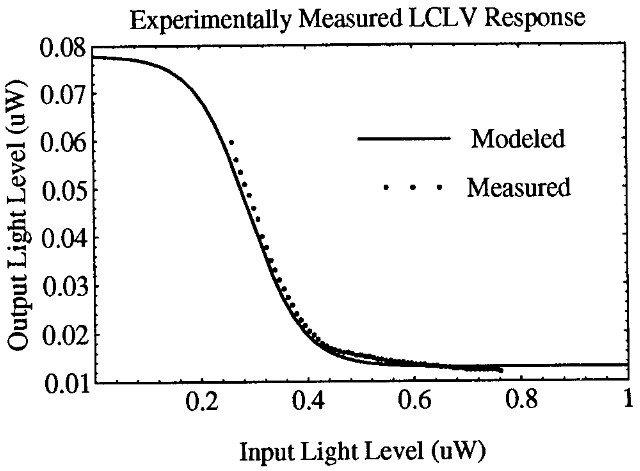

Figure 2.5: Measured and modeled LCLV transfer functions.

where the $*$ symbol denotes a convolution and h(i,j) is an $n \times n$ Gaussian convolution kernel. In our system the Gaussian kernel width is measured to be $\sigma_b = 0.85$ pixels of the centers SLM (labeled t^i_j in Figure 2.2) which can be adequately modeled with a 3×3 pixel kernel. Although the results presented here depend on the characteristics of the LCLV used, an increased uniform response area and higher speeds can be easily achieved in newer optically addressed SLM's.[25]

Detector Noise and Dynamic Range Characteristics

The effects of random detector noise on the optical distance computations can simply be represented by

$$\hat{d^i} = \sum_{j=1}^{N} (d^i_j + \eta_d),$$

where η_d is a random variable with a Gaussian distribution. The detector noise is modeled as stationary additive noise and is a result of detector dark current. The noise variance was assumed to be on the order of 2% of the detector's full scale range.[26] The analog-to-digital (A/D) quantization error also effects the distance computation. This noise source can be represented as

$$\hat{d^i} = \sum_{j=1}^{N} (d^i_j / d^i_{j_{max}})(2^{n\,bits}),$$

where $2^{n\,bits}$ is the number of quantization levels in the A/D conversion. Eight bit quantization was used in our optical RBF system. These noise sources can also represent the effects of reduced accuracy in the analog realization of the electronic postprocessor.

Dual-Leg Beam Ratio

Another important error source that affects our optical distance measurements is the ratio of light intensities between the two legs (*LEGS 1* and *2* as shown in Figure 2.2) in the optical distance

Parameter	Measured Values
Nonuniform Illumination	
Imaging Axis, σ_1	97.5 pixels
Collimating Axis, σ_2	131.7 pixels
Read-out Beam, σ_3	340 pixels
SLM Finite Contrast, C_r	> 100:1
Leg Intensity Ratio, β_1/β_2	0.96
LCLV Blur, σ_b	0.85
Detector Noise, σ_d	0.02
A/D Quantization, $2^{n\,bits}$	8 bits

Table 2.2: Measured error values from optical system.

computer. The impact of this light intensity ratio on the optical distance computation can be expressed as

$$\hat{\overline{d_j^i}} = \beta_1(x_j t_j^i) + \beta_2(\overline{x_j}\overline{t_j^i}),$$

where β_1/β_2 is the leg intensity ratio. We make use of a polarizing beam-splitter to direct orthogonally polarized light into each of the two legs in order to minimize interference fringes at the summing beam-splitter. Placing a wave plate before the polarizing beam-splitter allows us to adjust the leg intensity ratio.

2.4.2 Optical System Model

Using our computer model of the optical system together with the measured system noise parameters shown in Table 2.2, we generate a set of simulated optical testing set distances. As in the previous section, we again use the center widths and interconnect weights computed off-line from the software RBF neural network to test the system. This time we use the simulated optical distances instead of the actual optical distances. Our testing recognition result is 29.0%, which agrees well with the recognition rate of 31.0% obtained with the actual optically computed data. The class-wise performance comparison between the model predictions and the optical system is shown in Table 2.3. This result suggests that we can now accurately predict the system performance of our optical hardware RBF neural network by using our computer model. Furthermore, the model allows us to simulate other system operating points from which we can deduce which noise terms are most critical for the overall system performance. These issues will be pursued further in the next section.

2.5 Adaptive Optical Radial Basis Function Neural Network

As we have seen in Sections 2.3 and 2.4, our optical hardware network performance greatly suffers because of errors in distance computation caused by optical system imperfections and noise. We expect that by training our optical system in the presence of these imperfections, the network will

Class	Optical	Model
	(Correct out of 30)	
0	0	0
1	0	0
2	0	0
3	4	1
4	30	30
5	28	28
6	5	2
7	26	26
8	0	0
9	0	0
Total	93	87
Overall recognition rate	31.0%	29.0%

Table 2.3: Performance comparison between optical and modeled optical RBF classifier.

compensate for these errors and its performance will more closely match that of the software RBF network. In this section we consider an adaptive postprocessor[20] and present computer model comparisons between adaptive and non-adaptive RBF classifier systems.

2.5.1 Adaptive Postprocessor

By modifying the non-adaptive single-output postprocessor presented in Section 2.3, we can incorporate error feedback in order to implement on-line network training. Using the electronic module shown in Figure 2.6 we can directly implement the gradient descent update Equations (2.4) and (2.5). In addition to error feedback for each module, we also require accumulation registers for adapting both the RBF widths and weights in an iterative fashion. As in the non-adaptive case we require ten arrays, each consisting of 198 modules for the optical fully-parallel implementation of the digit recognition RBF network shown in Figure 2.1.

2.5.2 Influences of Noise on System Performance

Using a computer model of the optical Euclidean distance computation subsystem and its ability to include the effects of optical system error sources, we have studied the effects of these errors on the optical RBF system's recognition performance. Optical systems that use adaptive or on-line training to compensate for these errors can be compared with optical systems that are trained off line in order to estimate the utility of adaptive training in this application. We now present computer model performance comparisons between the adaptive and non-adaptive hardware systems. Two types of error models are quantified for each of the error sources found to be most critical in our system. The first error model assumes the optical system to be ideal (error free) except for the error under consideration. This case is represented by the circular symbols in Figures 2.7-2.10. The

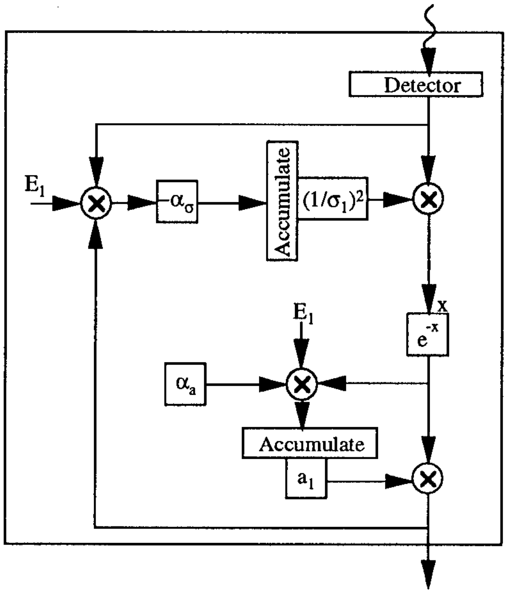

Figure 2.6: Adaptive RBF postprocessing chip module.

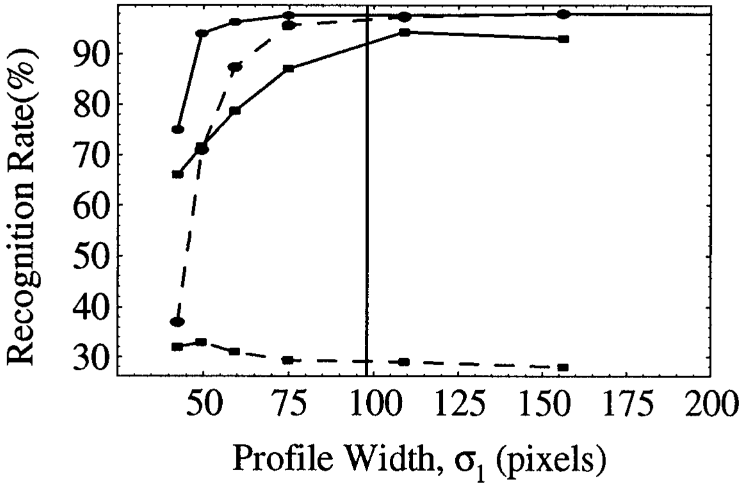

Figure 2.7: Predicted recognition rate versus imaging axis beam profile. Circles indicate isolated error model, squares indicate multiple error model, solid curves indicate adaptive case and dashed curves indicate non-adaptive case.

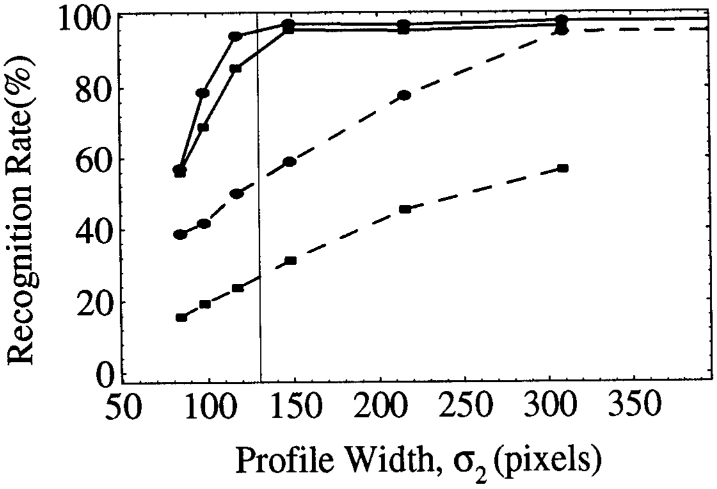

Figure 2.8: Predicted recognition rate versus collimating axis beam profile. Symbols are as defined in Figure 2.7.

