FEATURE EXTRACTION WITH THE FRACTIONAL FOURIER TRANSFORM

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Özgür Güleryüz
September 1998

TK
7872
·F5
G85
1998

FEATURE EXTRACTION WITH THE FRACTIONAL FOURIER TRANSFORM

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Özgür GÜLERYÜZ

By

Özgür Güleryüz

September 1998

TK
7872
.F5
.G85
1998
B 044007

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Haldun M.Özaktaş(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Orhan Arıkan

tarafından

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

FEATURE EXTRACTION WITH THE FRACTIONAL FOURIER TRANSFORM

Özgür Güleryüz
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Haldun M. Özaktaş
September 1998

In this work, alternative design and implementation techniques for feature extraction applications are proposed. The proposed techniques amount to decomposing the overall feature extraction problem into a global linear system followed by a local nonlinear system. Different output representations for representation of input features are also allowed and used in these techniques. These different output representations bring an additional degree of freedom to the feature extraction problems. The systems provide multi-outputs consisting of different features of the input signal or image. Efficient implementation of the linear part of the system is obtained by using fractional Fourier filtering circuits. Expressions for the proposed techniques are derived and several illustrative examples are given in which efficient implementations for feature extraction applications are obtained.

Keywords: Image analysis, feature extraction, fractional Fourier transform, fractional Fourier domains, fractional Fourier filtering circuits.

ÖZET

KESİRLİ FOURIER DÖNÜŞÜMÜ İLE ÖZNİTELİK BULMA

Özgür Güleryüz
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Dr. Haldun M. Özaktaş
Eylul 1998

Bu tezde öznitelik bulma yöntemleri için tasarım ve gerçekleştirme yöntemleri sunulmuştur. Önerilen yöntemler öznitelik bulma sistemlerini genel doğrusal sistemi takip eden yerel doğrusal olmayan sistemler şeklinde modeller. Giriş bilgileri çıkışta değişik biçimlerde temsil edilebilmektedir. Giriş bilgilerinin çıkışta değişik biçimlerde temsil edilebilmesi öznitelik bulma problemlerine fazladan bir özgürlük katmaktadır. Ayrica, önerilen yöntemler birden fazla çıkış sağlama özelliğine de sahiptir. Önerilen sistemin doğrusal kısmının verimli bir şekilde gerçekleştirilmesi kesirli Fourier süzgeç devreleri kullanımı ile sağlanmıstır. Önerilen yöntemlerin ifadeleri türetilmiş ve öznitelik bulma uygulamalarında verimli sonuçlar elde edildiği örneklerle gösterilmiştir.

Anahtar Kelimeler: İmge incelenmesi, öznitelik bulma, kesirli Fourier dönüşümü, kesirli Fourier domenler, kesirli Fourier süzgeç devreleri.

ACKNOWLEDGEMENT

I gratefully thank my supervisor Assoc. Prof. Dr. Haldun Özaktaş for his suggestions, supervision and guidance throughout the development of this thesis.

I would also like to thank Assoc. Prof. Dr. Orhan Arıkan for his suggestions, and Prof. Dr. Enis Çetin, the members of my jury, for reading and commenting on the thesis.

Many thanks to all my friends, especially to Hakan, Onur, Çağatay and Sadullah, for their valuable discussions, help and friendship.

It is a pleasure to express my special thanks to my mother, father and sister for their sincere love, support and encouragement.

Contents

List of Figures

To my family, and close friends...

Chapter 1

Introduction

The aim in a large number of image processing applications is to extract some features from image data. These features may be used not only to provide a description, interpretation of the scene present in that image, but also to understand the scene [1]. This area of image processing is called image analysis. In a text-reading problem, image analysis should be used to recognize the characters. Image analysis is employed in radar imaging to detect and identify a target. In medical imaging size and shape of internal organs may be obtained by using image analysis. The examples of application areas of image analysis may easily be enlarged.

One of the basic tools of image analysis is matching the scene in the image to a previously known scene. But matching is not generally enough in most image analysis applications. In most applications, some features from the input image must be extracted. Many tools and methods are developed for this purpose, but still a huge amount of research is being done to increase the performance and decrease the cost of these tools.

Extracting features from an input signal or image is generally a nonlinear operation. The nonlinear systems are difficult to implement, especially by optical systems. Some features can be extracted by using linear systems, but most of the features can only be extracted by nonlinear systems. Therefore, an efficient way of implementing these nonlinear systems should be obtained. It should also be noted that as the implementation costs of linear systems are generally high a system that provides easy and efficient implementations for feature extraction applications would be useful.

In this thesis, new design and implementation techniques for image analysis problems are proposed. The proposed techniques have fast implementations. The design and implementation of these techniques are presented and their performances are compared with the performances of the previously used methods. Simulation results on some image analysis examples are presented. It should be noted that the system can be applied to all feature extraction problems, but it will be applied only to some simple feature extraction problems in this thesis. Application of this system to more complicated feature extraction problems is considered as the future work.

The rest of Chapter 1 gives the motivation to use the proposed methods for image analysis. Chapter 2 provides a survey about the commonly used techniques for recognition and image analysis. In Chapter 3, fractional Fourier transform and fractional correlation is examined in detail. Chapter 4 provides the structural and mathematical details of fractional Fourier filtering circuits. In Chapter 5, the proposed technique for image analysis applications and the proposed implementation techniques are presented. Chapter 6 provides the simulation results on some image analysis examples and performance analysis of different techniques. Chapter 7 gives the conclusions and future directions for research.

1.1 Motivation

The research that resulted in this thesis is motivated by the enormity of the applications that employ image analysis. Image analysis is applied in many problems to discover, identify and understand patterns that are relevant to the performance of an image-based task. In some applications, it is enough to figure out the existence, and position of a pattern. For such cases, matched filtering is the most commonly used tool. But, its application areas are restricted with these applications. Also its performance degrades rapidly when distortions are present in the input image. Therefore, some other tools which are invariant of these input distortions are developed. These algorithms are generally fast and easy to implement, but their applications are limited with problems of checking the presence of a certain pattern. But, the main purpose of image analysis is to identify some features of the input image. Therefore, a tool which can figure out certain properties of the input pattern, is fast and has an easy implementation, would be useful.

In this thesis, two implementation techniques, which can provide solutions to some image analysis problems and which can be implemented with low implementation costs are proposed.

In a medical image analysis problem like the measurement of size and shape of internal organs, a system like the one we propose may be applied to provide the sizes, orientation and position of internal organs. Similarly in a Robotics application, such a system may be used to recognize and interpret objects in a scene. Another example may be given as a radar imaging problem in which a target should be detected and its properties such as position, orientation should be obtained. The proposed method can be used to obtain these properties.

Most commonly used image analysis tools such as matched filtering are implemented as a single filtering operation, but they can only provide simple information at the output such as existence or position of a specific pattern in the input scene. It would be advantageous to have a system that provides more than one feature of the input scene at the output. These features may be displayed with a display panel, where each display shows a different feature of the input pattern.

Feature extraction operation is generally a nonlinear operation. It is difficult to implement nonlinear systems especially by optical systems. But it is easy to implement linear systems by optical systems. In this thesis, we will try to decompose the overall image analysis system into a global linear system, which can be implemented optically, and a local nonlinear system, which can be implemented by simple electronic circuitry. The linear system is called as the global linear system as it can be applied to all points in the input plane and provides the results for all output points when the system is implemented optically. The nonlinear system is called as a local nonlinear system as it is difficult to make nonlinear operations on the whole output plane. The nonlinear operations are restricted to be applied on some local points at the output plane. The proposed system provides efficient implementation alternatives for feature extraction problems. The proposed system structure is shown in Figure 1.1.

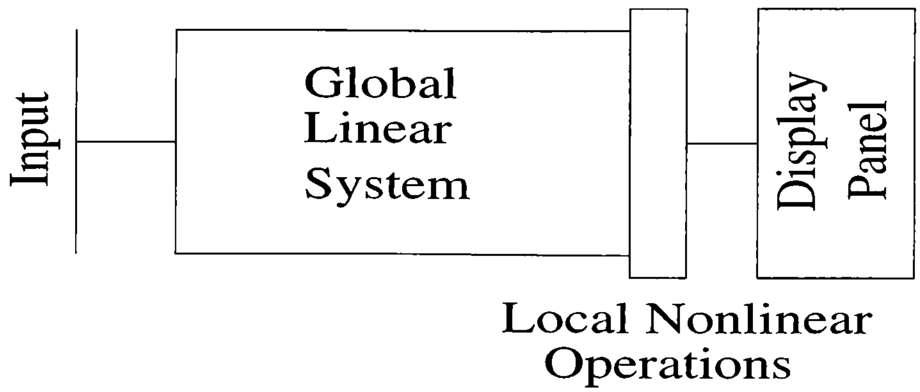

Figure 1.1: Proposed System Structure

The implementation costs of linear systems are generally high. Therefore, the linear part of the proposed system will be implemented with fractional Fourier filtering circuits to decrease the implementation costs with acceptable decreases in accuracy.

Chapter 2

Methods Used for Recognition

In this chapter of the thesis, some of the most commonly used methods used for recognition and image analysis will be examined in detail. Matched filtering is an important tool used to detect a target in the presence of noise. It has many applications in recognition problems, but it has some disadvantages [2]. Therefore some other techniques like Distortion Invariant Filtering or Synthetic Discriminant Functions are developed. A similar tool, fractional correlation, which provides shift-variant recognition property is introduced in the following chapter. These most commonly used techniques will be examined in this chapter in detail.

2.1 Matched Filtering

The matched filter is the optimum linear filter for maximizing the ratio of peak signal to mean-square noise when we process a deterministic signal in the presence of additive, stationary random noise. Consider a filter with an input function given by:

$$x(t) = g(t) + w(t) \tag{2.1}$$

where $w(t)$ is additive noise. Since the filter is linear, the resulting output $y(t)$ may be expressed as:

$$y(t) \;=\; g_o(t) + n(t) \tag{2.2}$$

where $g_o(t)$ and n(t) are produced by the signal and noise components of the input $x(t)$ respectively. The peak pulse signal-to-noise ratio can be expressed as:

$$\eta \;=\; \frac{\mid g_o(T) \mid^2}{E[n^2(t)]} \tag{2.3}$$

where $\mid g_o(T) \mid^2$ is the instantaneous power in the output signal at time T and $E[n^2(t)]$ is a measure of the average output noise power. In matched filtering, the requirement is to specify the impulse response $h(t)$ of the filter such that the output signal-to-noise ratio is maximized at the same point T. We know that:

$$\begin{aligned}
g_o(t) &= \int_{-\infty}^{+\infty} H(f)G(f)exp(j2\pi ft)df & (2.4)\\
\mid g_o(T) \mid^2 &= \mid \int_{-\infty}^{+\infty} H(f)G(f)exp(j2\pi fT)df \mid^2 & (2.5)\\
E[n^2(t)] &= \int_{-\infty}^{+\infty} S_N(f)df = \int_{-\infty}^{+\infty} N(f) \mid H(f) \mid^2 df & (2.6)\\
\eta &= \frac{\mid \int_{-\infty}^{+\infty} H(f)G(f)exp(j2\pi fT)df \mid^2}{\int_{-\infty}^{+\infty} N(f) \mid (f) \mid^2 df} & (2.7)
\end{aligned}$$

where $N(f)$ is the noise power spectral density. The ratio is maximized when:

$$H(f) = \frac{G^*(f)exp(-j2\pi fT)}{N(f)} \tag{2.8}$$

For the white noise the transfer function and impulse response reduces to:

$$\begin{aligned}
H_{opt}(f) &= kG^*(f)exp(-j2\pi fT) & (2.9)\\
h_{opt}(t) &= k\int_{-\infty}^{+\infty} G^*(f)exp(-j2\pi f(T-t))df & (2.10)\\
h_{opt}(t) &= kg^*(T-t) & (2.11)
\end{aligned}$$

It is known that, matched filtering is equal to correlating the input and reference functions. In order to show this, let us consider a matched filtering application. The input of the system is $x(t)$ and the output of the filter $h(t)$

is given by $y(t)$. We can express the relation between $x(t)$ and $y(t)$ by the convolution integral given by:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau \tag{2.12}$$

Note that $h(t) = \phi^*(T-t)$ where ϕ is the reference function to which h is matched. So:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)\phi^*(T-t+\tau)d\tau \tag{2.13}$$

Sampling this output at time t=T we get :

$$y(T) = \int_{-\infty}^{+\infty} x(\tau)\phi^*(\tau)d\tau \tag{2.14}$$

which is the value of the cross-correlation of $x(t)$ and $\phi(t)$ at $t = T$. Therefore we can say that matched filtering equals to correlating the input and reference functions in the time domain.

2.1.1 Limitations of Matched Spatial Filtering

Matched Spatial Filtering has two major limitations [3]:

1- The output correlation peak degrades rapidly with geometric image distortions.

2- The Matched Spatial Filter (matched to one given image) cannot be used for multi class pattern recognition.

In order to overcome these limitations, the concept of matched filtering has been greatly extended by several types of generalized filters. These methods can be classified in 2 categories:

1- Distortion Invariant Filtering : Concerns in-plane 2-D scaling and rotation distortions. Such methods include the use of space-variant transforms and circular harmonic functions. In these techniques, the intensity at the origin of correlation function can not generally be specified during synthesis.

2- Synthetic Discriminant Functions : These category filters use training images that are sufficiently descriptive and representative of the expected distortions. These filters can be viewed as generalizations of matched spatial filters for the identification of multiple targets in the presence of virtually any type of distortion. The intensity at the center of the cross-correlation function (defined as the filter output) can be specified for each training image during synthesis, and several objects can be handled by one filter by including all object classes in the training set.