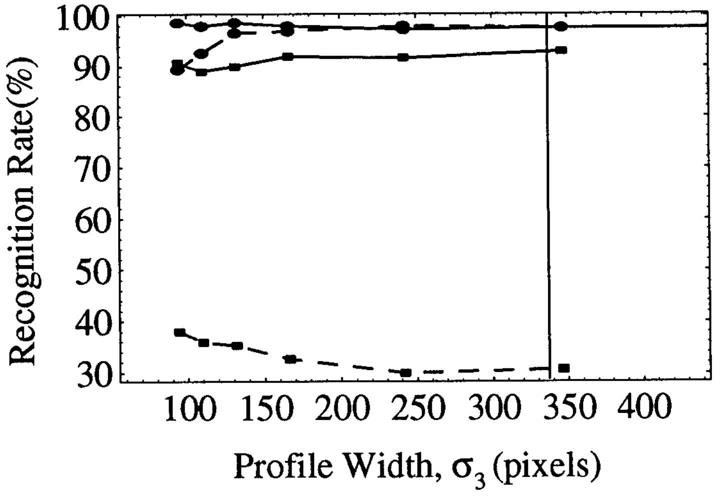

Figure 2.9: Predicted recognition rate versus LCLV readbeam profile. Symbols are as defined in Figure 2.7.

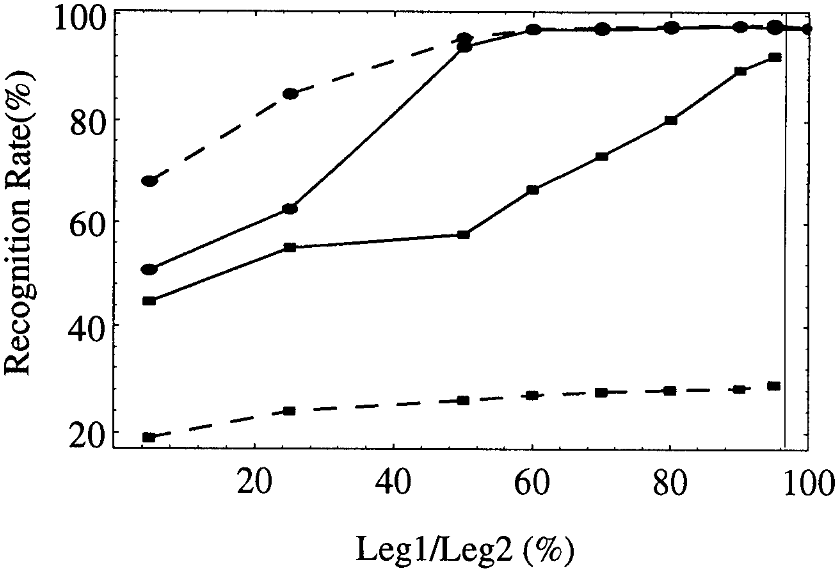

Figure 2.10: Predicted recognition rate versus leg intensity ratio. Symbols are as defined in Figure 2.7.

second type of error model predicts the influence of the error under consideration, given that all other parameters are fixed at their values as measured from the optical system and as shown in Table 2.2. This case is represented by the squares in Figures 2.7-2.10. Associated with each type of error model are two curves; the solid curve indicating the system testing performance when trained in the presence of the imperfection (adaptive on-line training), and the dashed curve shows the testing results when trained off line as in the nonadaptive case. The measured system operating points are indicated in each plot by a vertical line.

Figure 2.7 shows the predicted system testing recognition rate as a function of the imaging axis beam profile, and Figure 2.8 shows the result for the collimating axis case. As expected, the performance in both cases degrades with an increasingly nonuniform beam profile. It can also be observed that the on-line training significantly improves the performance as compared to the off-line training. Although the imaging axis error effects all of the centers by the same degree, it will reduce a portion of the system's discrimination ability if the pixels on the edge of the SLM are not of sufficient intensity to transmit through the LCLV. The effect of the collimating beam error has more influence on system performance because it affects each center location vector to a different degree. It can be shown that the digit classes with centers near the outer edges of the SLM are more likely to be misclassified (e.g., see Table 2.1), attesting to the spatial dependence of the centers. Although it may be possible to use this *a priori* information to design a more robust sequence of the centers, it is more desirable to correct the beam profile by the use of increased beam-expansion, diffractive optic elements, or holograms. The predicted system performance in the presence of the LCLV readout beam profile is shown in Figure 2.9. The readout profile is shown to have little impact on the system performance. The system performance does not degrade appreciably until the SLM contrast C_r falls below 10. This differs from the results of previous systems[20], in which finite SLM contrast was found to be a dominant error source.

The predicted results of the leg intensity ratio are presented in Figure 2.10. It is interesting to

Class	Optical	Model
	(Correct out of 30)	
0	30	29
1	30	30
2	22	23
3	28	28
4	30	30
5	21	26
6	27	28
7	30	30
8	30	30
9	30	30
Total	278	284
Overall recognition	92.67%	94.67%

Table 2.4: Performance comparison between adaptive optical RBF classifier and the modeled adaptive optical RBF classifier.

note that in the ideal system case the adaptive training fails to outperform the non-adaptive system. This result confirms the requirement for a dual-leg system to eliminate biases in the input data. The detector noise and dynamic range were found to have little impact upon the system performance. The detector noise level is considered negligible compared with the other noise terms. Simulation results indicate that a SNR of ≈ 4 was tolerable in both the adaptive and non-adaptive cases. The system model also indicates that the network performance does not severely degrade until fewer than three bits are used. These results indicate that an analog electronic postprocessor should have adequate accuracy for this system.

2.5.3 Adaptive Optical RBF Neural Network Results

We now use the adaptive system model and the measured noise parameter values from the experimental optical system in Table 2.2 to make predictions about the adaptive system performance. The model predicts an overall testing recognition rate of 94.7% when trained on-line. Using the actual optically computed testing distances together with the on-line learning gives a final recognition rate of 92.67%. Table 2.4 shows the classwise recognition performance corresponding to these results. Again the model is in excellent agreement with the experimental system performance, illustrating the utility of the computer model predictions. The adaptive optical system provides us with nearly 60% increase in recognition performance as compared with the non-adaptive optical RBF classifier, attesting to the significant improvement imparted by adaptive on-line training.

3

GRAYSCALE INPUT NEURAL NETWORK CLASSIFIER

In this chapter we propose a grayscale input distance computer for use with our RBF classifier. We show that by modifying the distance computation optics of the binary classifier system we can develop a grayscale input capable classifier. In essence we replace the XOR functions with optical image subtraction optics. In this chapter we briefly present an appropriate optical image subtraction method and the characterization results of the phase shifting device employed in the experiment. It is expected that the binary-input neural hardware will be more tolerant to internal system error sources than will a grayscale-input implementation due to the dynamic range limitations of the optical detectors and the summation of multiple input channels into a single output channel.

3.1 Optical Image Subtraction

Because of its simplicity we chose to perform the real-time image subtraction by a phase shifting method[27], rather than by Fourier methods[28, 29] or electrooptic (via SLMs) effect methods[30]. Reviews of optical image subtraction techniques are given in Ebersole[27] and Liu and Chao[31].

In a coherent optical system, the simplest way to achieve image subtraction is by a phase shifting method. As shown in Figure 3.1 two paths are split from a single coherent source and each illuminates an image. A phase plate or variable phase retarder is placed in one of the two paths to create a phase delay of π which multiplies the image in that path by $e^{i\pi} = -1$. The image subtraction is completed when the two paths are added together in the combining beamsplitter. The Euclidean distances $\{d^i\}$, between the input vector $\mathbf{x}$ and the centers $\{\mathbf{t}^i\}$, can be directly computed as $d^i = |\mathbf{x} - \mathbf{t}^i|^2$ by placing the input vector in one path and the centers in the other path. We make use of a variable phase retarder in order to easily adjust the phase delay without disturbing the position of the mounted optical components. The capability of phase delay adjustments also permits us to perform software error analysis over a wide operating range. We begin by characterizing the liquid crystal variable retarder to be used in our system. The detector array (i.e., postprocessing chip) detects the magnitude of the amplitude image subtraction, which is our desired distance

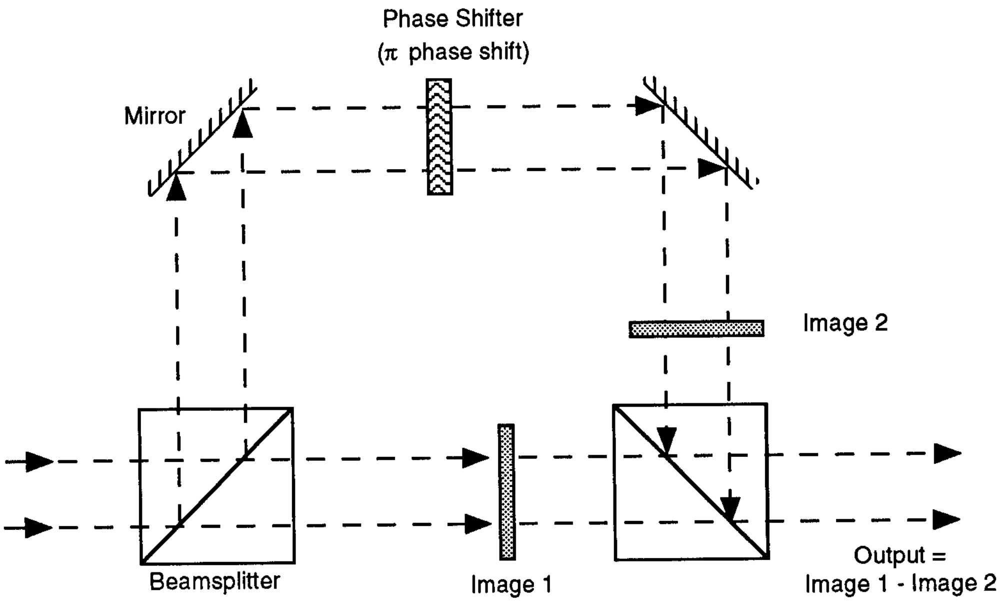

Figure 3.1: Optical image subtractor using phase-shifter.

measure.

3.2 Variable Retarder Characterization

In this section we characterize the Meadowlark Optics liquid crystal variable retarder. The objective is to measure the amount of retardance the device produces as a function of applied voltage in order to quantify the system performance degradation likely to be introduced by errors in the phase delay.

3.2.1 Operation

The retarder operates on the principle of birefringence. Birefringence is the property where a material exhibits two different indices of refraction along two different optical paths. These refractive indices are termed the ordinary refractive index, n_o, and extraordinary refractive index, n_e. Recall the equation for the speed of light, ν, traveling through some material is $\nu = c/n$ where c is the speed of light in a vacuum, and n is the index of refraction parallel to the axis of polarization of the light. For the type of material used in this retarder, $n_e > n_o$, and thus n_e is termed the slow axis and n_o is termed the fast axis. Light polarized parallel to the slow axis travels slower than light polarized parallel to the fast axis.