2.2 Distortion Invariant Filtering

The main disadvantage of matched filtering is that it provides no invariant properties (such as scale or rotation) except shift invariance. Such kind of invariances can be obtained by using harmonic expansions. In order to use the harmonic expansions, the object is first decomposed into a certain orthogonal harmonic expansion, and a single order of the expansion is chosen as the filter. The SNR performance of these systems is high as the chosen harmonic contains a significant amount of object's energy. Generally circular harmonic expansions are used for rotation invariance and radial harmonics are used for scale invariance [4,5]. Synthetic Discriminant Functions, which will be examined in detail in next section, are also a kind of distortion invariant filtering.

2.2.1 Scale Invariance

As it is told before, radial harmonics are the most commonly used tools to maintain scale-invariance [4]. In Mellin radial expansions, a function $f(r,\theta)$ is expanded into a set of orthogonal functions $[r^{i2\pi N-1}]$, which is mathematically given by:

$$\begin{aligned} f(r,\theta : x_0, y_0) &= \sum_{N=-\infty}^{\infty} f_N(\theta; x_0, y_0) r^{i2\pi N-1} & (2.15) \\ f_N(\theta; x_0, y_0) &= \frac{1}{L}\int_{r_0}^{R} f(r,\theta; x_0, y_0) r^{-i2\pi N-1} r dr & (2.16) \end{aligned}$$

where r_0 and R are minimal and maximal radius of the object, L is an integer number satisfying $r_0 = Re^{-L}$, N is the order of expansion, (x_0, y_0) is the Cartesian coordinate origin of the (r, θ) polar coordinates. In practice the r_0 value should be chosen optimal. If it is chosen too big then the areas of the object would not effect the harmonic f_M and if it is chosen too small then the filter $f_M(\theta)r^{i2\pi M-1}$ would have too high spatial frequencies.

After expanding the input function $F(r, \theta)$ into its radial harmonics, the Mth harmonic $f_M(\theta)r^{i2\pi M-1}$ is chosen as the filter function and the filter output is obtained by correlating the input function with the filter function. The output of the filter is scale-invariant.

2.2.2 Rotation Invariance

For rotation invariance, the input object is expanded into a set of orthogonal functions $[\exp(iN\phi)]$, called the circular harmonics [4-6]. The expansion is given mathematically by:

$$f(\rho, \phi) = \sum_{N=-\infty}^{\infty} f_N(\rho) \exp(iN\phi) \tag{2.17}$$

$$f_N(\rho) = \frac{1}{2\pi} \int_0^{2\pi} f(\rho, \phi) \exp(-iN\phi) d\phi \tag{2.18}$$

where $f(\rho, \phi)$ is the input function $f(x, y)$ in circular coordinates and N is the expansion order.

Similar to the above cases, the filter function is again chosen to be a single harmonic given by:

$$g(\rho, \phi) = f_M \exp(iM\phi) \tag{2.19}$$

where

$$f_M(\rho) = \frac{1}{2\pi} \int_0^{2\pi} f(\rho, \phi) \exp(-iM\phi) d\phi \tag{2.20}$$

The filter output is the correlation of the input and the filter functions and it is rotation invariant.

2.2.3 Advantages and Disadvantages of Distortion Invariant Filtering

As previously explained, the matched filter is optimum to detect a target in white noise, but its performance decreases rapidly when there are distortions like scale, projection and rotation. Therefore distortion invariant filtering is developed to overcome these problems. They provide distortion invariance but they have poor discrimination ability as they have a lower information content than the matched filter. Because, the input function is expanded into its harmonics and a single harmonic is chosen as the filter function. Therefore the information included in the other harmonics is lost, which causes a loss in the discrimination ability. Synthetic Discriminant Functions are introduced to obtain distortion invariant filters that have better discrimination availability. Synthetic Discriminant Functions will be examined in detail in the next section.

2.3 Synthetic Discriminant Functions

Synthetic Discriminant Function (SDF) technique is used to design filters which have the flexibility of achieving invariance to any type of distortion provided that a sufficiently descriptive training set of images is available [3, 7–12]. The filter designed with this technique yields specified output values with each of the training images. So the training image set should be chosen such that it represents the expected distortions. In addition to SDF's, generalized matched filtering and composite filter techniques are developed by using this kind of methodology [3].

2.3.1 Conventional Synthetic Discriminant Functions

In order to explain the design of SDF's, let $x_1, x_2, ..., x_N$ denote N column vectors of dimension d representing N images, each with d pixels in it. Generally the number of pixels d is much larger than N, the number of training images.

If we want to recognize only one class of image, we choose all the training images from that set. On the other hand, if we want to design a filter which can discriminate between two classes, then the training images are chosen from both of the classes [3, 11].

In the conventional SDF technique, the filter is matched to an image h, which is a linear combination of the training vectors.

$$h = Xa \tag{2.21}$$

where X is the matrix containing the N images $x_1, x_2, ..., x_N$ as its columns, and a is the vector containing the weights of each output image.

The constraint vector a should be chosen such that the following constraints are satisfied.

$$X^+h = u^*, \tag{2.22}$$

where + denotes the conjugate transpose, * denotes the transpose and u is the desired output vector.

If an SDF satisfying the above constraints can be found, it is guaranteed that the cross-correlation value at the origin will be the desired constraints. The constraint vector a can be found by substituting Eq.(2.22) into Eq.(2.21)

$$a = (X^+X)^{-1}u^* \tag{2.23}$$

then h can be found as :

$$h = X(X^+X)^{-1}u^* \tag{2.24}$$

Note that X^+X is an N*N symmetric matrix. If X^+X is invertible, then a can easily be obtained as $(X^+X)^{-1}u$. But if X^+X is not full rank, we will have infinite solutions if it is consistent and will have no solution when it is not consistent. For X^+X to be full rank, the N training images should form a linearly independent set. But generally the training images are not linearly independent, so some preprocessing techniques such as Gram-Schmidt orthonormalization procedure may be used to form a linearly independent set.

Choosing the synthetic discriminant function h as the linear combination of the input images is unnecessary. To remove this ambiguity, a more general definition of h can be given as:

$$h = X(X^+X)^{-1}u^* + [I_d - X(X^+X)^{-1}X^+]z \qquad (2.25)$$

where I_d is the $d*d$ identity matrix and z is any column vector with d complex entries.

Synthetic Discriminant Function seems as if it is a very attractive tool for optical correlation and recognition. But we have some important problems with these conventional SDF's, which may be eliminated by some improvements.

2.3.2 Problems with the Conventional Synthetic Discriminant Functions

The first problem with the conventional SDF is that, it does not consider the occurrence of random noise in the input. As an example, if we examine a two-class problem, we may design our filter such that the correlation peak value at the origin is 1 for one class and 0 for the other class. In the presence of noise the output values will not exactly be 0 or 1. Therefore we must design filters which can tolerate input noise. Minimum Variance Synthetic Discriminant Function (MVSDF) is one of the approaches employed in order to achieve this goal [3, 12].

The second problem with the conventional synthetic discriminant functions arises as we control only one point (the origin) in the correlation output plane. Correlators are widely employed as they not only detect a target, but also locate it. In the conventional synthetic discriminant functions, it is not easy to locate the target if it is shifted by an unknown amount. Because as we control only the output value at the origin, the output value other than at the origin can take any value which may cause many points to have the desired output value for the incoming image. Therefore it is impossible to find the exact location of the target. In order to overcome this difficulty, we must design filters which produce sharp correlation peaks. Minimum Average Correlation Energy (MACE) Filters is a kind of design procedure to overcome this difficulty.

The third problem arises from the selection of the filter, h as a linear combination of the training images. Because this assumption forces us to have a linear equation set consisting of much more unknowns than the number of equations. By forcing h to be a linear combination of the training images, we are throwing out huge amount of degrees of freedom and find a unique solution. We can use these degrees of freedom to achieve objectives such as peak sharpness and noise minimization.

2.3.3 Minimum Variance Synthetic Discriminant Functions

The Minimum Variance Synthetic Discriminant Function (MVSDF) aims to find SDF's to minimize the variance in the output when the input image is a noisy training image [3, 12].

Consider the situation when the input image is one of the training images x_i corrupted by additive noise n. Then the value of the cross-correlation at the origin, y is given by:

$$y = h^+(x_i + n) = u_i + h^+n \tag{2.26}$$

where u_i is the desired output and $+$ is the complex conjugate.

Assuming that the real-noise vector n is a zero-mean vector with a $d * d$ covariance matrix Σ, the variance of the output is caused by h^+n and can be expressed as:

$$\sigma_y^2 = E[| h^+n |^2] = E[h^+nn^+h] = h^+\Sigma h \tag{2.27}$$

minimizing σ_y^2 with respect to the constraints in Eq 2.22. leads to the following MVSDF:

$$h = \Sigma^{-1}X(X^+\Sigma^{-1}X)^{-1}u^* \tag{2.28}$$

where $u = [u_1, ..., u_N]^T$ and $X = [x_1, ..., x_N]$ is the matrix containing the training images as its columns.

For the special case of white noise MVSDF is equal to the conventional SDF, but for other noise types, the conventional SDF is not the optimal solution.

Similar to the conventional SDF, MVSDF has some problems. The MVSDF, like conventional SDF, controls only the output value at the origin. Therefore we may observe large side lobes in the correlation output, which causes finding the location of the target to be more difficult. Another problem arises as we generally do not know the the noise covariance matrix Σ exactly. In some cases although we know Σ exactly, it may not be invertible. Some approaches like Frequency-Domain Synthetic Discriminant Functions are applied to overcome this problem.

2.3.4 Minimum Average Correlation Energy Filters

The Minimum Average Correlation Energy (MACE) Filters are capable of producing sharp correlation peaks which are good for location accuracy and discrimination [3, 10].

Assuming that $X_i(u,v)$ is the two-dimensional Fourier transform of the ith training image x_i and $H^*(u,v)$ is the transmittance of the filter function, the following constraint should be satisfied:

$$\int\int X_i(u,v)H^*(u,v)dudv = u_i, \tag{2.29}$$

where $i = 1, ..., N$.

The MACE filters should minimize the average correlation energy given by:

$$E_{ave} = \frac{1}{N}\sum_{i=1}^{N}\int\int \mid X_i(u,v) \mid^2 \mid H^*(u,v) \mid^2 dudv \tag{2.30}$$

By using the vector notation, E_{ave} can be expressed as:

$$E_{ave} = h^+Dh \tag{2.31}$$

where D is a $d * d$ diagonal matrix whose entries are obtained by averaging $\mid X_i(u,v) \mid^2$, i=1,...,N and then scanning the average from left to right and from top to bottom.

The filter minimizing E_{ave} and satisfying the constraints in Eq 2.26. is found as:

$$h_{MACE} = D^{-1}X(XD^{-1}X)^{-1}u^* \tag{2.32}$$

Similar to conventional SDF's and MVSDF's, the minimum average correlation energy filters have some problems. Firstly, the MACE filters are more sensitive to intraclass variations than the other types of composite filters. The second problem is that these filters are designed without considering noisy input conditions. Therefore these filters are not much tolerable to noise.

Chapter 3

Fractional Fourier Transform and Fractional Correlation

It is known that correlation is a useful tool for pattern recognition or comparison which can easily be implemented optically [13]. The conventional correlation operation is a shift-invariant operation, which means that shifting the input by a certain amount causes the same shift at the output plane. For many cases this property is necessary, but for some applications it is unnecessary and it can even disturb the recognition. As an example, we can give a situation in which an object should be recognized when it is placed inside a certain area and rejected otherwise. For these cases fractional correlation is more useful than the conventional correlation [2, 14–16]. Optical fractional correlation can easily be implemented with a similar setup to the conventional correlation [2, 14, 17–19].

3.1 Fractional Fourier Transform

The fractional Fourier Transform is known to be a shift-variant transform [20, 21]. Therefore, the fractional Fourier transform is used to maintain the shift-variant property in fractional correlation.

The ath order fractional Fourier transform $f_a(u)$ of the function $f(u)$ is a linear operation defined by the integral :

$$f_a(u) = \int_{-\infty}^{\infty} K_a(u,u')f(u')du' \tag{3.1}$$

$$K_a(u,u') = \begin{cases} A_\alpha \exp[i\pi(\cot\alpha u^2 - 2\csc\alpha uu' + \cot\alpha u'^2)] & a \neq 2j \\ \delta(u-u') & a = 4j \\ \delta(u+u') & a = 4j \mp 2 \end{cases} \tag{3.2}$$

where j is an integer and $\alpha = a\pi/2$ and $A_\alpha = \sqrt{1 - i\cot\alpha}$. It should be noted that the square root is defined such that the argument of the result lies in the interval $(-\pi/2, \pi/2]$.

The fractional Fourier transform is by definition linear, but it is not shift-invariant unless $a = 2j$.

It can be observed that the 0th order transform of $f(u)$ is simply $f_0(u) = f(u)$ itself and the 1st order transform $f_1(u) = F(u)$ is the ordinary Fourier transform. Likewise, it can easily be seen that $f_{-1}(u)$ is the ordinary inverse Fourier transform of $f(u)$. Also, the $\pm$2nd order transform is equal to $f(-u)$ by definition.

The fractional Fourier transform satisfies the index additivity. That is, the a_1th order fractional Fourier transform of the a_2th order fractional Fourier transform is equal to the $(a_1 + a_2)$th transform.

The ath order fractional Fourier transform of $f(u)$ is also called as the representation of $f(u)$ in the ath fractional Fourier domain. It can also be said that the ath fractional Fourier domain makes an angle $\alpha = a\pi/2$ with the time (or space) domain in the time-frequency (or space-frequency) plane [22, 23].

3.2 Fractional Correlation

The conventional correlation of $u_0(x)$ and $v_0(x)$ is defined as:

$$C_1(x) = \int_{-\infty}^{\infty} u_0(x_0)v_0(x_0 - x)^\star dx_0 \quad (3.3)$$

$$C_1(x) = \int_{-\infty}^{\infty} u_1(\nu)v_1(\nu)^\star exp(2\pi i\nu x)d\nu \quad (3.4)$$

where $u_1(\nu) = F^1 u_0(x)$,$v_1(\nu) = F^1 v_0(x)$, F^1 is the Fourier transform operation.

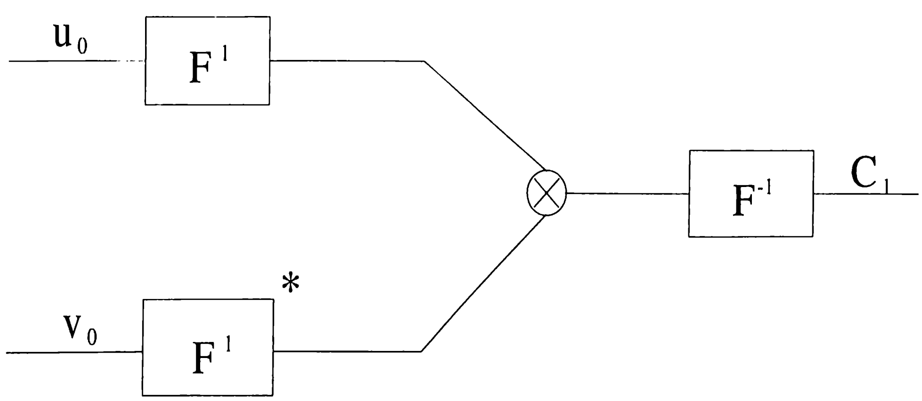

Figure 3.1: Conventional Correlation Operation.

If we give the definition of the conventional correlation in words:
Perform Fourier transforms of both objects, take the complex conjugate of one of the objects, multiply the results and finally perform an inverse Fourier transform.

It is known that the correlation operation is not invariant of unitary transforms. That is, if we transform the two functions u and v with a unitary transformation, the correlation of u and v after the transformation will be different than the correlation before the transformation. Therefore, as fractional Fourier transform is a unitary transform, we can claim that the correlation output will change when we correlate the two functions in different domains.

The definition of fractional correlation is similar to the definition of conventional correlation operation: Perform fractional Fourier transforms of both objects, find the representations of the objects in that fractional Fourier domain, then repeat the conventional correlation operation. That is, perform the Fourier transforms of both objects, take the complex conjugate of one of the objects, multiply the results and finally perform an inverse conventional Fourier transform. [2,14]

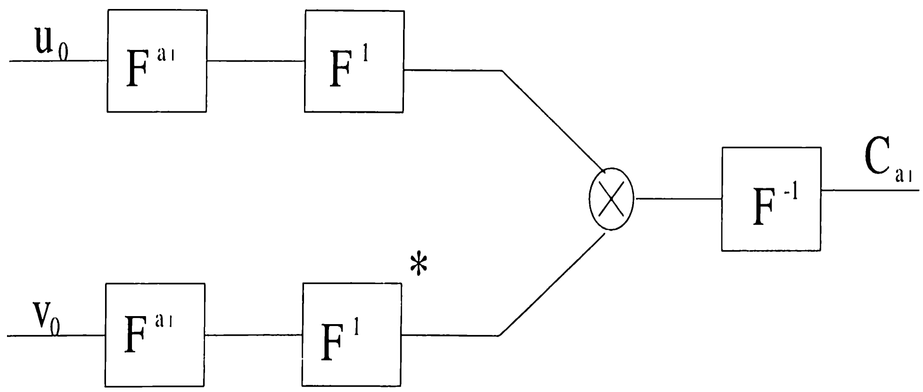

Figure 3.2: Fractional Correlation Operation.

This definition seems as if we are applying the conventional correlation operation to two signals. But, for the fractional correlation, the signals are not the signals themselves but their representations in fractional Fourier domains. That is, the definition of the correlation is extended from the time/space domain (which is also a fractional Fourier domain) to other fractional Fourier domains.

It should be noted that, instead of taking ath fractional Fourier transform and then taking the Fourier transform, the $(a+1)$th fractional Fourier transform could be taken directly. Because, the fractional Fourier transform has the index additivity property. But this procedure will not be applied in this thesis to keep the similarity with the conventional correlation definition.

The most common performance measures of correlation outputs are signal-to-noise ratio (SNR) and peak-to-correlation energy (PCE). The SNR measures the sensitivity of the correlation peak to the additive noise at the input which is given by:

$$SNR = \frac{| E[C_{u,v+n}(0)] |^2}{VAR[C_{u,v+n}(0)]} \tag{3.5}$$

where u and v are the input signals, n is the additive noise and $C_{u,v}$ is the correlation of the input signals u and v and VAR is the variance operation.

For the fractional correlation, it is easy to see that:

$$C^a_{u,v}(0) \quad = \quad C^1_{u,v}(0) \tag{3.6}$$

As the SNR depends on the correlation peak only (the value of $C^a_{u,v}(0)$), the SNR remains unchanged for all fractional Fourier domains. [2, 15].

Therefore, it can be said that the SNR performance of the fractional correlation is exactly the same as that of the conventional correlation. So, fractional correlation may be useful in applications where a shift-variant correlator is needed without any decrease in SNR performance.

The peak-to-correlation energy (PCE) is mathematically given by:

$$PCE \quad = \quad \frac{| \ C_{u,u}(0) \ |^2}{E_c} \tag{3.7}$$

where $E_c = \int_{-\infty}^{\infty} | \ C_{u,u}(x) \ |^2 \ dx$ is the correlation signal energy.

PCE is a measure of peak sharpness. As the fractional correlation is a shift-variant operation, the shape of the peak is irrelevant, so PCE is not a good measure for fractional correlation performance analysis [2, 15].

When fractional correlation is used in matched filtering instead of the conventional correlation, it was observed that the SNR performance was the same with the case where conventional correlation is used. But, the shapes of the correlation outputs were differing for different fractional Fourier orders. It should be noted that, the matched filter with fractional correlation is totally shift-variant for $a = 0$ and shift-invariant for $a = 1$. For the fractional Fourier orders increasing from 0 to 1, the shift-invariance property of the system decreases and it becomes to get more shift-variant.

It was observed from the research made on the fractional correlation that the fractional correlation provides the same performance in detecting the occurrence of a signal with white noise, but it is not a good tool for localization of the input object. In most image analysis applications, many other features should be extracted from the input pattern. Therefore, a more general technique, which also employs fractional Fourier transform, that can not only detect the presence of a certain input pattern, but also extract some features is proposed and presented in this thesis.

Chapter 4

Fractional Fourier Filtering Circuits

The digital implementation of a general linear system takes $O(N^2)$ time, where N is the space-bandwidth product of the image. This implementation time may be considered as slow. It is possible to obtain either exact realizations or useful approximations of linear systems or matrix-vector products, by synthesizing them in the form of repeated or multi-channel filtering operations in fractional Fourier domains. This technique provides much more efficient implementations with acceptable decreases in accuracy [24,25].

The single-stage fractional Fourier filtering configuration shown in Figure 4.1 relates the input and output vectors f and g respectively as:

$$g = [F^{-a}\Lambda F^{a}]f \tag{4.1}$$

where Λ denotes the diagonal matrix whose elements are equal to the samples of the filter function $h(u)$, and F^a represents the discrete ath order fractional Fourier transform matrix. [24]

In single-stage fractional Fourier filtering, the input function is transformed into the ath fractional Fourier domain, where it is multiplied with the filter $h(u)$. The result is transformed back into the original domain.

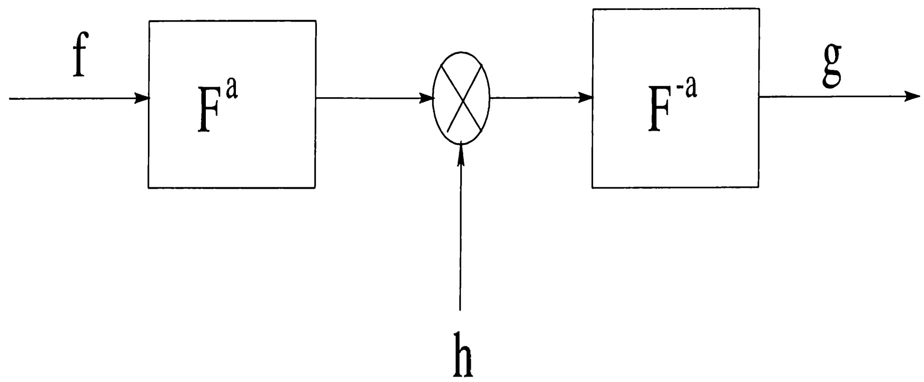

Figure 4.1: Single-stage fractional Fourier Filtering configuration.

The multi-channel fractional Fourier filtering structure shown in Figure 4.2 consists of M single-stage blocks in parallel. This configuration relates the input and output vectors f and g respectively as: [24]

$$g = [\sum_{k=1}^{M} F^{-a_k} \Lambda_k F^{a_k}] f \tag{4.2}$$

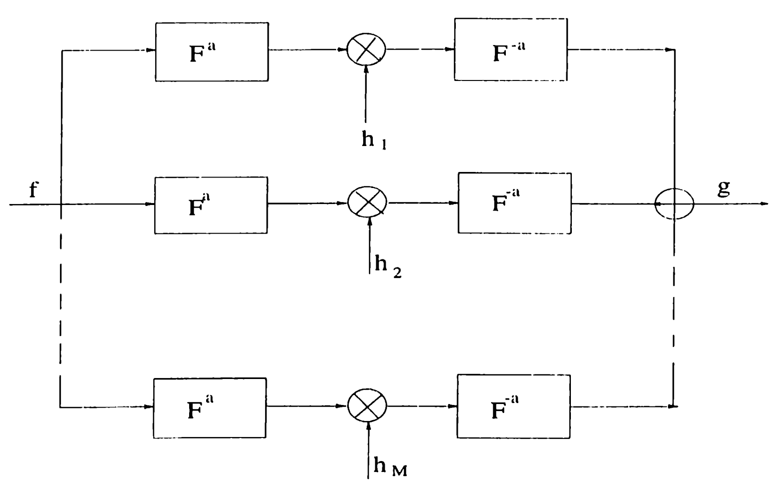

Figure 4.2: Multi-channel fractional Fourier filtering configuration.

For each channel k, the input is transformed to the a_kth domain, multiplied with a filter $h_k(u)$ and then transformed back to the original domain. (More generally, we may choose not to back transform, or we may transform to another domain. In this thesis, we will transform into the original domain.)

The repeated filtering structure is shown in Figure 4.3. In repeated filtering, the input is first transformed to the a_1th domain and multiplied by the filter $h_1(u)$ at that domain and then transformed back to the original domain. The result is transformed into the a_2th domain, multiplied with the filter $h_2(u)$ and transformed back. This operation is repeated M times. Therefore, it can be said that the repeated filtering employs M single-stage fractional Fourier filtering operations in series.

For repeated filtering, the relation between the input and output vectors f and g are given by [26]:

$$g = [F^{-a_M}\Lambda_M....F^{a_2-a_1}\Lambda_1 F^{a_1}]f \tag{4.3}$$

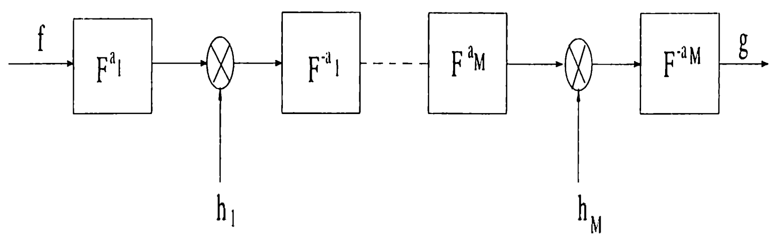

Figure 4.3: Repeated fractional Fourier filtering configuration.

The fractional Fourier transform can be implemented in $O(N \log N)$ time. Therefore, single-stage fractional Fourier filtering can also be implemented in $O(N \log N)$ time, while the digital implementation of repeated and multi-channel filtering configurations take $O(MN \log N)$ time [26]. The optical implementation of repeated and multi-channel configurations requires M-stage or M-channel optical system, each with space-bandwidth product N.

It can be said that the fractional Fourier filtering circuits interpolate between general linear systems and shift-invariant systems both in terms of cost and flexibility. For small M, the cost of the filtering circuits are low. The implementation cost increases as M increases. The flexibility of the system is also low for small M and it increases with increasing M. Therefore, there is a trade of between efficiency and flexibility. It should also be noticed that as M approaches N, the number of degrees of freedom for repeated filtering configuration approaches that of a general linear system.

Therefore, we have the possibility to approximate a given linear system with greater accuracy by increasing the number of filters M. For given value of M, the fractional Fourier filtering circuits can realize a certain subset of all linear systems exactly or to some other specified value of accuracy. It can be said that, the subset of linear systems increase with increasing M. Also, if we are given a linear system and want to approximize it with fractional Fourier filter circuits, we can decrease the error in approximation by increasing M. It should be noted here that, a shift-invariant system can be realized without any error with $M = 1$. Therefore, if the maximum cost of a specific problem is given to us, we can set M such that the cost of fractional Fourier filtering does not pass the given cost and find the filters that provide the best performance with that cost. In the other case, we may be given the desired accuracy in a specific problem. By increasing M we can obtain the fractional Fourier filter circuit that provides an implementation with the desired accuracy.

The fractional Fourier filtering can be applied in two different ways. Depending on the type of the problem, we may find the linear approximation of a system or directly given the kernel, H, which is multiplied by the input vectors to obtain the output vectors. We may synthesize this kernel by finding the fractional Fourier filter circuit that is closest to the given matrix H according to some specified criteria, such as minimum mean-square error. In the second approach, we can directly substitute the fractional Fourier filter circuits into the input-output relation and find the optimal fractional Fourier orders and the filters that correspond to them that optimize the given criteria, such as the minimum mean-square error.