The retarder consists of a liquid crystal material sandwiched between two glass windows. Each of these windows has a layer of optically transparent, electrically conductive indium tin oxide (ITO) applied. The liquid crystal is a birefringent material whose birefringence is dependent upon the alignment of the liquid crystal molecules. This alignment can be changed by applying an electric field, via the ITO, which changes the orientation of the molecules. The long axis of the

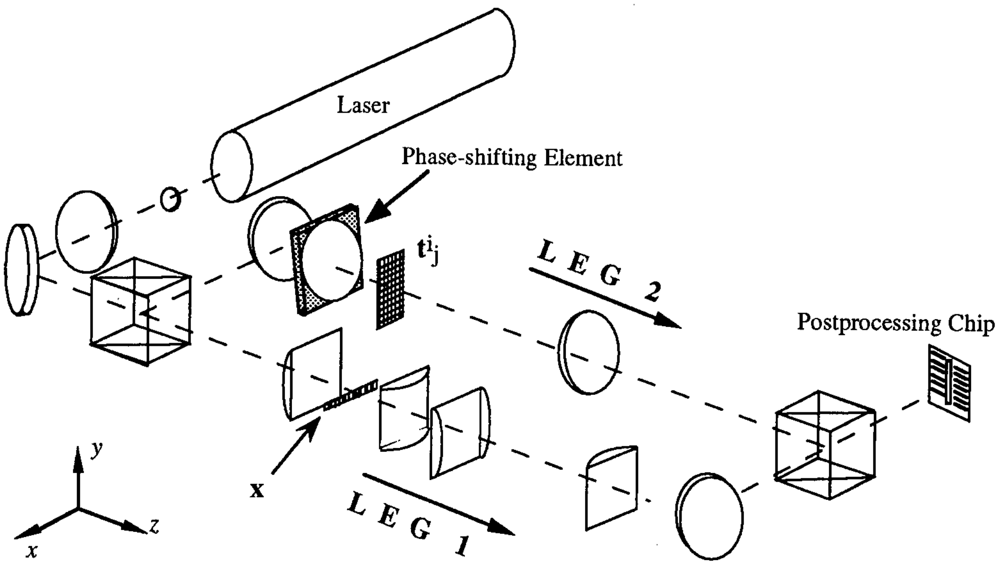

Figure 3.2: Optical image subtractor used as a grayscale-input parallel distance computer.

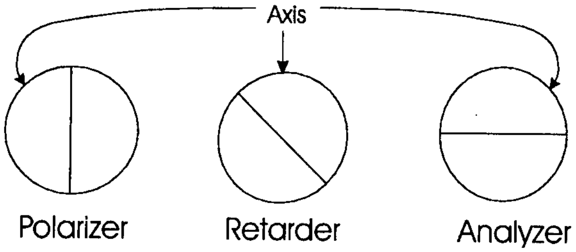

Figure 3.3: Retarder used as an attenuator.

molecules determines the slow axis, and with no electric field applied, the molecules lie parallel to the glass windows and exhibit a maximum index of refraction, n_e. When an electric field is applied, the molecules start to align themselves with the field which decreases n_e, and thus decreases the amount of birefringence.

To demonstrate a use of the retarder, assume linearly polarized light, whose axis of polarization is aligned between the fast and slow axis of the birefringent material, is incident on the retarder. This causes the phase of one orthogonal component of the light to lag the other which changes the polarization of the light. By varying the electric field, one can obtain various output polarizations. These polarizations range from linear to elliptical to circular, depending on how much one orthogonal component is lagged behind the other. This change in polarization can be used to make the device act as a variable attenuator by placing a polarizer and analyzer before and after the device, respectively. If the incident light's axis of polarization is 45 degrees from the slow (or fast) axis of the retarder (Figure 3.3), the resulting transmittance, T is the following function of retardance, R:

$$T = \sin^2(R/2).$$

Another use of the retarder is to adjust the optical path length of light. If one aligns the axis of polarization of linearly polarized light to the slow axis of the retarder, that polarization will undergo a phase change caused by n_e. Thus, varying n_e by applying an electric field across the liquid crystal material, one can dynamically adjust the optical path length. This is how the retarder is to be used in the experiment.

3.2.2 Procedure

The most apparent way to make the measurement was to setup an interferometer, place the device in one of the optical paths of the interferometer, and measure the fringe movement as a function of voltage. The fringe movement is proportional to the change in optical path length. A Michelson interferometer was to be used with a digital CCD camera (Cohu 4110) that would measure the fringe movement. The movement is characterized by

$$\Delta d = N(\lambda/2),$$

where (Δd is the change in the optical path, N is the number of fringes one of the fringes moved, and λ is the wavelength of the light used.

A cable had to be ordered that interfaced the camera to the frame buffer board inside the PC. While waiting for the cable (and the retarder), software was written to find and track the sinusoidal fringes, and the interferometer was fabricated. Since the CCD camera is also capable of producing RS-170 analog output, and the frame buffer board we had was also capable of frame grabbing analog video, some preliminary tests of the test system could be made.

It became readily apparent the software was subject to locking onto local maxima if the resulting frame grabbed image was noisy. If the noise was not caused by the A/D converters of the framegrabber, and hence would appear in the final setup, either the images would have to be averaged, and/or an algorithm would have to be added to find the true peak of the fringe. A simple global maxima algorithm could not be used since we needed to find the same peak each time which may have an amplitude less than a neighboring peak. There was also a logistics problem in getting the correct cable for the camera. While waiting for supplies, it seemed prudent to explore alternative methods for making the measurements.

A second way of making the measurements exploits the fact the retarder can vary the polarization of linearly polarized light. Using the principle shown in Figure 3.3, the setup in Figure 3.4 would allow us to measure the amount of retardance the device produces. This has several advantages over the previous method. First, the controlling software is greatly simplified. Second, the dynamic range and accuracy of an optical power meter is greater than the CCD camera. The disadvantage is that a method needed to be produced to align a linearly polarized beam 45 degrees from one of the optical axis of the retarder. Because of the advantages of this approach, and the fact that when the cable did arrive, it did not work with the camera, the above method was abandoned, and this one was used to make the measurements.

A five step process was used to align the system shown in Figure 3.4. First, circularly polarized light was required. This will become clear when step (4) is discussed. A quarter waveplate was used to turn the linearly polarized light into circularly polarized light. Although the HeNe laser was linearly polarized, the contrast ratio is only approximately 500:1. This was cleaned up by adding a Glan-Thompson polarizer with a contrast ratio of approximately 100,000:1 (note: all polarizers

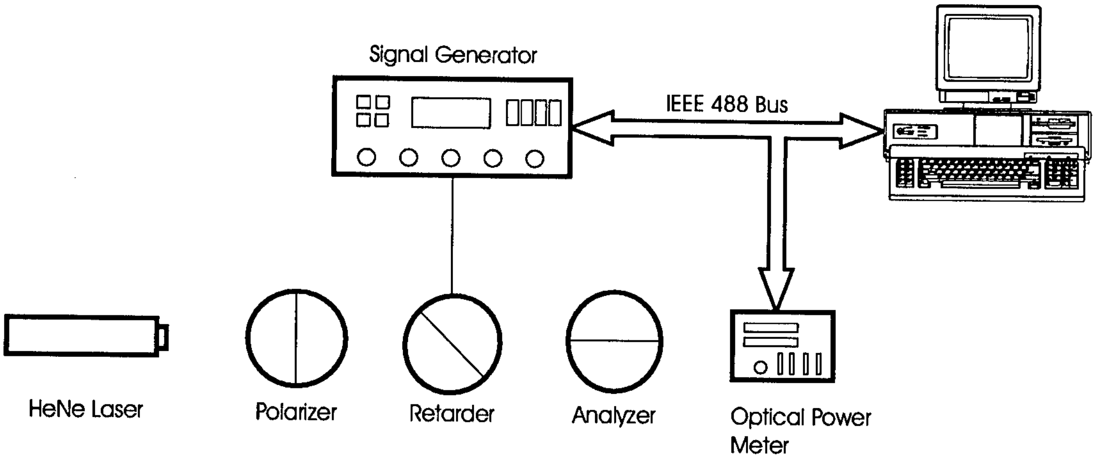

Figure 3.4: Variable retarder characterization test setup.

used were Glan-Thompson or Glan-Taylor). Correct polarization was confirmed by adding another polarizer and rotating it while monitoring the power of the light with an optical power meter. When the light was circularly polarized, there was no change in the optical power seen by the meter.

In step (2), a polarizer, PA, was rotated until its axis was aligned to the fast (or slow) axis of the retarder. This was verified by placing an analyzer, PB after the retarder and monitoring the power of the light. When PA is aligned properly, there will be no change in optical power when voltage is applied to the retarder. See Figure 3.5.

Step (3) added a quarter waveplate, WP, to turn the linearly polarized light into circularly polarized light. It was verified using polarizer PB as in step (1). Since a quarter waveplate turns linearly polarized light when its axis is 45 degrees from the light's axis of polarization, the waveplate's axis is now aligned 45 degrees from the axis of the retarder.

Step (4) aligns PA to that of WP. PA and PB are both rotated until maximum power is incident on the meter. The light going into PA must be circularly polarized or PA will be rotated until its axis is aligned to the axis of the linearly polarized light. PA is now properly aligned.

Step (5) aligns the analyzer, PC, perpendicular to PA. Using the graph that came with the retarder, apply the proper voltage that will yield zero retardance. Then, rotate PC until minimum power is incident on the meter. Adjust the voltage on the retarder to verify it's not retarding. This can be determined by observing the meter. If the retarder was retarding, the output polarization would be a small ellipse that would allow light to go through PC, and so if adjusting the voltage to the retarder gets it closer to true zero retardation, there would be a decrease in the power measured by the meter caused by the minor axis of the ellipse getting smaller and hence the polarization becoming closer to linear. Step (5) would then be repeated until an absolute minimum is read by the meter.

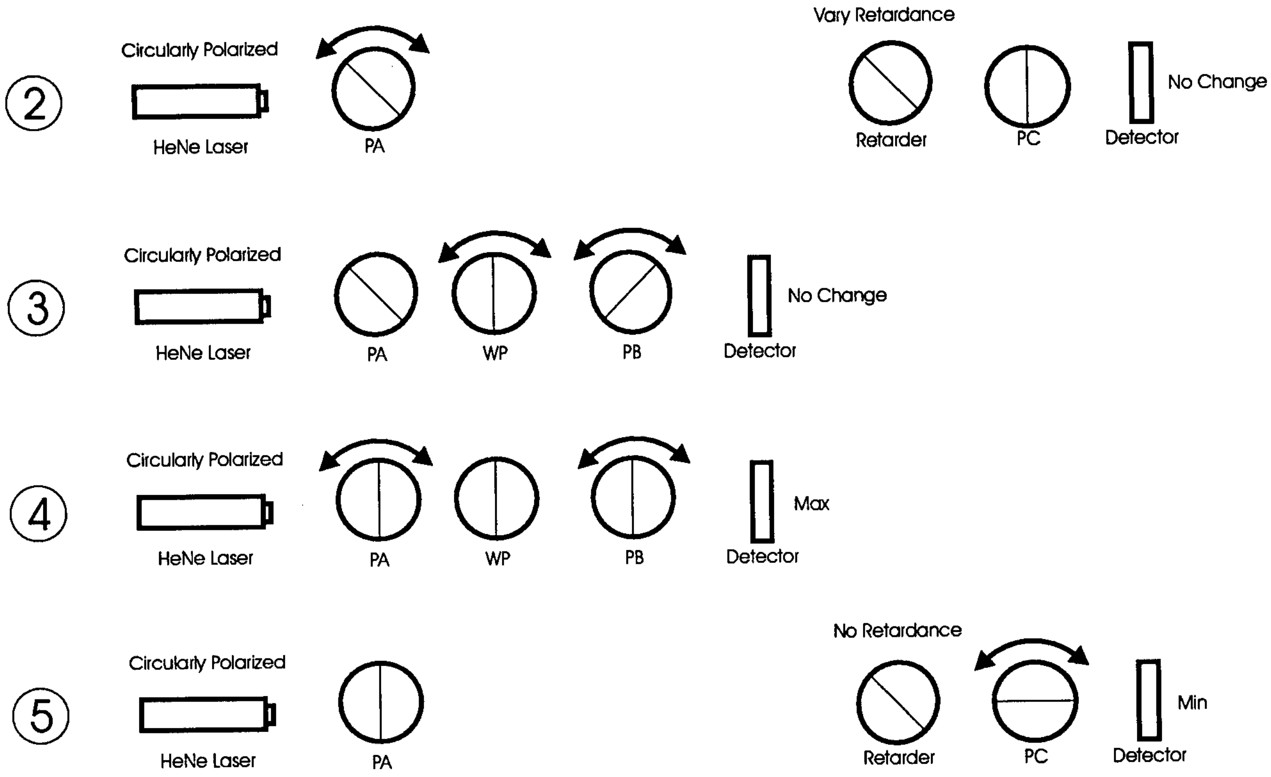

Figure 3.5: Alignment steps for variable retarder characterization.

3.2.3 Results

After taking a set of measurements, it was clear there was noise present in the system–either the laser or the power meter, or both. Eleven sets of data were taken over three days to be averaged together. However, each day's worth of data differed slightly. The peaks in the waveforms occurred at different levels. This was probably caused by the varying temperatures we experienced in the lab due to faulty air-conditioning which affected the detector's responsivity, and the laser's output. To account for this, the data was normalized before the averaging. Figure 3.6 shows the final output.

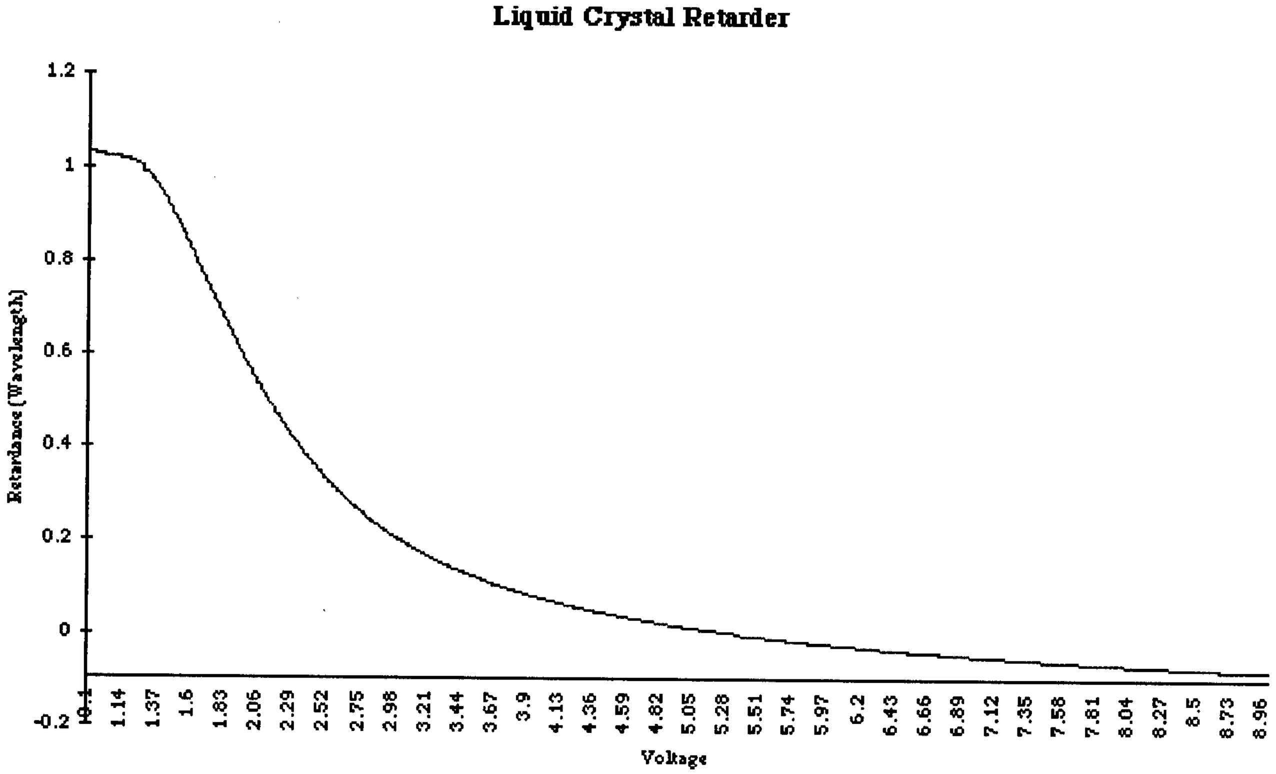

Figure 3.6: Final results of variable retarder characterization.

4

WAVELET PREPROCESSING

In this chapter we present a multiscale optical wavelet transform for use as a preprocessor for our optical neural network classifier system. The purpose of the preprocessor is to extract important features from the input signal and provide these features to the classifier. The performance of the classifier is partially determined by the choice of features provided as input. In applications using focal plane imagery, the image pixels themselves can be used as features. However, image pixels are typically numerous and form an unnecessarily large feature set. This large feature set requires an unacceptably large neural network.

Edges are important features in an image and are therefore usually a better choice for input features. They define object boundaries as well as textures. In addition, they form a more compact feature set than the image pixels. To represent objects and textures of different sizes requires multiscale edge features [32]. Wavelet transforms, currently popular in signal processing, perform a multiscale or multiresolution analysis of an image. The resulting wavelet representation contain edge features at multiple scales.

The remainder of this chapter describes an imaging system based on the joint-transform correlator (JTC) that produces a multiple wavelet-scale version of an input image. In the next section, we briefly describe the general theory of a conventional JTC and then we extend the discussion to multiple inputs. Next, we describe how the multiple-input JTC is used for multispectral analysis using wavelets. Finally, we show some simulations and presented our conclusions.

4.1 Conventional Joint Transform Correlator

To perform the correlation operation with a JTC, functions are encoded in the input plane. A schematic diagram of a conventional JTC is shown in Figure 4.1. To perform the cross-correlation between the images $b(x, y)$ and $d(x, y)$ they are centered at $x = \pm\alpha$ as shown in Figure 4.1 and Figure 4.2a. A lens produces the Fourier transform when the input plane is illuminated with coherent light. In the Fourier plane, the complex light field is

$$U = B(p,q)e^{j\alpha p} + D(p,q)e^{-j\alpha p} ,$$

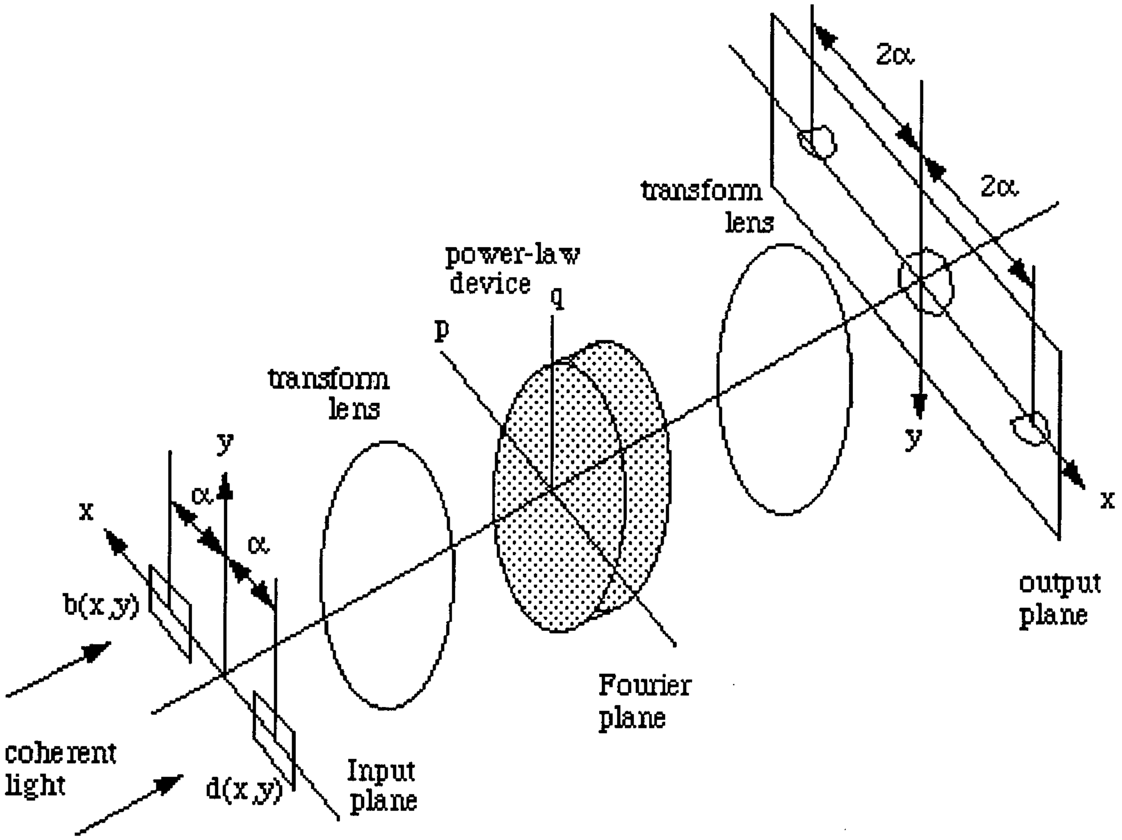

Figure 4.1: Schematic diagram of a conventional joint-transform correlator.