In this thesis, single-stage or multi-channel fractional Fourier filtering configurations will be used either for kernel synthesis of the linear section of the image analysis problem or they will directly be substituted into the input-output relations and the filtering circuits that provides efficient implementations will be obtained.

It should also be noted that the repeated and multi-channel configurations may be combined to increase the performance in some applications.

Chapter 5

Feature Extraction with The Fractional Fourier Transform

This chapter of the thesis includes the theory and mathematical derivations of the proposed system for image analysis applications. Advantages and general framework of the system is discussed in order to clarify both the need for such a method and the framework used in the proposed techniques. Different information types that should be recovered in an image analysis application are classified and the representation types that can be employed to represent those informations at the output are provided. The nonlinear image analysis systems are approximated as a linear system followed by local nonlinearities at the output. Fast implementation techniques for the linear part with fractional Fourier filtering circuits and possible nonlinear operations at the output are discussed.

5.1 General Framework

In Chapter 2 and 3, most of the systems used for recognition of objects are analyzed. All of these systems have some advantages and disadvantages.

Matched filtering which is equal to correlating the input and output functions provides the optimum signal-to-noise ratio under white noise when there is no distortion in the input image. It also provides shift-invariance. When there are geometric distortions in the input image, the output degrades rapidly and therefore it becomes difficult to recognize the input object. Also as the filter is matched to one given image, it cannot be used for multi-class pattern recognition [3].

Distortion Invariant Filtering is proposed to provide some distortion invariances to the matched filtering, such as scale, projection and rotation invariances. These filters provide the desired invariances, but their signal-to-noise ratio performance is worse than matched filtering. Also the filters designed as distortion invariant add only one kind of invariance property to the shift invariant property of the matched filter [4,5].

Synthetic Discriminant Functions are introduced to obtain distortion invariant filters that have better discrimination availability. But Synthetic Discriminant Functions also have some problems. Firstly conventional Synthetic Discriminant Functions do not consider the occurrence of random noise in the input. So Minimum Variance Synthetic Discriminant Functions are introduced to overcome this problem. Secondly, as the conventional Synthetic Discriminant Functions control only one point (the origin) at the output, it is difficult to locate the input object. Minimum Average Correlation Energy Filters are designed to overcome this problem, but this time the noise performance of these kind of filters are not good enough. Therefore it can be said that Synthetic Discriminant Functions are useful to recognize an object with input distortions, but they are not very useful when the problem is to recognize and locate an object in noise [7-10,12].

Most of the above techniques provide shift-invariance, but in some cases shift invariance may disturb the detection process. Therefore Fractional Correlation is introduced which is a shift-variant operation. The signal-to-noise ratio performance of Fractional Correlation is equal to the conventional matched filtering operation and it can easily be implemented optically. It can be said that fractional correlation has similar advantages and disadvantages with matched filtering as its performance degrades rapidly with geometric image distortions and can be used to detect a single object [2, 14, 15].

It can be observed that all the above techniques are used to recognize the incoming patterns. These systems can also be used for feature extraction. The features of all possible input patterns may be stored and the features of the incoming pattern can be assigned as the features of the library input function to which the incoming pattern is matched. But this is an inefficient method. We propose that a system that gives the features of the input function as the output can be obtained and used.

Most image analysis problems are nonlinear problems which are difficult to implement. Especially in optics, it is difficult to implement nonlinear systems. We propose that, the nonlinear image analysis problem can be approximated by decomposing it into two parts, a global linear system and a local nonlinear system. Linear systems can easily be implemented either optically or digitally. Therefore the linear part of the system can be implemented easily but its cost may be high depending on the kind of implementation. The local nonlinear part can handle operations like comparison, decision, thresholding and morphological operations. The nonlinear part will not be complicated and will be handled by simple logic operations or thresholding and its implementation can be obtained by simple electronic circuitry.

We also claim that the output of this system may not be unique depending on the type of the application. More than one feature of the input scene may be extracted and can be displayed at the output by a display panel.

Therefore, the proposed system structure will be as it is shown in Figure 5.1.

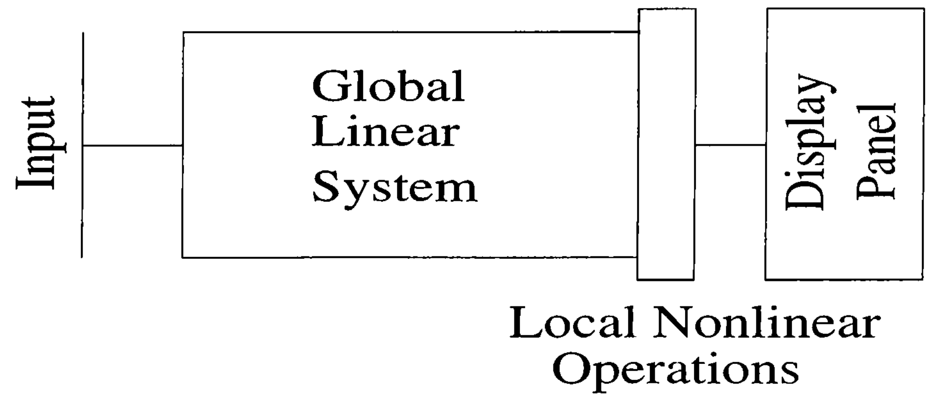

Figure 5.1: Proposed System Structure

We now discuss each part of the system in detail. The first and the main part is the global linear system. This system will be designed and implemented with two different procedures. The first procedure finds the linear kernel which provides all the input-output relations with minimum error with respect to a certain criteria, such as minimum least-squares error.

After obtaining the linear kernel of the linear system, it can be implemented either digitally or optically. Ordinarily, the cost of direct digital implementation is in $O(N^2)$, which can be considered as high. Therefore, we propose to implement the system by synthesizing its kernel with fractional Fourier filtering circuits. The digital implementation cost decreases to $O(N \log N)$ for single-stage and to $O(MN \log N)$ for repeated and multi-channel filtering configurations.It should be noticed that, the kernel we obtained is also an approximation and implementing it without any decrease in accuracy with high implementation costs is needless. Instead, we approximize this approximation again and provide lower cost for implementation with acceptable decreases in accuracy.

The second procedure to implement the global linear system places the fractional Fourier filter circuits directly into the input-output relation and finds the filters that provide the minimum error in the minimum mean-square error sense in that fractional Fourier domains. By this way, we do not need to approximize the linear kernel, which is also an approximation. Instead we directly find the fractional Fourier filters that satisfy the input-output relations with acceptable errors. Therefore, we expect this procedure to provide lower errors than the first procedure which employs kernel synthesis with fractional Fourier filters.

It should be noticed that, designing the system with this technique is more efficient then the previous technique. Because, we use least-squares algorithm only once for this technique, while we use it first to find the kernel and then to synthesize it in kernel synthesis with fractional Fourier filters for the first technique.

In the second part of the system, the local nonlinear operations are held. The nonlinear operations are restricted to be done locally to make its implementation easy. Depending on the type of the output representation, some nonlinear operations like comparison, decision, thresholding or morphological operations are done in this part. This part is implemented by some logic gates or simple electronic circuitry.

The last part of the proposed system is the display panel. As it will be described in the next section, different representations at the output may be employed. Also, the system will give all the desired features of the scene at the output. Therefore, displaying these multi-outputs which can be in different representations with a display panel, where every feature is displayed with a different display is beneficial.

It should be noted that, using different output representations for informations of the input brings a freedom to the problem. For different representations, the problem reduces to a different problem. Depending on the type of the application, some features may not be obtained accurately when they are represented by some representation types. Therefore, different representation types may be tested and applied to increase the performance of the system. By applying many different output representations, we can obtain the one providing the best performance for some features and a library showing which representation is best for which information can be obtained and used in similar applications.

The most common information types that are desired to be extracted from the input scene and possible output representations to represent those informations are discussed in the following section.

5.2 Information and Representation Types

In the previous sections, it is described that the proposed system is a general system which can be applied for most image analysis problems, but information and representation types which can be employed in image analysis problems will be examined in detail in this section.

5.2.1 Information Types

The basic information types which are desired to obtain from a given object or image can be classified as:

1. **Does a pattern exist?**

Depending on the type of the problem, it may be desirable to check if a certain input pattern or an objects exists in a given image. Maybe this is the most basic problem in image analysis but it has many applications.

2. **How many of the patterns are there?**

In some problems, it may be desirable to figure out how many of the patterns that are included in the library are present in the given input image. Also the number of presence of a particular pattern may be desired.

3. **Position, orientation, scaling of the given pattern**

Sometimes it may be desirable to obtain the position, orientation and scaling of a certain pattern in addition to its existence. Position of a pattern may be found by using the conventional correlation but orientation and scaling are not that much easy to find. The proposed method may be useful to figure out these properties of a given pattern.

4. **Area, perimeter, moment etc.**

In addition to position, orientation or scaling of a given pattern, some other geometric properties such as area, perimeter and moment of a given pattern may be needed. These properties of a certain pattern may be obtained by complex digital techniques, but if we can find a single filter that can recover these very fast, it would be very advantageous.

5. Intensity of a given pattern

In some optical problems, it is desired to determine the intensity of a pattern in addition to its other properties. Therefore finding the intensity of a certain pattern can be considered as one of the information types that is to be found by using recognition techniques.

6. Advanced Characteristics

Depending on the type of the problem, some other kinds of information may be expected to be recovered from a given image. Therefore the information types that are listed above can be extended depending on the type of the problem.

5.2.2 Representation or Coding Types

In the previous section, the information types that may be desired to extract from an input scene are described. These information types may be represented with many different representation types at the output. We can employ all output representation types for a specific information, but some representation types may be more advantageous than others for that information. Therefore, in order to increase the performance of the image analysis system, many different output representations may be tested and the ones that provide the best performance can be employed to represent different informations. In this section, the basic output representation types will be classified and examined.

1. Peak Amplitude Coding:

In this technique, the information is represented by the continuous amplitude of a certain point in the output plane. The amplitude of that point directly shows the value of desired information.

1. Peak Position Coding (Continuous):

In this technique, a peak is provided at the output and the position of that peak provides the information that is desired to be extracted. Each possible position location corresponds to a specific information level and the output of the system is forced to provide a peak whose position corresponds to the information level that is desired to be represented. A thresholding may be applied to find the peak at the output.

3. Point Binary Coding:
This representation technique may be used to check the existence of a certain pattern. The existence of the pattern may be coded to the amplitude of a single point and if that point amplitude is above the threshold, the system decides that the pattern exists and does not exist otherwise. This representation type can also be used as the discrete version of peak position coding when the position of every point corresponds to a certain information level. The position of the point whose amplitude is above the threshold gives the desired information.

4. Binary or Other Finite Codes:
Instead of representing the information by the amplitude or position of the peak at the output, we can code it to words like binary, ternary words or alphabets. A nonlinear operation (thresholding) is applied at the output. The system may give different results for different word representations and the option providing the best result may be chosen. We can use additional bits such as error correction bits in addition to the word representations. Also, we can use some redundant bits that provide more than one output possibilities for the same information at the input. The resulting output may be obtained by a decision rule consisting of logic operations, that decides on the output when one of these possible outputs is obtained.

5. Combinations of Different Representation Types
Combinations of the above representations types and other techniques may also be used. As an example, we can code an information to the amplitude of the peak while coding another information to its position. All other combinations are also allowed at the output and they can provide better results for different applications.

5.3 Design of the Kernel

For the proposed image analysis system, the most important part is the one where global linear operations are realized. We will design this global linear system with two different techniques. The first technique finds the kernel (the matrix) that satisfies the input-output relations with minimum error in the mean square sense. Then this kernel is synthesized and implemented by fractional Fourier filtering circuits. The other technique directly finds the fractional Fourier filters that satisfy the input-output relations with minimum error. In this section the design of the kernel is examined. The linear kernel, H, is obtained by using the least squares solution

Mathematically we want the system H to satisfy:

$$g_i \;=\; Hf_i \tag{5.1}$$

for all i's. Where f_i is the ith possible input and g_i is the desired output corresponding to the ith input. Note that f_i's are all the possible inputs that we want to extract features from. They may be patterns whose type, length, position, area, orientation and other properties vary. Depending on the application, one or more features from these patterns may be desired to be extracted. The output vectors, g_i's, are the outputs corresponding to those inputs that give the desired features if the input patterns in any of the representations described in the previous section. We want this equation to be satisfied for all possible inputs. Therefore we should write these linear equations for all possible inputs and try to find H which satisfies all these equations or provides the output values which are closest to the desired output values.

If we convert these equations into matrix form, they can be written as:

$$F.H \;=\; G \tag{5.2}$$

where F is the matrix containing the ith input f_i as its ith row for all i, H is the matrix containing the filter coefficients and G is the matrix containing the ith output g_i as its ith row. So the problem reduces to finding the elements of H from these linear set of equations.

For most problem types, the number of possible inputs are much higher than the number of filter coefficients. Therefore we have a linear equation set consisting of more equations than the number of unknowns and, in general, no solution exists. In this case, the equations are inconsistent and the solution is said to be over determined. The approach that is commonly used in this situation is to find the "least-squares solution", i.e the H that minimizes the norm of the error in mean-square error sense. In this thesis least squares solutions are used as the filter matrices when the equations are inconsistent [27].

The norm of the error is given by:

$$||e||^2 = ||G - FH||^2 \tag{5.3}$$

the least squares solution has the property that the error,

$$e = G - FH \tag{5.4}$$

is orthogonal to the column vectors of F. This orthogonality implies that :

$$F^H e = 0 \tag{5.5}$$

or,

$$F^H F H = F^H G \tag{5.6}$$

which are known as the normal equations. If the columns of F are linearly independent (F has full rank), then the matrix $F^H F$ is invertible and the least squares solution is:

$$H_0 = (F^H F)^{-1} F^H G \tag{5.7}$$

or,

$$H_0 = F^+ G \tag{5.8}$$

where the matrix,

$$F^+ = (F^H F)^{-1} F^H \tag{5.9}$$

is the pseudo-inverse of the matrix F for the over determined problem. Furthermore, the best approximation $\hat{G}$ is given by:

$$\hat{G} = FH_0 = F(F^H F)^{-1} F^H G \tag{5.10}$$

it follows that minimum least squares error is :

$$min||e||^2 = FH_0 = G^H G - G^H F H_0 \tag{5.11}$$

The least squares solution shown above gives the filter matrix for over determined case.