where $B(p,q)$ is the Fourier transform of $b(x,y)$, and similarly for $d(x,y)$. A square-law device such as a liquid crystal light valve is placed in the Fourier plane before an additional Fourier transform is performed. The output intensity distribution from a square-law detector can be written as

$$|U|^2 = UU^* = [B(p,q)e^{j\alpha p} + D(p,q)e^{-j\alpha p}][B(p,q)e^{j\alpha p} + D(p,q)e^{-j\alpha p}]^* \ .$$

Multiplying terms, taking the Fourier transform with a second lens, and grouping terms results in

$$I = b(x,y) \otimes b(x,y) + d(x,y) \otimes d(x,y) + b(x,y) \otimes d(x+2\alpha,y) + d(x,y) \otimes b(x-2\alpha,y) \ ,$$

in the output plane, where $\otimes$ indicates the correlation operation. The first two terms are the autocorrelations of the input functions and appear on the optical axis. The third and fourth terms are the cross-correlation between the two input functions and appear at $x = \pm 2\alpha$. This is shown in Figure 4.1 and Figure 4.2b.

To perform the wavelet transform at one scale using a JTC, one input is the image of interest and the other is the wavelet function. The wavelet transform at a particular scale then appears at the output of the JTC. Because wavelets have zero mean, this may present a difficulty in the implementation. To remove the DC component from the wavelet function, it was experimentally shown that the input functions could be encoded in phase to eliminate the DC component.[33]

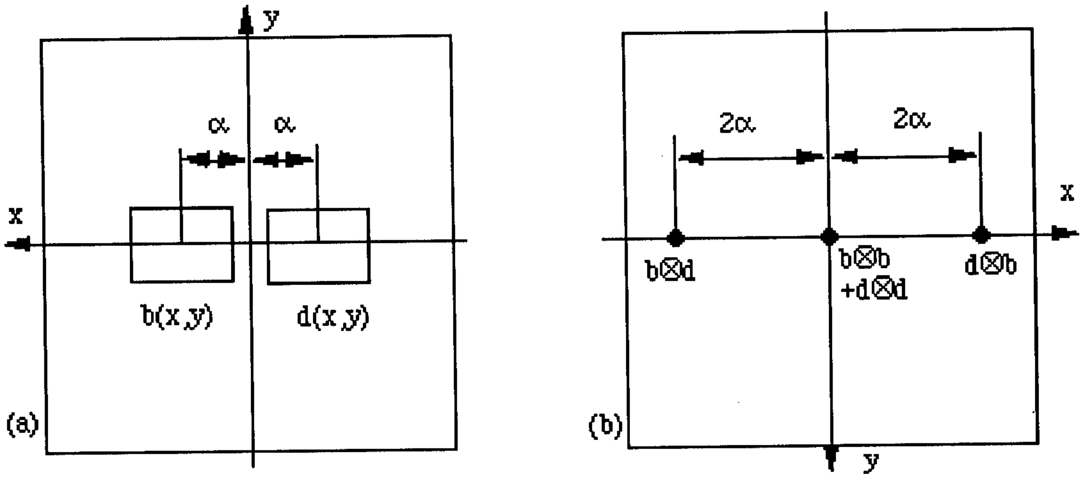

Figure 4.2: Input and output planes of a JTC showing locations of inputs and outputs (a) input plane (b) output plane.

4.2 Multiple-input Joint Transform Correlator

4.2.1 Theory

We considered a JTC with the use of any number of inputs arranged along a line as the inputs. We used the JTC as in Figure 4.1 but considered n inputs separated by α in the input plane arranged along the x-axis as shown in Figure 4.3. The images were labeled $a_1(x,y)$ to $a_n(x,y)$, with the center image labeled as $a_{(n+1)/2}(x,y)$. Using this configuration, the complex light field in the Fourier plane was

$$U = A_1(p,q)e^{j(\frac{n-1}{2})\alpha p} + A_2(p,q)e^{j(\frac{n-3}{2})\alpha p} + \cdots A_{\frac{n+1}{2}}(p,q) + \cdots A_n(p,q)e^{-j(\frac{n-1}{2})\alpha p}\ .$$

The output intensity distribution from a square-law detector was written as $|U|^2 = U \times U^*$. Multiplying $n \times n$ terms, taking the Fourier transform, and grouping terms resulted in $2n-1$ locations in the output plane where a correlation response would occur. The output plane was described as

$$I = \left[\sum_{i=1}^{n} a_i(x,y) \otimes a_i(x,y)\right] +$$
$$\left[\sum_{i=1}^{n-1} a_i(x,y) \otimes a_{i+1}(x+\alpha,y) + \sum_{i=1}^{n-2} a_i(x,y) \otimes a_{i+2}(x+2\alpha,y) + \cdots \sum_{i=1}^{n-(n-1)} a_i(x,y) \otimes a_{n-1}(x+(n-1)\alpha,y)\right]$$
$$+\left[\sum_{i=1}^{n-1} a_{i+1}(x,y) \otimes a_i(x-\alpha,y) + \sum_{i=1}^{n-2} a_{i+2}(x,y) \otimes a_i(x-2\alpha,y) + \cdots \sum_{i=1}^{n-(n-1)} a_{n-1}(x,y) \otimes a_i(x-(n-1)\alpha,y)\right] \quad (4.1)$$

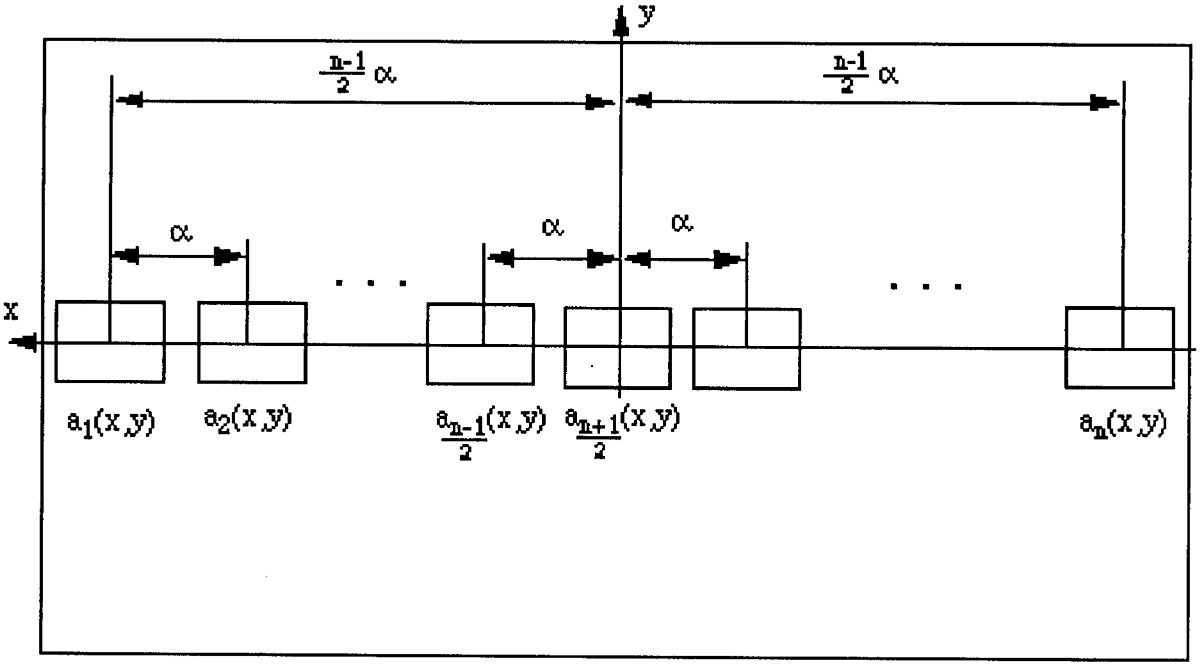

Figure 4.3: Input plane of multiple-input JTC.

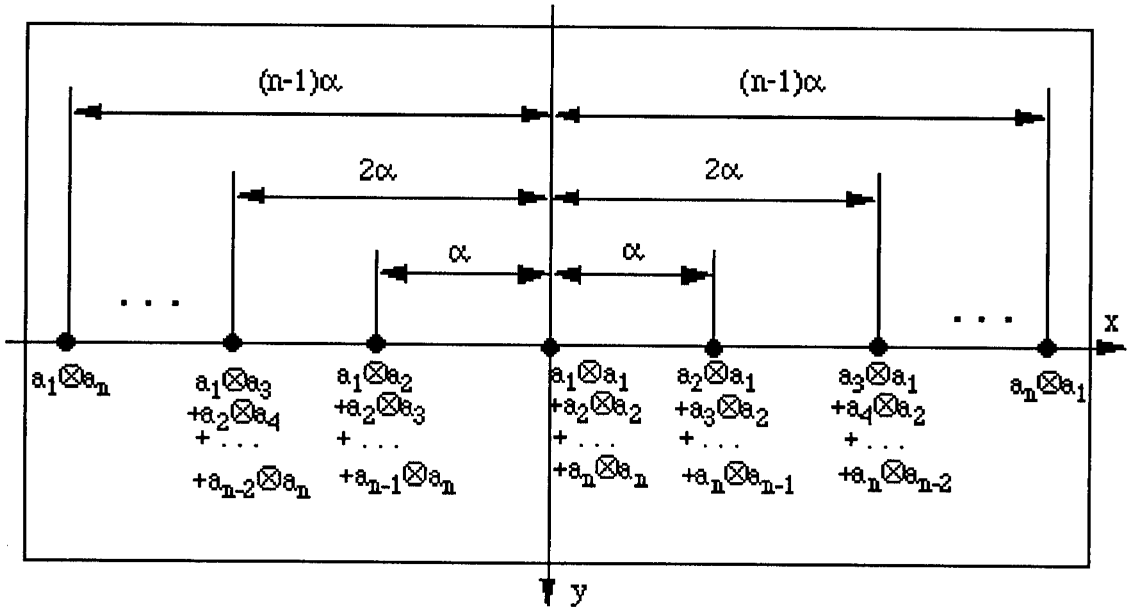
$(n-1)\alpha$
$(n-1)\alpha$
2α
2α
α
α
. . .
. . .
x
$a_1 \otimes a_n$
$a_1 \otimes a_3 + a_2 \otimes a_4 + \ldots + a_{n-2} \otimes a_n$
$a_1 \otimes a_2 + a_2 \otimes a_3 + \ldots + a_{n-1} \otimes a_n$
$a_1 \otimes a_1 + a_2 \otimes a_2 + \ldots + a_n \otimes a_n$
$a_2 \otimes a_1 + a_3 \otimes a_2 + \ldots + a_n \otimes a_{n-1}$
$a_3 \otimes a_1 + a_4 \otimes a_2 + \ldots + a_n \otimes a_{n-2}$
$a_n \otimes a_1$
y

Figure 4.4: Output plane of multiple-input JTC.

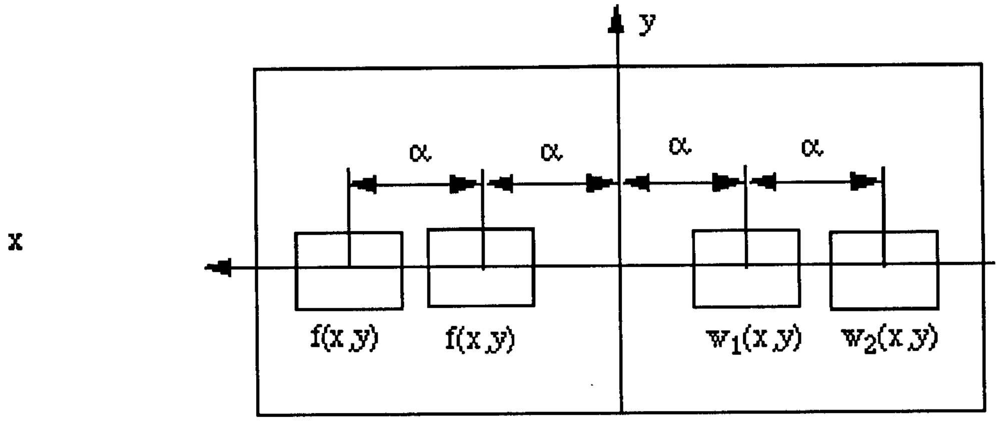

Figure 4.5: Input plane of multispectral example.

The first term in brackets in Eq. 4.1 contains the DC terms which is the sum of the autocorrelations of all the input images. The second term in brackets correspond to $n-1$ terms to the left of the DC term which are shown in Figure 4.4; each one of these terms is separated by a distance α. The third term is similar to the second but corresponds to the right of the DC term in Figure 4.4. In addition, the terms on each side of the DC term are the same but rotated by π radians as in a conventional JTC.

4.2.2 Multispectral example using wavelets

Under certain conditions the input arrangement in Figure 4.3 may produce a multispectral version of the input image corresponding to multiple wavelet scales. We considered an example that produced a version of an input image that corresponded to two wavelet scales. In other words, the output image corresponded to the sum of the correlations of the input image and two different wavelets. To perform this type of operation we referred to Figure 4.3 and set $n = 5$, and $a_3(x, y) = 0$. We considered the input image $f(x, y)$ at locations $a_1(x, y)$ and $a_2(x, y)$, so $f(x, y) = a_1(x, y) = a_2(x, y)$. Finally, we set a wavelet corresponding to one scale $w_1(x, y) = a_4(x, y)$, and the wavelet of another scale $w_2(x, y) = a_5(x, y)$. The configuration in the input plane is shown in Figure 4.5.

The response in the output plane was obtained by substituting the appropriate variables in Eq. 4.1, and the output plane was represented schematically as shown in Figure 4.6. The third term from the DC is the term of interest here. It was the sum of correlations of the input image and two different wavelets.

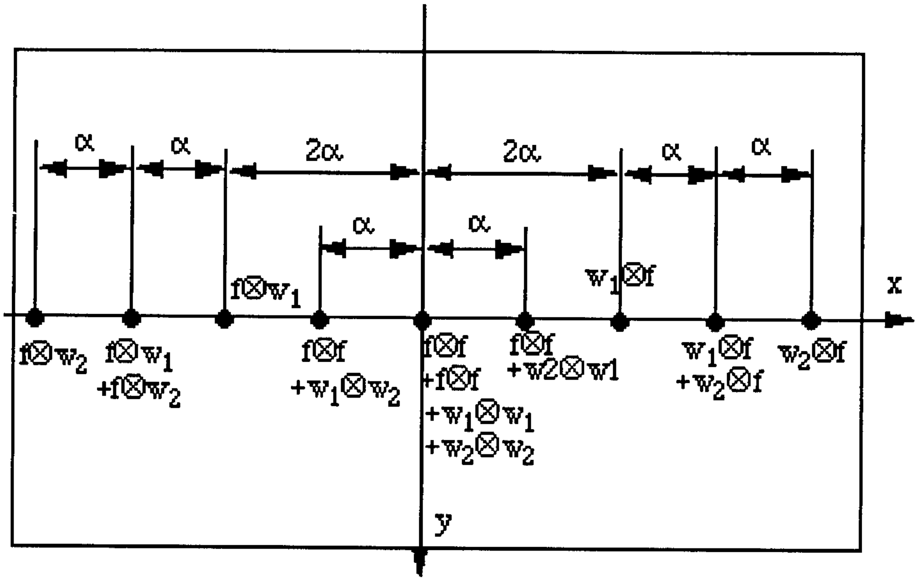

Figure 4.6: Output plane of multispectral example.

4.3 Simulation Experiments

We simulated the operation of a JTC used for multispectral analysis using two scales. We used the input image shown in Figure 4.7, and a Bessel-Gaussian wavelet.[33] The Bessel-Gaussian wavelet was considered because it would be useful for rotation-invariant pattern recognition because it is circularly symmetric. It was described as

$$\frac{1}{\sqrt{a}}w\left(\frac{r}{a}\right) = \frac{1}{\sqrt{a}}J_0\left(\frac{r}{a}\right) exp\left[-\left(\frac{r}{2a\sigma}\right)^2\right] ,$$

where σ is related to the width of the bandpass response, a is the scale factor, and r is the distance form the origin.

We performed the simulation with an input image corresponding to the configuration of Figure 4.5 using a total of 512 x 512 pixels. The image of the plane was used for $f(x, y)$ and contained 50 x 50 pixels and α was set to 50 pixels. The wavelet was used for both $w_1(x, y)$ and $w_2(x, y)$, both with $\sigma = 0.6$, and $a_1 = 0.7$ and $a_2 = 1.2$. The output plane of the correlator is shown in Figure 4.8 with the grayscale inverted for display purposes. The output consisted of nine responses; the three in the center were clipped because of their large value due to the autocorrelation responses and overlapped each other so one large spot appeared. The autocorrelation responses also overlapped the responses immediately next to the autocorrelation responses. The two remaining responses on the right hand side of Figure 4.8 were expanded and shown in Figure 4.9. The response on the right corresponded to $a_2 = 1.2$ and the one on the left corresponded to the sum of a_1 and a_2. Changing a_1 to $a_1 = 1.0$ changed the response on the left but not on the right as shown in Figure 4.10. A small part of the correlation between the image and the wavelet with associated with a_1 can be seen in the left side of the image. Rotating the input image simply rotated the output image because of the circular symmetric nature of the wavelet as shown in Figure 4.11.

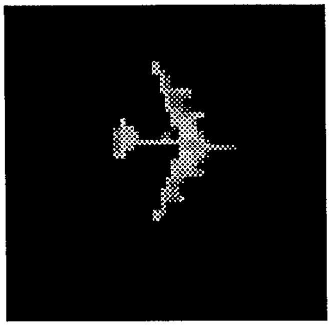

Figure 4.7: Image used in simulation experiments.

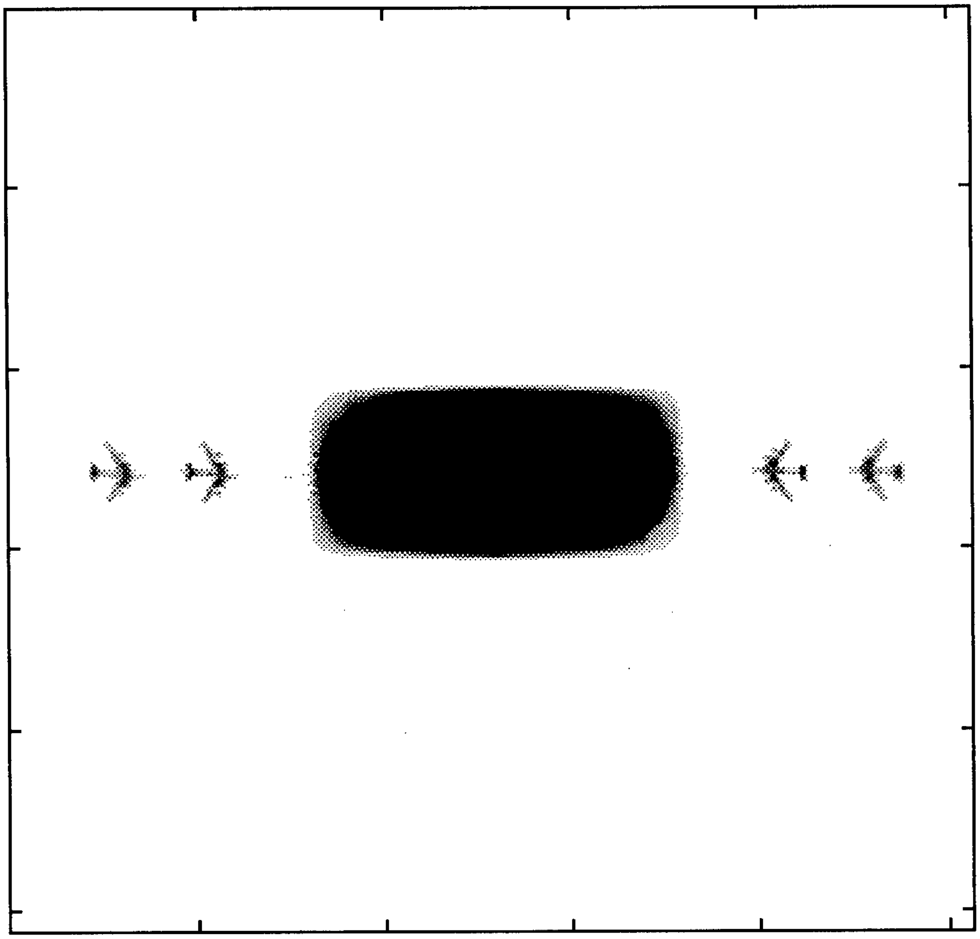

Figure 4.8: Output plane of multispectral simulation experiment.