In the above equations we considered that FF^H is invertible, which meant that the columns of F were linearly independent. If this condition is not satisfied, the solution H_0 will not be unique. In those cases the solution which has the minimum length should be chosen as the solution.

In order to find this solution, suppose that the singular value decomposition of F is $F = Q_1 \Sigma Q_2^T$, where the singular values $\sigma_1, ..., \sigma_r$ are on the diagonal of Σ. Then the pseudo-inverse of F is:

$$F^+ = Q_2 \Sigma^+ Q_1^T \tag{5.12}$$

where the reciprocals of singular values, that are $1/\sigma_1, ..., 1/\sigma_r$ are on the diagonal of Σ^+. Then the minimum length least squares solution to $FH = G$ is:

$$H^+ = F^+ G = Q_2 \Sigma^+ Q_1^T G \tag{5.13}$$

To prove this, write the error as:

$$||FH - G|| = ||Q_1 \Sigma Q_2^T H - G|| \tag{5.14}$$

Multiplication by the orthogonal matrix Q_1^T leaves lengths unchanged, so :

$$||FH - G|| = ||\Sigma Q_2^T H - Q_1^T G|| \tag{5.15}$$

Introduce a new unknown $y = Q_2^T H = Q_2^{-1} H$ which has the same length as H. Then minimizing $||FH - G||$ is the same as minimizing $||\Sigma y - Q_1^T G||$. This has a diagonal matrix and it is easy to see that the y minimizing the error is $\Sigma^+ Q_1^T G$. Therefore the best H is:

$$H = Q_2 y = Q_2 \Sigma^+ Q_1^T G \tag{5.16}$$

Depending on the type of the problem, the number of equations may be less than unknowns. Therefore, provided that the equations are not inconsistent, there are many solutions, i.e., the solution is under determined or incompletely specified. One approach that is often used to define a unique solution is to find the solution that has the minimum norm. If the rows of F are linearly independent then FF^H is invertible and the minimum norm solution is :

$$H_0 = F^H (FF^H)^{-1} G \tag{5.17}$$

As the last case, the input functions matrix F may be a square matrix with full rank and at that case the filter matrix can be calculated by using Gaussian elimination as:

$$H = F^{-1}G \tag{5.18}$$

While designing the linear system, it may difficult to create the input and output matrices that contains all possible input and output values. But this operation is done only once and the designed filter will be used to obtain outputs with small errors for all possible inputs. Also by tolerating higher errors we may design the system by not using all possible inputs but by selecting a group of inputs that can represent an important amount of input variations. In this case the error made in the output will increase, especially for the possible inputs that are not used to design the system, but it will be easier to form the input and output matrices and so to design the linear system. The obtained system can be implemented by matrix vector product in $O(N^2)$ time if the input and output functions have the same length N and will be implemented in $O(NM)$ time if input/output function has length N and output/input function has length M.

It is costly to implement this linear system with a matrix-vector multiplication. Therefore we will implement this system with fractional Fourier filter circuits whose implementation costs are lower. Kernel synthesis with fractional Fourier circuits is also an approximation. Therefore, instead of implementing the kernel directly, which is also an approximation, we can implement it with fractional Fourier filtering circuits with an acceptable decrease in accuracy. The kernel synthesis with fractional Fourier filtering circuits is described in detail in the following section.

5.4 Kernel Synthesis

In the previous section, we examined the design of the global linear system part, that is to be employed in our whole image analysis system, with least squares solution. As it is explained before, we will implement this system by synthesizing its kernel with fractional Fourier filter circuits to decrease the implementation cost.

In this thesis, the kernel will be synthesized by using a single-stage or multi-channel(parallel) fractional Fourier filter circuits. As it is explained before, the single-stage fractional Fourier filter relates the input and output vectors as:

$$g = [F^{-a} \Lambda F^{a}] f \tag{5.19}$$

where Λ denotes the diagonal matrix whose elements are equal to the samples of the filter function $h(u)$, and F^a represents the discrete ath order fractional Fourier transform matrix.

In the previous section, we presented the design of the linear kernel which obtains the outputs for all inputs with an acceptable error. H is the least squares solution of all the input-output equation set. The kernel relates the input and output vectors as:

$$g = H.f \tag{5.20}$$

Therefore, it is easy to observe that for kernel synthesis with single-stage fractional Fourier filter, the kernel is approximated as:

$$H = F^{-a} \Lambda F^{a} \tag{5.21}$$

Once we fix the a, this equation reduces to a linear set of equations with the elements of Λ as the unknowns. The elements of Λ may be obtained by the least squares solution described in the previous section. This can more easily be seen if we write the same equation as:

$$H = \sum_{k=1}^{N} \lambda_k f_k f_k^H \tag{5.22}$$

where λ_k is the kth element on the diagonal of the diagonal matrix Λ and f_k is the kth column vector of the $-a$th fractional Fourier transform matrix F^{-a}. The only unknowns in these equations are λ_k's and they can be obtained by least squares solution. λ_k's are the samples of the filter $h(u)$. After λ_k's are obtained, we have the filter $h(u)$ in ath fractional Fourier domain. The implementation is easy. The input function is transformed to the ath fractional Fourier domain by multiplying it with F^a. Then it is multiplied with the filter function $h(u)$ in that domain and transformed back to the original domain by multiplying the result with F^{-a}.

It should be noted that, for different a's, the obtained filter will be different and the overall performance of the system also differs. Therefore, an optimization over a's can be applied and the one providing the best performance can be used.

The implementation time of single-stage fractional Fourier filtering is in $O(N \log N)$ where implementing the kernel directly by matrix-vector multiplication takes $O(N^2)$ time. Therefore, if we can provide acceptable decreases in accuracy with single-stage fractional Fourier filtering, it is advantageous to use it as its implementation cost is low.

The multi-stage fractional Fourier filtering configuration can be applied in a similar manner. This time the kernel is approximated as:

$$H = \sum_{k=1}^{M} F^{-a_k} \Lambda_k F^{a_k} \tag{5.23}$$

where Λ_k denotes the diagonal matrix whose elements are equal to the samples of the filter function $h_k(u)$and F^a represents the discrete ath order fractional Fourier transform matrix. The same equation can be written as :

$$H = \sum_{i=1}^{M} \sum_{k=1}^{N} \lambda_{ik} f_{ik} f_{ik}^H \tag{5.24}$$

where λ_{ik} is the ith element of the diagonal matrix Λ_i and f_{ik} is the ith column of the $-a_k$th order fractional Fourier transform matrix. For given a_k's the system reduces to a linear set of equations with λ_{ik}'s as the unknowns and the solutions can be obtained by using least squares method. After finding the λ_{ik}'s, we have the filters $_k(u)$'s in a_kth fractional Fourier domains. The input function is transformed to $_k$th fractional Fourier domain at each branch, multiplied with the filter $h_k(u)$ and then transformed to the original domain. The addition of the obtained values for all branches gives the output value.

It should be noted that the performance of the system differs for different a_k's. Therefore, an optimization over the a_k's may be applied to increase the performance.

The implementation of multi-channel fractional Fourier filtering takes $O(MN \log N)$ time which can be considered as fast when compared with $O(N^2)$ for large N. Because M is generally much lower than N.

We can represent the approximation of H by fractional Fourier filtering by T, which means that the relation between input and output vectors f and g is now given by:

$$g = Tf \tag{5.25}$$

It should be noted here that, the above equations are all given for systems where input and outputs have the same length (i.e., the Kernel is a square matrix). Extension to rectangular case is straight forward. The kernel for the rectangular case for single-stage fractional Fourier filtering is given by:

$$T_{N_{out} \times N_{in}} = F^{-a}_{N_{out}} \Lambda_{N_{out} \times N_{in}} F^{a}_{N_{in}} \tag{5.26}$$

and the kernel for the multi-stage fractional Fourier filtering is given by:

$$T_{N_{out} \times N_{in}} = \sum_{k=1}^{M} F^{-ak}_{N_{out}} (\Lambda_k)_{N_{out} \times N_{in}} F^{ak}_{N_{in}} \tag{5.27}$$

5.5 Design and Implementation in fractional Fourier domains

In the previous sections, the design of kernel of the global linear system and its implementation by using kernel synthesis with fractional Fourier filtering circuits is explained. In the design procedure, the input-output relations are written in matrix form given by:

$$F.H = G \tag{5.28}$$

where F is the matrix containing the ith input f_i as its ith row for all i, G is the matrix containing the ith output g_i as its ith row. Then the system kernel H which provides the minimum error in the mean square error sense to the input-output relations is obtained. This was done by using the least squares algorithm for the solution of linear equation set.

After the kernel is designed, it was proposed to be implemented by fractional Fourier filter circuits. For single-stage fractional Fourier filtering the kernel is given by [24]:

$$H = F^{-a} \Lambda F^{a} \tag{5.29}$$

where Λ denotes the diagonal matrix whose elements are equal to the samples of the filter function and F^a represents the discrete ath order fractional Fourier transform matrix.

For fixed a, this equation set reduces to a set of linear equations with the elements of Λ as the unknowns. This linear equation set can be solved by least squares and the filter which provides the minimum error can be obtained.

For the multi-channel filtering case, the kernel is given by [24]:

$$H = \sum_{k=1}^{M} F^{-a_k} \Lambda_k F^{a_k} \tag{5.30}$$

and for known a_k's, this equation system reduces to a similar set of linear equations and con be solved by using least squares.

In this section, the theory for designing the system directly in fractional Fourier domains will be given. The system will be either a single filter in a fractional Fourier domain or a multi-channel fractional Fourier filtering configuration in different fractional Fourier domains. That is, instead of finding the kernel H and then implementing it with fractional Fourier filters by using kernel synthesis, the fractional Fourier filters can be placed in the input output relations and the filter coefficients that provide a solution to this system with minimum error can be calculated.

For single-stage fractional Fourier filtering, the relation between input and output vectors can be written as:

$$g = F^{-a} \Lambda F^{a} . f \tag{5.31}$$

If we assume that the $-a$th order fractional Fourier transform matrix F^{-a} has $f_1, ..., f_N$ as its column vectors, such that:

$$F^{-a} = \begin{bmatrix} f_1 & | & f_2 & | & & | & f_N \end{bmatrix} \tag{5.32}$$

we can write the relation between input and output vectors as:

$$g = \sum_{k=1}^{N} \lambda_k f_k f_k^H . f \tag{5.33}$$

where λ_k is the kth element on the diagonal of the diagonal matrix .

It can easily be seen that $f_k^H f$ product gives a constant value, which is denoted by c_k. Then the input-output relation can be written as:

$$g = \sum_{k=1}^{N} \lambda_k f_k c_k \tag{5.34}$$

For fixed a, the only unknowns in these equations are λ_k's and they can be obtained from a matrix vector product given by:

$$\begin{bmatrix} f_1 c_1 & | & f_2 c_2 & | & .... & | & f_N c_N \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \\ \\ \lambda_N \end{bmatrix} = \begin{bmatrix} \\ g \\ \\ \end{bmatrix} \tag{5.35}$$

If these equations are written for all input-output pairs in the training set, the λ_k's that provide the closest outputs to the expected ones can be obtained in that fractional Fourier domain. These equation sets can also be solved by least squares. Therefore the designed filter provides the output matrix that is closest to the output matrix in the training set in the mean square error sense.

The extension to multi-channel fractional Fourier filtering is straight forward. This time, the relation between input and output vectors can be given by:

$$g = \sum_{i=1}^{M} F^{-a_i} \Lambda_i F^{a_i} . f \tag{5.36}$$

which can also be written as:

$$g = \sum_{i=1}^{M} \sum_{k=1}^{N} \lambda_{ik} f_{ik} f_{ik}^{H}.f \tag{5.37}$$

where λ_{ik} is the kth element of the diagonal matrix Λ_i and f_{ik} is the kth column of the $-a_i$th order fractional Fourier transform matrix. Similar to single-stage filtering case, $f_{ik}^{H}.f$ multiplication is a constant denoted by c_{ik}.

These equations can also be reduced to a single matrix-vector product and the λ_{ik}'s can be obtained from this product by the least squares solution method.

For the multi-channel filtering configuration, the equations for a single input-output pair are written in matrix vector product form as:

$$\left[\begin{array}{ccccccc} f_{11}c_{11} & | & & | \; f_{N1}c_{N1} & | \; f_{12}c_{12} & & | \; f_{NM}c_{NM} \end{array}\right] \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \\ \\ \\ \lambda_{MN} \end{bmatrix} = \begin{bmatrix} \\ g \\ \\ \end{bmatrix} \tag{5.38}$$

The above equations can be written for all input-output pairs and the λ_{ik}'s can be obtained by using least squares solution. The obtained λ_{ik}'s provide a solution closest to the outputs in the mean square sense in that fractional Fourier domains.

It can be said that designing the system directly in fractional Fourier domains is more efficient than first finding the kernel, which gives the optimum linear approximation of the nonlinear system, and than using kernel synthesis with fractional Fourier filter circuits for implementation.

It should be noted that the above equations consider that the input and output vectors have the same length. Therefore, the equations should be modified for systems with unequal input and output lengths. The extension to rectangular case can simply be held by taking the filter matrix rectangular as in Eq. 5.26 and Eq. 5.27.

5.6 Analogies with Neural Networks and Optical Interconnections

If a careful observation is made, it can be said that there is an analogy between neural networks and our proposed techniques. In the proposed techniques we design the system such that desired outputs should be obtained for each input, which is done in single-layer neural networks similarly. It can be said that the outputs of a single layer neural network are linear combinations of the inputs plus a thresholding at the output stage. In some neural network applications the weights of the inputs that constitute the outputs may be chosen such that the output is the Fourier transform of the input function. Similarly, the weights can be chosen such that the outputs are the fractional Fourier transforms of the inputs. As the proposed systems are implemented in fractional Fourier domains, it can be said that the outputs are linear combinations of the inputs where the weights are chosen such that the outputs are combinations of the fractional Fourier transforms of the inputs [28]. Therefore it can be said that the proposed systems are analog to a single layer neural network.

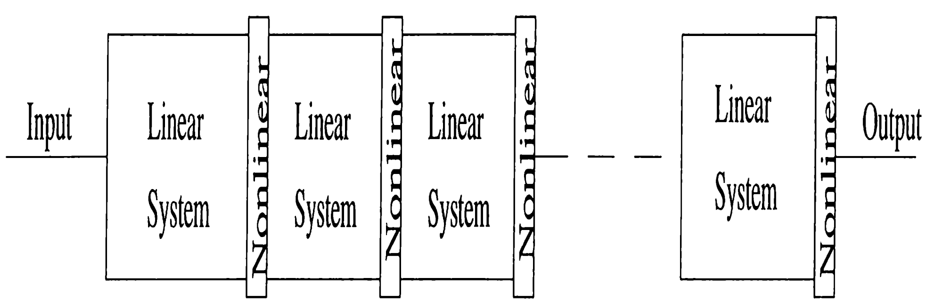

Figure 5.2: Extension to multi-layers.

Similarly, by using the proposed systems repeatedly, such that the output of a system is the input of the one following it, most of the problems that can be solved by multi-stage neural networks can be solved. Therefore, the proposed system may be applied in most neural network applications.

It should also be noted that, in addition to its analogy to neural networks, the system also contains similarities to the proposed optoelectronic computer architectures. Therefore it can also be applied in optical interconnections and in optical computing [29,30].

Chapter 6

Simulations

This chapter of the thesis consists of simulations that illustrate the applications and performance of the proposed methods for feature extraction. The simulations are classified in two groups. In the first part of simulations, the advantages and properties of different kinds of output representations are examined. As the aim in these simulations is to provide some comparisons and reach some conclusions about different output representations, we do not deal with the implementation of the system in these simulations. The efficiency considerations are examined in the second part of simulations.

The second part of simulations provides results and comparisons of two different techniques for the implementation of the linear part of the whole feature extraction problem. Kernel synthesis with fractional Fourier filtering circuits and direct implementation in fractional Fourier domains with the fractional Fourier filtering configurations are applied on some feature extraction problems and the results are compared and discussed in order to reach some conclusions.

6.1 Simulations: Part I

These simulations are done to provide a comparison between different output representation types and to reach some conclusions. Simulations start with a simple 1-D application to show the advantages and properties of different output representation types. Then, the example is extended to a more realistic input type. The following simulations cover most of the possible features that can be extracted from a 1-D signal. Simulations on some 2-D examples are provided as the last simulations of this part.

It should be noted that two error criterias will be used in most simulations. The first one is the average error. It gives the absolute value of average of the derivations of the obtained values from the expected ones. Its is provided in the same units with the information types. That is, the average error is the error in pixels. The second error criteria is used for binary or other word representations. Because, for these representation types the effect of a possible error in any of the bits is different than the other representations. Therefore, we will obtain the error as the ratio of wrong outputs for all possible inputs in simulations that employ these representation types.

Example 1:

As the first example, we consider 1-D rectangular binary pulses whose position and length varies. The length of the pulse is 16 points maximum and the position is considered as the position of the left corner of the pulse. Different output representation types will be applied to represent the length and position information in each simulation.

Case 1:

Feature to be Extracted	*Output Representation Type*	*Average Error*
Length of the pulse	Peak Amplitude Coding	0
Position of the pulse	Peak Amplitude Coding	2.35

Some possible input-output pairs are shown in Figure 6.1. It was observed that

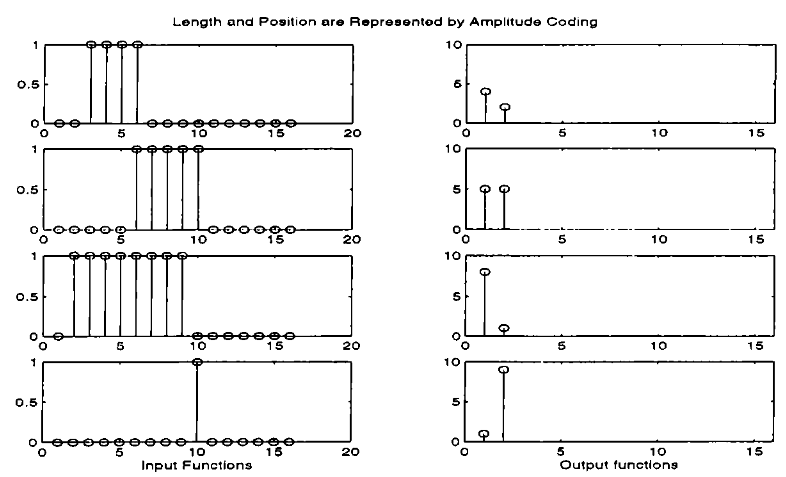

Figure 6.1: Examples of input and output pairs. The length of the pulse is coded to the amplitude of the first peak at the output and the position of the pulse is coded to the amplitude of the second peak.

the system is capable of finding the length of the pulse without any error as it is a linear operation. The error in obtaining the position of the pulse when it is represented by amplitude coding is high. Therefore some other representations should be applied to represent the position information.

Case 2:

Feature to be Extracted	*Output Representation Type*	*Average Error*	*Error in Number of Correct Outputs*
Length of the pulse	Peak Amplitude Coding	0	0%
Position of the pulse	Binary Word	Not Applicable	36.7%

Some possible input-output pairs are shown in Figure 6.2. A nonlinear oper-

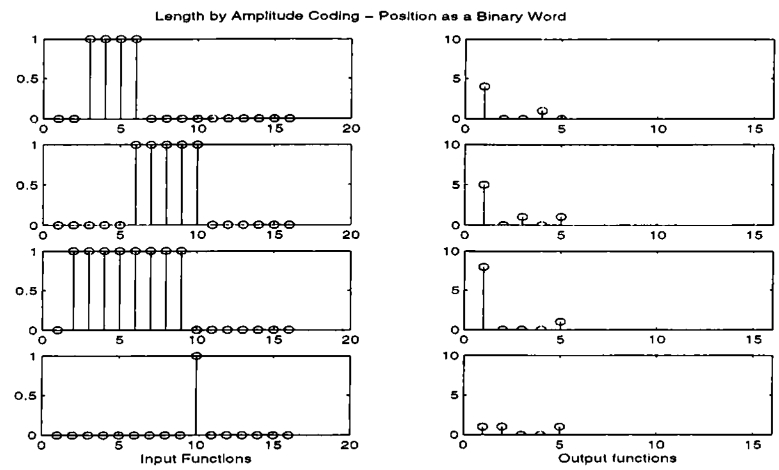

Figure 6.2: Examples of input and output pairs. The amplitude of the first bit of the output shows the length of the pulse. Remaining 4 bits of the output shows the position of the left corner of the pulse in binary words.

ation, thresholding is applied at the output. It is observed that representing the position information with a binary word instead of the amplitude of a single peak increases the performance of the system. The system is capable of extracting both the length and position of 63.3% of inputs without any error, but the error can still be considered as high.

Case 3:

Feature to be Extracted	*Output Representation Type*	*Average Error*	*Error in Number of Correct Outputs*
Length of the pulse	Peak Amplitude Coding	0	0%
Position of the pulse	Ternary Word	Not Applicable	45%

Some possible input-output pairs are shown in Figure 6.3. We can represent

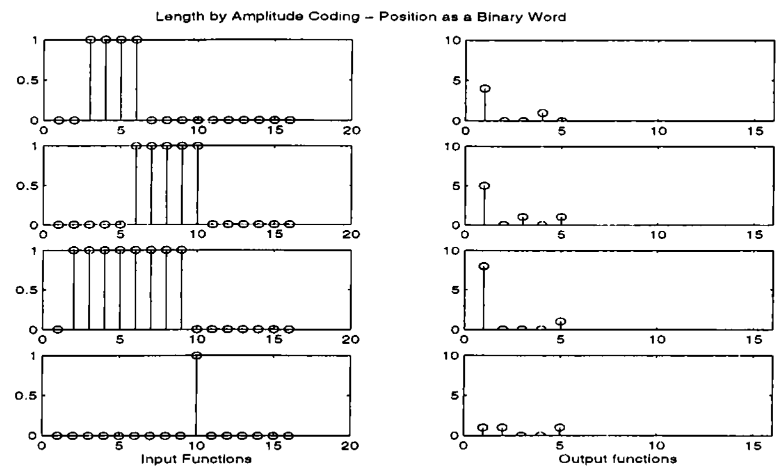

Figure 6.3: Examples of input and output pairs. The amplitude of the first bit of the output shows the length of the pulse. Remaining 3 bits of the output shows the position of the left corner of the pulse in ternary words.

any information with different word representations and alphabets. An optimization can be done on these different alphabets and the one providing the best result can be used. We used ternary words to represent the position information in this simulation. It was observed that its performance is worse than the system employing binary words. Therefore, representing the position information with a binary word is better for this application.

Case 4:

Feature to be Extracted	*Output Representation Type*	*Average in Error*	*Error in Number of Correct Outputs*
Length of the pulse	Peak Amplitude Coding	0	0%
Position of the pulse	Binary Word + Error Bit	Not Applicable	36%

Some possible input-output pairs are shown in Figure 6.4. It was observed

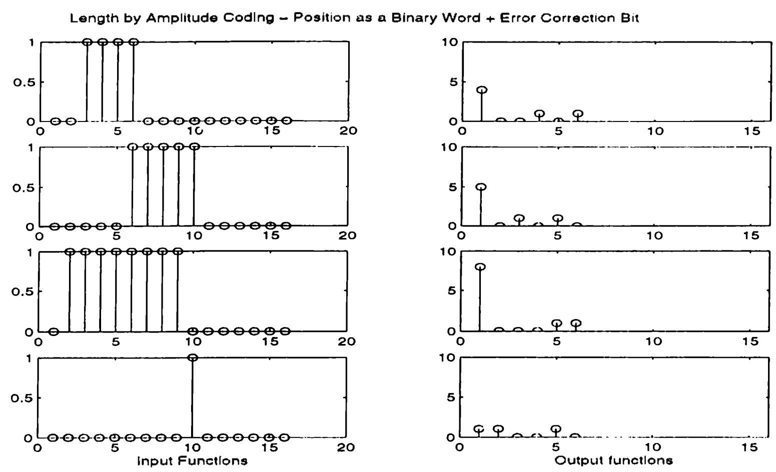

Figure 6.4: Examples of input and output pairs. The amplitude of the first bit of the output shows the length of the pulse. The following 4 bits are the binary word representation of the position of the pulse and the last bit is the error correction bit.

that, adding an error correction bit does not increase the systems performance significantly. Therefore, as it will increase the cost of the nonlinear part of our proposed system, there is no need to use an error correction bit for this application.

Case 5:

Feature to be Extracted	*Output Representation Type*	*Average Error*
Length of the pulse	Peak Amplitude Coding	0
Position of the pulse	Peak Position Coding	0

Some possible input-output pairs are shown in Figure 6.5. It was observed that,

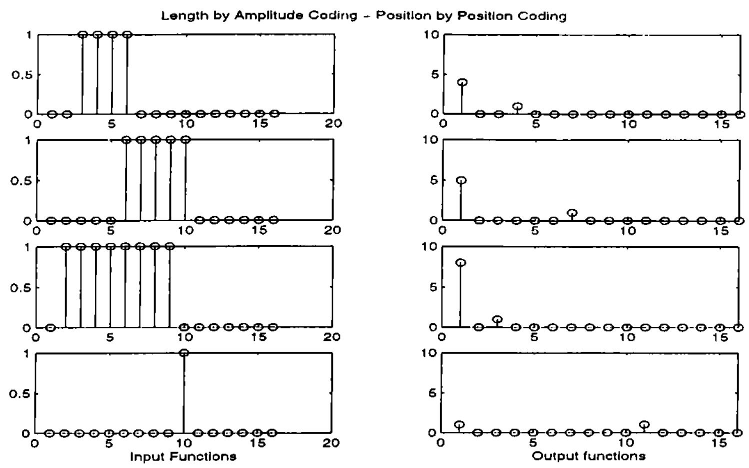

Figure 6.5: Examples of input and output pairs. The amplitude of the first bit of the output shows the length of the pulse. The following 16 bits are used to represent the position of the left corner of the pulse.

with a proper thresholding at the output, the system was able to extract the features of all possible inputs without any error. Therefore, it can be concluded that, representing the length information by peak amplitude coding and the position information with peak position coding provides the optimum result for this application.