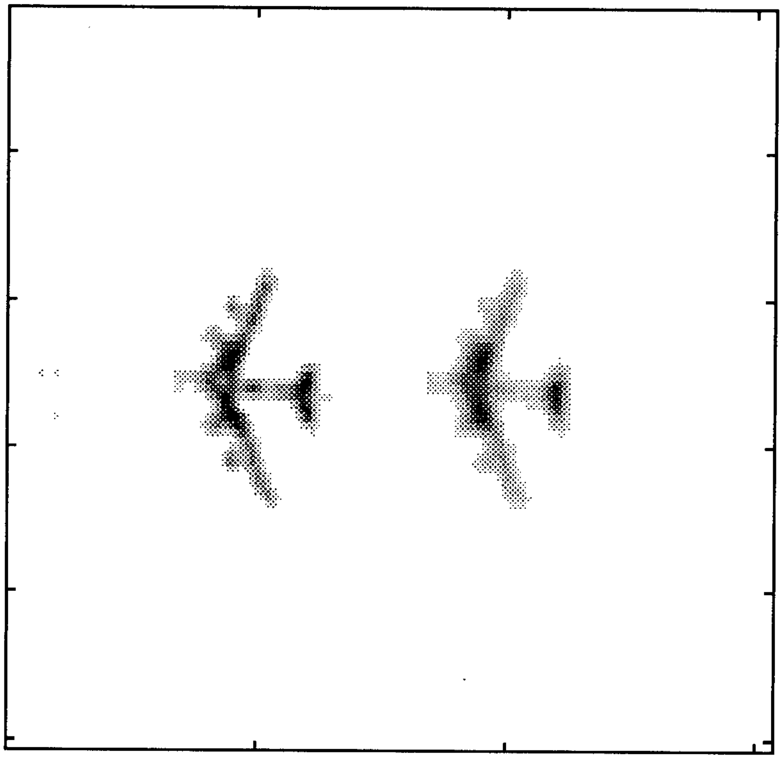

Figure 4.9: Close-up view of right hand side of Figure 4.8. Response on right is for $a_2 = 1.2$, response of left is for $a_1 + a_2$, where $a_1 = 0.7$.

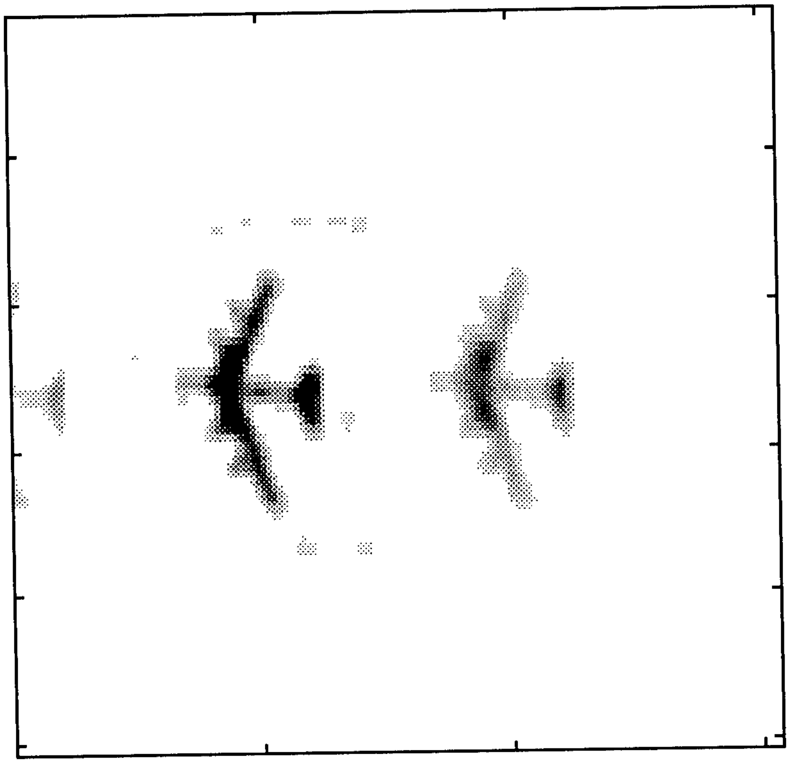

Figure 4.10: Response on right is for $a_2 = 1.2$, response of left is for $a_1 + a_2$, where $a_1 = 1.0$.

4.3.1 Implementation in Optics

To implement the multiple-input JTC optically a few points must be considered. One is the removal of the DC from the wavelet image. Wavelets have zero mean, so the wavelet image cannot be used directly. One solution is to display the wavelet in phase;[33] however, we proposed to use a DC block arrangement.

Because more than one scale of a wavelet is used in our arrangement, multiple wavelet images are needed; one for each scale of the wavelet used. Displaying multiple images on an SLM would increase the space-bandwidth product of the system, so we proposed to use conventional optics to produce different versions of the wavelet at different scales. Similarly for copying the input image.

A proposed multiple-input JTC configuration for the multispectral example was shown in Figure 4.12. It used an input SLM to display an input image and wavelet image as in a conventional JTC. The DC of the wavelet could be eliminated using a DC block in the Fourier transform plane of a telescope arrangement. The scale of a wavelet could be adjusted by varying the focal length of lenses in another telescope arrangement. Neutral density filters may be used to adjust the amplitude of the wavelet. The input lens in the JTC must capture all images, but a detector is required for only a portion of the output plane.

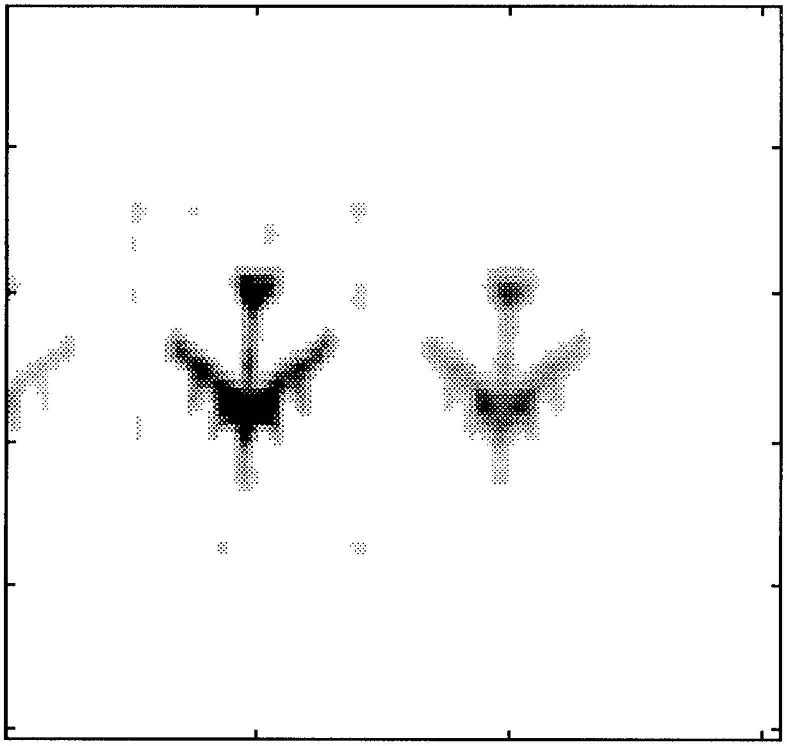

Figure 4.11: Response when input image was rotated by 90 degrees. Response on right is for $a_2 = 1.2$, response of left is for $a_1 + a_2$, where $a_1 = 1.0$.

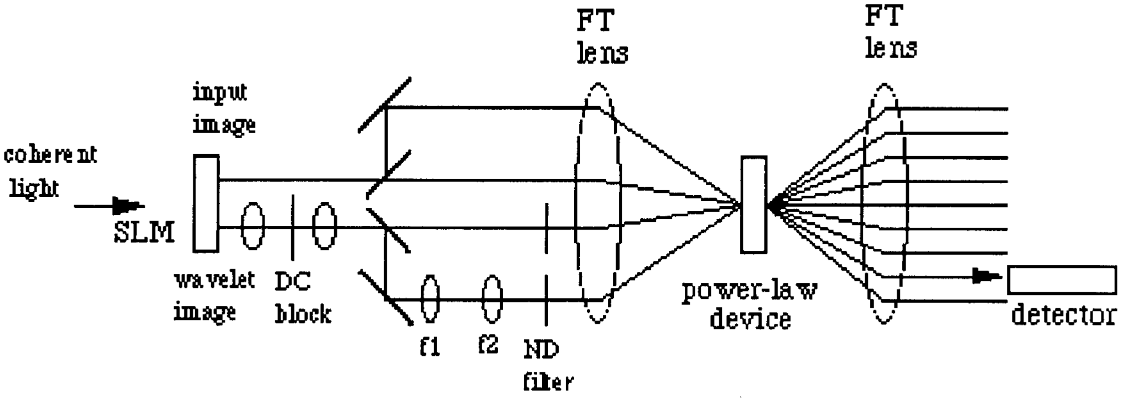

Figure 4.12: Schematic diagram of multiple-input JTC used for multispectral analysis using two wavelet scales.

4.4 Discussion

We showed how a multiple-input JTC can be used for multispectral analysis of an input image. An input image and wavelet image are required as inputs. For m wavelet scales, m versions of the wavelet and m copies of the input image generated using conventional optics are used as inputs to a JTC. The output consists $4m - 1$ correlation results, one of which is the desired output. The space-bandwidth product of the system is the same as for a conventional JTC.

The JTC was simulated using two wavelet scales and 2 identical input images. The wavelet used was a circularly symmetric for rotation-invariance. Rotation of the input image resulted in rotation of the output cross correlation image.

5

CONCLUSIONS

We have demonstrated a binary-input spatially multiplexed parallel optical system capable of implementing a RBF classifier. The system was trained using 600 handwritten digits 0-9, and tested on a disjoint set of 300 digits. The optical system computes the Euclidean distances between a 100 bit input vector and 198 centers in parallel. Our experimental results show that training the RBF network in the presence of optical imperfections and noise versus training the network 'off-line' can improve the testing recognition rate by nearly 60%. When the network is trained 'off-line' and then tested on data from the optical system its recognition rate is 31.0% which agrees well with our computer model's prediction of 29.0%. The testing performance improves considerably when the network is trained 'on-line', using data from the optical system. The experimental recognition rate of 92.67% agrees well with our model's prediction of 94.67% for the adaptive case. In order to implement a real-time system it is necessary to replace the film masks with dynamic SLMs and to design and fabricate an electronic adaptive postprocessor. The primary error source in our system arose from the LCLV used for contrast-reversal. The use of an alternative device is recommended for higher performance and reliability.