Case 6:

Feature to be Extracted	*Output Representation Type*	*Average Error*
Length of the pulse	Amplitude and Position	10
Position of the pulse	of the Same Peak	1.7

Some possible input-output pairs are shown in Figure 6.6. The system pro-

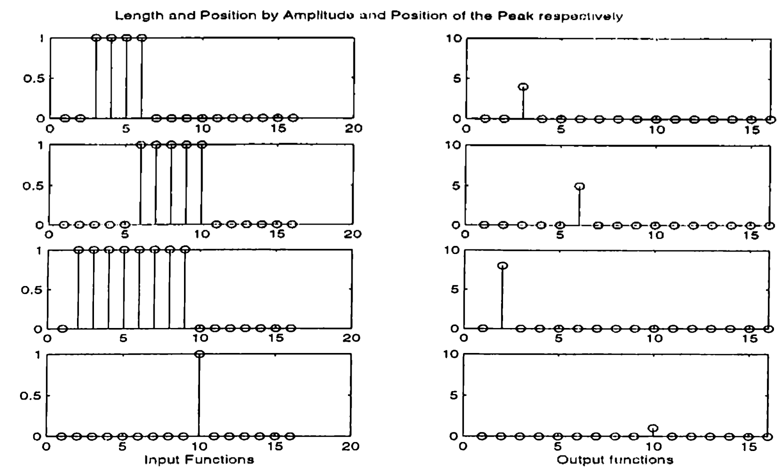

Figure 6.6: Examples of input and output pairs. The amplitude of the peak at the output shows the length of the input pulse and the position of the peak shows the position of the left corner of the pulse.

vides a poor performance for these kind of representations for this application. Therefore, it can be said that, representing the length and position information with the amplitude and position of the same peak is not a good solution for this application. It is interesting to note that when these representations are used separately, they provide the optimum error, but they provide very high errors when they are combined. The system was able to obtain the length of the pulse by making an integration operation for peak amplitude coding, but when the representations are combined the lengths can not be obtained by an integration operation. The performance of the system also decreases for calculating the position of the pulse when the two representations are combined.

Example 2:

In this example, the lengths of the inputs in example 1 are extended to 128 in order to achieve a more realistic case. The training set does not consider all possible input-output pairs to increase the design efficiency. The aim in this example is to examine the interpolation capability of the system for the input signals that are not present in the training set.

Case 1:

Feature to be Extracted	*Output Representation Type*	*Average Error (Training Set)*	*Average Error (Non-Training)*
Length of the pulse	Peak Amplitude Coding	0	0
Position of the pulse	Peak Position Coding	0	32

Some possible input-output pairs are shown in Figure 6.7. It was observed that

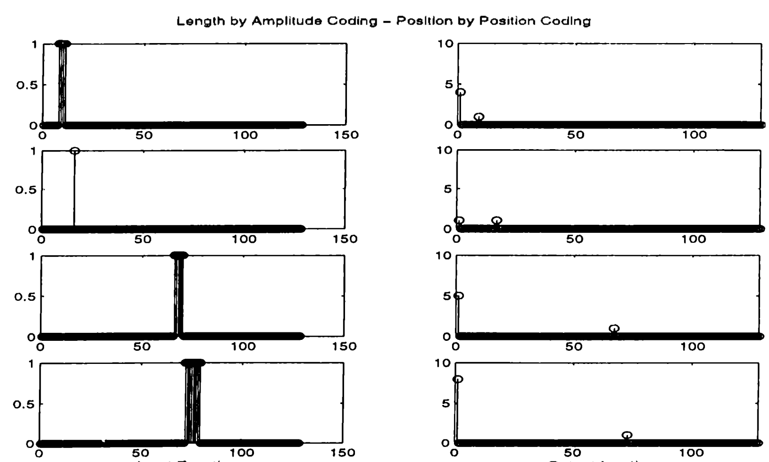

Figure 6.7: Examples of input and output pairs. The amplitude of the first bit of the output represents the length of the pulse and the remaining 128 bits represent the possible position locations.

the system was able to extract the features of the inputs that are present in the training set without any error. The system could also extract the length information of the inputs that do not exist in the training set but could not extract the positions of them. There were only 16 possible position locations for the inputs in the training set. Therefore, it can be said that, increasing the number of possible position locations may increase the interpolation capability of the system.

Case 2:

Feature to be Extracted	*Output Representation Type*	*Average Error (Training Set)*	*Average Error (Non-Training)*
Length of the pulse	Peak Amplitude Coding	0	0
Position of the pulse	Peak Position Coding	0	8

The same example is simulated with higher number of possible locations for the inputs in the training set. 64 possible position locations are used in the training set and the size of the training set is kept constant by decreasing the possible lengths of the inputs in the training set. It was observed that the system was again able to interpolate the lengths without any error and the error made in interpolating the positions is decreased. Therefore, in a given application, the training set should be chosen such that it includes minimum information about the features the system can interpolate and contains more information about the features the system can not interpolate without any error.

Example 3:

In a 1-D application, the basic features that can be recovered from a signal are length, position, intensity, number of occurrence of a single pattern and occurrence of a certain pattern among many different patterns. Extraction of length and position features are examined in the previous examples. In this example, the intensity of the incoming pulse is going to be extracted in addition to its position and length.

Feature to be Extracted	*Output Representation Type*	*Average Error*
Length of the pulse	Peak Amplitude Coding	0
Intensity of the Pulse	Peak Amplitude Coding	0
Position of the pulse	Peak Position Coding	0.4

Some possible input-output pairs are shown in Figure 6.8. It was observed that

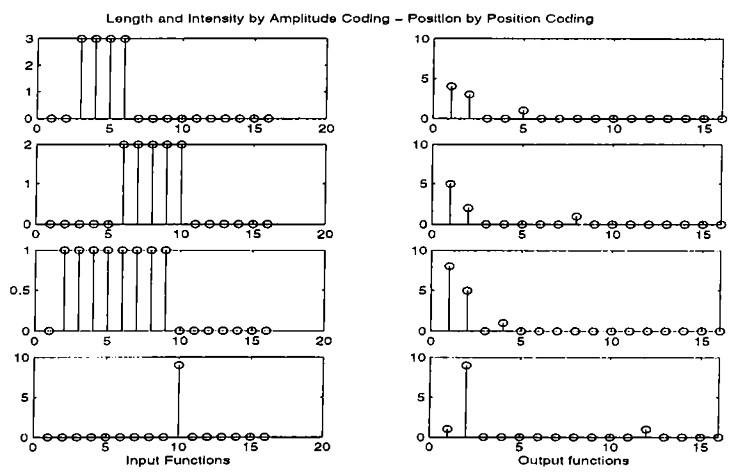

Figure 6.8: Examples of input and output pairs. The amplitude of the first and second bits of the output represent the length and intensity of the pulse and the remaining 16 bits represent the possible position locations.

the system could obtain these features with small errors.

Example 4:

In this example, the number of occurrence of a certain pattern is going to be extracted from the input signal.

Feature to be Extracted	*Output Representation Type*	*Average Error*
Number of Occurrence	Peak Amplitude Coding	0

Some possible input-output pairs are shown in Figure 6.9. It was observed that the system was able to recover this feature without any error as this is a liner operation.

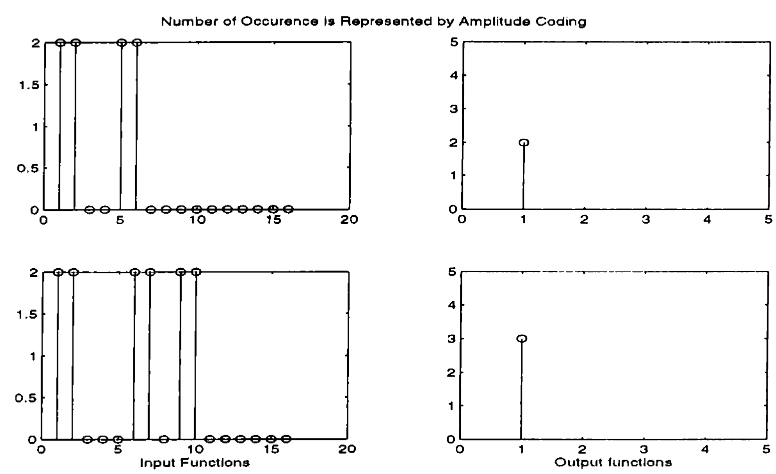

Figure 6.9: Examples of input and output pairs. The amplitude of the peak represents the number of occurrence of the pulse .

Example 5: In this example, three different patterns whose position and length varies are applied as the inputs. Some possible input-output pairs are shown in Figure 6.10.

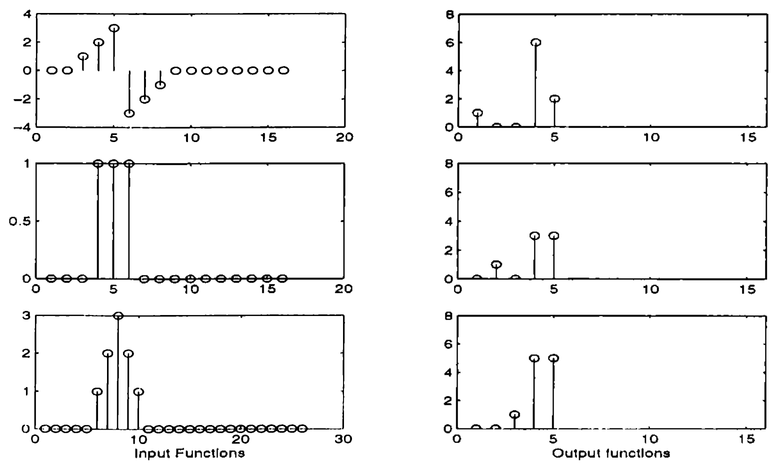

Figure 6.10: Examples of input and output pairs. The first three bits are used to determine the kind of the input pattern. The amplitudes of the following two bits represent the length and the position pattern respectively.

Case 1:

Feature to be Extracted	*Output Representation Type*	*Average Error*	*Error in Number of Correct Outputs*
Pattern Type	Point Binary Coding	Not Applicable	45%
Length of the Pattern	Peak Amplitude Coding	3.4	Not Applicable
Position of the Pattern	Peak Amplitude Coding	6	Not Applicable

It was observed that the system was not able to extract these features. Because, the error obtained for this simulation is very high.

Case 2:

Feature to be Extracted	*Output Representation Type*	*Average Error*	*Error in Number of Correct Outputs*
Pattern Type	Point Binary Coding	Not Applicable	40%
Length of the Pattern	Peak Amplitude Coding	3.6	Not Applicable
Position of the Pattern	Peak Position Coding	2	Not Applicable

In both of the simulations, it was observed that the system was not capable of extracting these three features at the same time. Some other representation types are also tested but the error obtained was similar for all cases.

Example 6: In all the previous examples, the input signals were in 1-D. In this example, a simple 2-D application is analyzed. The possible input functions are squares whose position and area changes. The position of the square is considered as the position of the upper left corner of the square. Some possible input-output pairs are shown in Figure 6.11.

Feature to be Extracted	*Output Representation Type*	*Average Error*
Area	Peak Amplitude Coding	0
Position of the Square	Peak Position Coding	0

With a proper thresholding at the output, the system is able to extract these features nearly without any error. Other representation types are also tested for this 2-D example and the results obtained were similar with the ones obtained in example 1 for different representation types.

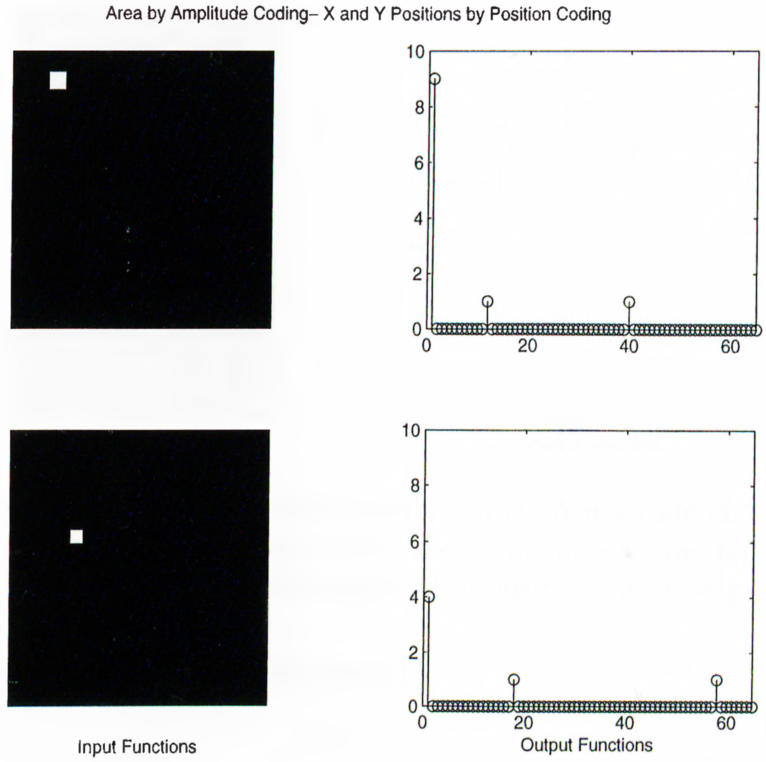

Figure 6.11: Examples of input and output pairs. The amplitude of the first bit of the output represents the area of the square.The following 32 bits represent the position of the left upper corner of the square in x direction and the following 32 bits represent the position in y direction.

Example 7: In this 2-D example, the input functions are rectangles whose positions, dimensions and intensities vary. Some possible input-output functions are shown in Figure 6.12.

Feature to be Extracted	*Output Representation Type*	*Average Error*
Length in x-direction	Peak Amplitude Coding	0
Length in y-direction	Peak Amplitude Coding	0
Intensity	Peak Amplitude Coding	0

It was observed that the system was capable of extracting these features without any error at the output.

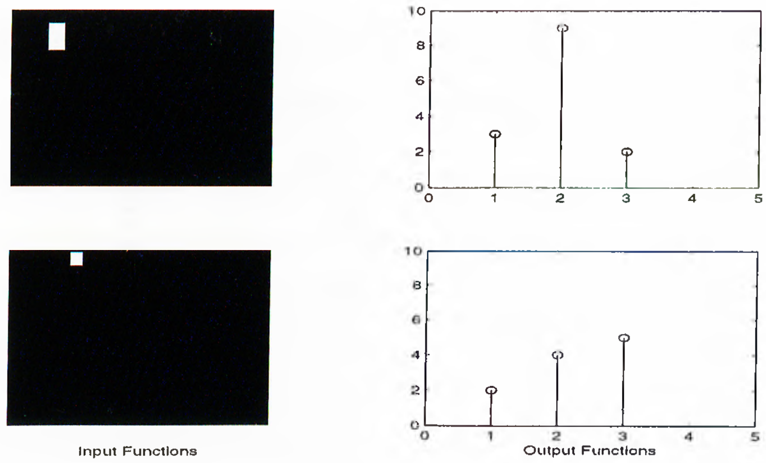

Figure 6.12: Examples of input and output pairs. Amplitude of the first peak represents the length in x-direction and the amplitudes of the following peaks represent the length in y-direction and the intensity respectively.

6.2 Simulations: Part II

All of the simulations in part I provided results about advantages and properties of different output representations. The implementation efficiency was not considered for those simulations. In second part of simulations, some of the examples used in part I are implemented with the two different techniques proposed in this thesis. Comparisons between these techniques are provided and the effect of number of filters, M, on the system performance is examined and discussed.

Example 1: The input signal is a binary pulse whose length and position information is going to be extracted. The maximum length of the pulse is 16. The length of the pulse is represented by peak amplitude coding and the position of the pulse is represented by peak position coding. A thresholding is applied at the outputs for the points used to represent the position of the pulse. It was observed that both the techniques were able to extract the length of the pulse without any error. Therefore, the errors obtained in finding the position of the pulse will be given here. The given errors show the average number in units of pixels.

Number of Filters	*Kernel Synthesis*	*Direct Design*
	Average Error in Position	*Average Error in Position*
1	2.87	2.16
2	1.25	0.83
3	0.83	0.69
4	0.80	0.47
5	0.56	0.03
6	0.25	0.02

For this simulation, it is more meaningful to choose the error criteria as the average error in position. The error obtained for the two different representation types are plotted in Figure 6.13. vs the number of filters It is seen that

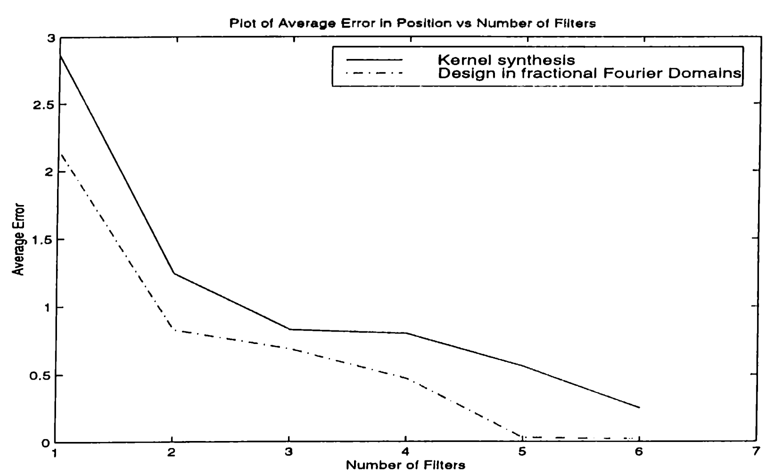

Figure 6.13: Comparison of two implementation techniques for increasing M..

the performance of implementing the system designed directly in the fractional Fourier domains is always better than the kernel synthesis which is as expected. Because, in kernel synthesis we apply the approximation of a system two times while we make the approximation only once for the direct design in fractional Fourier domains case. It is also observed that the performance of the system saturates when M is increased above a certain value. The saturation can be observed more easily when the error criteria is taken as the error in energy.

Example 2: In the second example, the inputs are again rectangular pulses whose lengths and intensities vary while the position is kept constant. The length information is represented by position coding and the intensity is represented by amplitude coding at the output. The maximum length of the input is 32. Therefore the output has length 33. It was observed that the system was able to recover the intensity information without any error. Therefore, the error obtained in finding the length will be given here. The error shows the average error in pixels.

Number of Filters	*Kernel Synthesis*	*Direct Design*
	Average Error in Length	*Average Error in Length*
1	20.2	11
2	14.4	7.7
3	2.1	1.2
4	1.35	0.2
5	0.26	0.16
6	0.10	0.08
7	0.13	0.075
8	0.12	0.08

It was observed that the system was capable of extracting the intensity information without any error as it is a linear operation to extract the intensity of such kind of inputs when it is represented by peak amplitude coding. The error arises from the extraction of the length information when it is represented by position coding. The results show that, the performances of designing and implementing the system directly in fractional Fourier domains is always better than the kernel synthesis.

It can be concluded from the simulations that, for a given accuracy we can obtain the number of filters, M, that provides the implementation of the system within that accuracy limit.

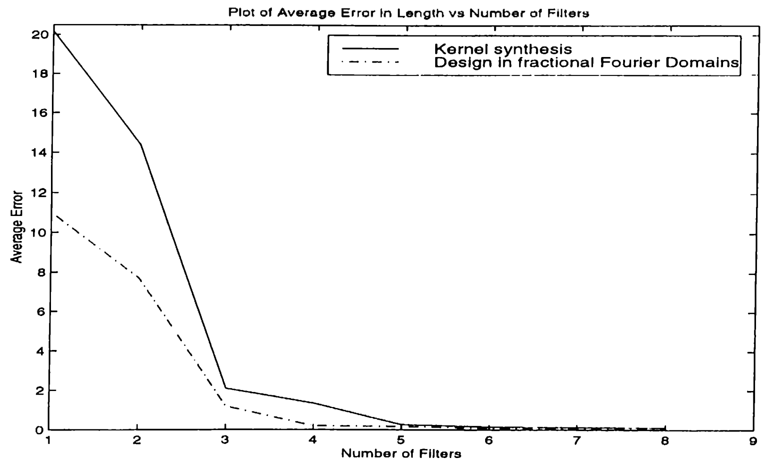

Figure 6.14: Comparison of two implementation techniques for increasing M .

As we use minimum mean-square error criteria when we are finding the fractional Fourier filter circuits either for kernel synthesis or direct design in fractional Fourier domains, the performance of the circuits approaches the performance of the kernel as M approaches N. Because,by increasing M, we can interpolate the system more accurately and reach the optimum linear approximation of the nonlinear problem, the kernel, when the error is considered as the energy difference. It should be noted that the kernel cannot also provide 0% error for most feature extraction problems as it seen from the first part of the simulations. Therefore, although the error obtained in terms of energy difference seems high for the above simulations, they are close to the optimum errors when a nonlinear system is approximated by a linear system. But in both of the above simulations, it is more meaningful to consider the error criteria as the average error for all inputs and the performances of the fractional Fourier filtering circuits are better in that case.

It should be noted that an optimization over the fractional Fourier transformation orders is not applied in these simulations. As the performance of the system is closely related with the fractional Fourier orders, there should be considerable developments in the system performance when an optimization is applied.

Chapter 7

Conclusion

In this thesis, it has been shown how the fractional Fourier transform can be applied to the feature extraction problem. A method which amounts to decomposing the overall feature extraction operation into a global linear and a local nonlinear system is proposed and discussed. Two different implementation techniques using fractional Fourier filter circuits to implement the global linear systems are provided.

The simulation results show that, employing different output representations to represent the input informations brings an additional degree of freedom to the feature extraction problem. All of the output representation types discussed in this thesis can be used for all information types, but some provides better results for specific informations. Therefore, many different output representations may be tested and the one providing the best performance can be used in different feature extraction problems.

The proposed system provides multi-outputs. That is, more than one feature of the input pattern is obtained at the same time. These multi-outputs can be displayed by a display panel where each display shows a different feature. Such kind of a structure may be beneficial in feature extraction applications.

Efficient implementation of the global linear part of the system is provided. The implementation is offered to be done in two different ways. The first techniques synthesizes the kernel with fractional Fourier filtering circuits and implements it with those filtering configurations. The second technique directly obtains the fractional Fourier filters that satisfy the input-output relations with minimum error with respect to a certain error minimization criteria. The simulation results show that, efficient implementations of the global linear system can be obtained with these two different techniques. The performance of the second technique is generally higher than kernel synthesis with fractional Fourier filters. The performances can be increased by applying an optimization over the fractional Fourier transformation orders.

The future work can be directed in two different ways. The first way consists of determining which output representation type provides the best performance for a particular information at the input. By this way, a library that shows which output representation is the best for which information type can be obtained and used in solutions of different image analysis problems.

The second direction for future work is to employ an optimization over the fractional Fourier transform orders and to combine parallel and repeated fractional Fourier filtering configurations to increase the performance of the system.

As a final word, feature extraction with fractional Fourier transform provides alternative efficient solutions to some feature extraction problems.

Bibliography

[1] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[2] D. Mendlovic and H. M. Ozaktas A. W. Lohmann. Fractional correlation. *Appl. Opt.*, 34:303–309, 1995.

[3] B. V. K. Vijaya Kumar. Tutorial survey of composite filter designs for optical correlators. *Appl. Opt.*, 31:4773–4800, 1992.

[4] D. Mendlovic, Z. Zalevsky, I. Kiryuschev, and G. Lebreton. Composite harmonic filters for scale-, projection-, and shift-invariant pattern recognition. *Appl. Opt.*, 34:310–316, 1995.

[5] J. Garcia, J. Campos, and C. Ferreira. Circular-harmonic minumum average correlation energy filter for color pattern recognition. *Appl. Opt.*, 33:2180–2187, 1994.

[6] Y. S. Cheng and T. C. Liang. Rotation invariant pattern recognition using a composite circulat harmonic and 2D isotropic Mexican-hat wavelet filter. *Opt. Commun.*, 112:9–15, 1994.

[7] G. Ravichandan and D. Cassasent. Minimum noise and correlation energy optical correlation filter. *Appl. Opt.*, 31:1823–1832, 1992.

[8] R. K. Wang, C. R. Chatwin, and M. Y. Huang. Modified filter synthetic discriminant functions for improved optical correlator performance. *Appl. Opt.*, 32:7646–7654, 1994.

[9] V. R. Riasati and M. A. G. Abushagur. Projection-slice synthetic discriminant functions for optical pattern recognition. *Appl. Opt.*, 36:3022–3034, 1997.

[10] A. Mahalanobis, B. V. K. Vijaya Kumar, and D. Cassasent. Minimum average correlation energy optical filters. *Appl. Opt.*, 31:3633–3640, 1987.

[11] C. F. Hester and D. Cassasent. Multivariant technique for multiclass pattern recognition. *Appl. Opt.*, 19:1758–1761, 1980.

[12] B. V. K. Vijaya Kumar. Minimum variance synthetic discrimant functions. *J. Opt. Soc. Am. A*, 3:1579–1584, 1986.

[13] Y. Ichioka, T. Iwaki, and K. Matsuoka. Optical information processing and beyond. *Proc. IEEE*, 84:694–717, 1996.

[14] D. Mendlovic, Y. Bitran, R. G. Dorsch, and A. W. Lohmann. Optical fractional correlation - experimental results. *J. Opt. Soc. Am. A*, 12:1665–1670, 1995.

[15] Y. Bitran, Z. Zalevsky, D. Mendlovic, and R. G. Dorsch. Fractional correlation operation - performance analysis. *Appl. Opt.*, 35:297–303, 1996.

[16] S. Granieri, R. Arizaga, and E. E. Sicre. Optical correlation-based on the fractional Fourier-transform. *Appl. Opt.*, 36:6636–6645, 1997.

[17] D. Mendlovic and H. M. Ozaktas. Fractional Fourier-transforms and their optical implementation .1. *J. Opt. Soc. Am. A*, 10:1875–1881, 1993.

[18] H. M. Ozaktas and D. Mendlovic. Fractional Fourier-transforms and their optical implementation .2. *J. Opt. Soc. Am. A*, 10:2522–2531, 1994.

[19] A. W. Lohmann, Z. Zalevsky, and D. Mendlovic. Synthesis of pattern-recognition filters for fractional Fourier processing. *Opt. Commun.*, 128:199–204, 1996.

[20] H. M. Ozaktas and D. Mendlovic. Fourier-transforms of fractional order and their optical interpretation. *Opt. Commun.*, 101:163–169, 1993.

[21] D. Mendlovic, H. M. Ozaktas, and A. W. Lohmann. Self Fourier functions and fractional Fourier-transforms. *Opt. Commun.*, 105:36–38, 1994.

[22] O. Aytur and H. M. Ozaktas. Nonorthogonal domains in phase-space of quantum optics and their relation to fractional Fourier-transforms. *Opt. Commun.*, 120:166–170, 1995.

[23] H. M. Ozaktas and O. Aytur. Fractional Fourier domains. *Signal Process.*, 46:119–124, 1995.

[24] M. A. Kutay, M. F. Erden, H. M. Ozaktas, O. Arıkan, Ö. Güleryüz, and Ç. Candan. Space-bandwidth efficient realizations of linear systems. *Opt. Letters*, 23:1069–1071, 1998.

[25] H. M. Ozaktas, O. Arıkan, M. A. Kutay, and G. Bozdağı. Digital computation of the fractional Fourier-transform. *IEEE Trans. Signal Process.*, 44:2141–2150, 1996.

[26] H. M. Ozaktas. Repeated fractional domain filtering is equivalent to repeated time and frequency domain filtering. *Signal Process.*, 54:81–84, 1996.

[27] G. Strang. *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, 1988.

[28] S. Y. Lee and H. H. Szu. Fractional Fourier transforms, wavelet transforms and adaptive neural networks. *Optical Eng.*, 36:2127–2136, 1997.

[29] H. M. Ozaktas. Towards an optimal foundation architechture for optoelectronic computing, Part I: Regularly interconnected device planes. *Appl. Opt.*, 36:5683–5696, 1997.

[30] H. M. Ozaktas. Towards an optimal foundation architecture for optoelectronic computing ,Part II: Physical construction and application platforms. *Appl. Opt.*, 36:5697–5705, 1997.