The performance of the binary-input optical system demonstrated here is limited primarily by the response time of the contrast reversing SLM. In our case the Hughes device was limited to operation at a page rate of 10 Hz. Utilizing a fast contrast-reversing SLM (e.g., FLC LCLV [34]), would increase the system clock to nearly 250 KHz, resulting in an aggregate computation rate of nearly 20 billion binary operations per second. Although it is straightforward for one to perform the RBF postprocessing functions at this rate using digital electronics, maintaining a minimum signal-to-noise ratio of 100 requires the total optical power to be increased to approximately 0.52 W to accommodate the SLM sensitivity and detector integration time at 250 kHz. The performance of such a modified system is competitive with existing all-electronic solutions and can be further improved through the use of a two-dimensional input SLM together with higher dimensional centers masks.

The first layer of the neural network is composed entirely of parallel processing optics and has been demonstrated for the binary-input classifier. An electronic postprocessor that implements the second layer of the neural network must be fabricated to realize the system's full potential. Two postprocessors versions have been proposed for this system, a simple nonadaptive version and a more complex adaptive version for on-line learning. The nonadaptive processor will be considerably easier to design, fabricate, and test. It is therefore desirable to develop and test the nonadaptive chip prior to designing the full-scale adaptive processor. The results of this design study will aid in the design of future adaptive processors for the in-house system.

The next effort in developing the computer model for the grayscale classifier is to incorporate the grayscale SLM parameters into the computer simulations after the devices have been obtained and characterized. The optical portion of the system is then to be fabricated and compared to the computer model.

We described the optical architecture for a multiple input JTC which performs a multiscale analysis using two wavelet scales. This system will be the focus of future preprocessor work.

One future area of research will be to extend the adaptive optical radial basis function neural network classifier to permit analog and complex-valued input signals. Many signal and image processing applications are best approached by analyzing the inherently complex-valued frequency-domain data. Recent studies have shown that complex-domain neural networks out-perform real-domain neural networks in applications that require the processing of frequency-domain or phase-plane data[35, 18]. One particular application of this proposed complex-input system is radar direction finding.

A new antenna beamforming approach 'neural beamforming', has been developed at Rome Laboratory by researchers Major Jeffrey Simmers and Dr. Hugh Southall of RL/ERAS and Terry O'Donnell from the ARCON Corporation [14, 15, 16, 17]. Their goal is to design neural processing algorithms that can adapt to low cost phased-array antennas, even if the antennas behave in a nonlinear manner, are imperfectly manufactured, or become degraded after some period of time. Neural beamforming techniques can decrease antenna manufacturing and maintenance costs and increase mission time and performance between repair actions. Their work has focused on signal detection and direction finding for single and multiple sources (targets).

The group has presented two neural beamformers, both based on RBF neural networks. Although other neural network paradigms have been previously considered the RBF approaches provide the best overall performance. Both an adaptive RBF (ARBF) network and a linear algebra RBF (LINNET) network training algorithm have been employed in the beamforming application. Most of their work has centered around the adaptive RBF neural beamformer that uses a training procedure similar to one used for our adaptive optical system. The input to the neural beamformer consists of complex-valued data from the output of a 32 element phased-array antenna. It should be noted that it is desirable to perform that direction finding at MHz data rates and military phased array antennas typically have between 200 and 1000 antenna elements. These data processing rates are not attainable with current or near-term sequential computer hardware systems. The group is now beginning to search for parallel processing hardware in order to meet this signal processing demand. An analog adaptive optical RBF classifier system may present a viable solution to this large, real-time problem.

REFERENCES

[1] T. Poggio and F. Girosi. Networks for Approximation and Learning. *Proc. of the IEEE*, 78(9):1481–1497, September 1990.

[2] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, 18 Jan 1990.

[3] G. Vrckovnik, C. Carter, and S. Haykin. Radial Basis Function Classification of Impulse Radar Waveforms. In *Proc. of IJCNN*, volume I, pages 45–50, June 1990.

[4] R. Brunelli and T. Poggio. Face Recognition: Features versus Templates. *IEEE Trans. Patt. Anal. Machine Intell.*, 15(10):1042–1052, 1993.

[5] N. Arad, N. Dyn, D. Reisfeld, and Y. Yeshurun. Image Warping by Radial Basis Functions: Application to Facial Expressions. *CVGIP*, 56(2):161–172, 1994.

[6] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellappa, and C. L. Wilson. Evaluation of Pattern Classifiers for Fingerprint and OCR Applications. *Pattern Recognition*, 27(4):485–501, 1994.

[7] S. Renals and R. Rohwer. Phoneme Classification Experiments Using Radial Basis Functions. In *Proc. of IJCNN*, volume I, pages 461–467, Wash. D.C., June 1989.

[8] Yuchun Lee. Handwritten Digit Recognition Using K Nearest-Neighbor, Radial Basis Function, and Backpropagation Neural Networks. *Neural Computation*, 3:440–449, 1991.

[9] M. A. Neifeld, S. Rakshit, and D. Psaltis. Handwritten zip code recognition using an optical radial basis function classifier. In *Proc. of SPIE*, volume 1469, pages 250–255, 1991.

[10] W. E. Foor and M. A. Neifeld. Adaptive, optical, radial basis function neural network for handwritten digit recognition. *Appl. Opt.*, 34(32):7545–7555, 1995.

[11] D. S. Broomhead and D. Lowe. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2:321–355, 1988.

[12] D. S. Broomhead. Signal processing for nonlinear systems. In *Proc. SPIE*, volume 1565, pages 228–243, 1991.

[13] L. Xu, A. Krzyzak, and A. Yuille. On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates, and Receptive Field Size. *Neural Networks*, 7(4):609–628, 1994.

[14] T. O'Donnell, J. Simmers, and H. Southall. An Introduction to Neural Beamforming. In *IEEE Dual-Use Technologies and Applications Conference*, volume I, pages 483–492, May 1994.

[15] J. Simmers, H. Southall, and T. O'Donnell. Advances in Neural Beamforming. In *Proc. of the 1993 Antenna Applications Symposium*, volume 1, pages 203–216, Univ. of Illinois, Sep 1993.

[16] T. O'Donnell, J. Simmers, and D. Jacavanco. Neural Beamforming for Phased-Array Antennas. In *Proc. of the 1992 Antenna Applications Symposium*, volume 1, pages 106–115, University of Illinois, Sep 1992.

[17] J. Simmers and T. O'Donnell. Adaptive RBF Neural Beamforming. In *IEEE Command, Control, Communications and Intelligence Technology Applications Conference*, pages 94–98, June 1992.

[18] S. Chen, P. M. Grant, S. McLaughlin, and B. Mulgrew. Complex-Valued Radial Basis Function Networks. In *Proc. Third IEE Int'l Conf. Artif. Neural Networks*, pages 148–152, May 1993.

[19] J. Anderson, J. Platt, and D. Kirk. *An Analog VLSI Chip for Radial Basis Functions*, pages 765–772. Morgan Kaufmann, 1993.

[20] M. A. Neifeld and D. Psaltis. Optical implementations of radial basis classifiers. *Appl. Opt.*, 32(08):1370–1379, 10 Mar 1993.

[21] W. E. Foor, H. G. Andrews, and M. A. Getbehead. Optical Correlator Optimizations and Extensions for Automated Optical Target Recognition. Rome Laboratory Technical Report RL-TR-94-220, December 1994.

[22] W. E. Foor. Adaptive Optical Radial Basis Function Neural Network Classifier. Rome Laboratory Technical Report RL-TR-94-239, December 1994.

[23] M. A. Neifeld, S. Rakshit, A. Yamamura, S. Kobayashi, and D. Psaltis. Optical disk implementation of radial basis classifiers. In *Proc. of SPIE*, volume 1347, pages 4–15, 1990.

[24] C. S. Sexton. Current Status of Hughes Liquid Crystal Light Valve Performance for Optical Data Processing. In *Proc. of SPIE*, volume 684, pages 96–100, 1986.

[25] G. Moddel, K. M. Johnson, W. Li, R. A. Rice, L. A. Pagano-Stauffer, and M. A. Handschy. High-speed, binary optically addressed spatial light modulator. *Appl. Phys. Lett.*, 55:537–539, 1989.

[26] B. K. Taylor and D. P. Casasent. Error-source effects in a high-accuracy optical finite-element processor. *Appl. Opt.*, 25(6):966–975, 1986.

[27] J. F. Ebersole. Optical image subtraction. *Opt. Eng.*, 14(5):436–447, 1975.

[28] D. Gabor, G. W. Stroke, R. Restrick, A. Funkhouser, and Brumm D. Optical image synthesis (complex amplitude addition and subtraction) by holographic Fourier transformation. *J. Opt. Soc. Am.*, 60(8):1037–1041, 1970.

[29] S. H. Lee, S. K. Yao, and A. G. Milnes. Optical image synthesis (complex amplitude addition and subtraction) in real time by a diffraction-grating interferometric method. *J. Opt. Soc. Am.*, 60(8):1037–1041, 1970.

[30] E. Marom. Real-time image subtraction using a liquid crystal light valve. *Opt. Eng.*, 25(2):274–276, 1986.

[31] H. K. Liu and T. H. Chao. Optical image subtraction techniques, 1975–1985. In *Proc. SPIE*, volume 638, pages 55–65, 1986.

[32] S. Mallat and S. Zhong. *Wavelet Maxima Representation*, pages 207–284. Masson and Springer-Verlag, Berlin, 1991.

[33] P. S. Erbach, D. A. Gregory, and X. Yang. Optical wavelet transform by the phase-only joint-transform correlator. *Appl. Opt.*, 35(17):3117–3126, 1999.

[34] I. Abdulhalim, G. Moddel, and K. M. Johnson. High speed analog spatial light modulator using a hydrogenated amorphous silicon photosensor and an electroclinic liquid crystal. *Appl. Phys. Lett.*, 55:1603–1605, 1989.

[35] T. Masters. *Signal and Image Processing with Neural Networks.* Wiley, 1994.

☆U.S. GOVERNMENT PRINTING OFFICE: 1997-610-130-61091

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.