RECOGNITION OF PLANAR OBJECTS USING MULTIRESOLUTION ANALYSIS

by

Nazlı Güney

B.S., E.E., Boğaziçi University, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronic Engineering

Boğaziçi University

2003

ACKNOWLEDGEMENTS

I am grateful to my thesis supervisor Ayşın Baytan Ertüzün for being the wonderful person she is and letting me work on this exciting and challenging topic. She guided me with her ideas full of insight and experience.

I would like to thank the juri members Prof. Lale Akarun and Prof. Emin Anarım for participating in the juri without hesitance and for their positive remarks and ideas about possible future work.

Lastly, I thank my family for always believing that *I can.*

ABSTRACT

RECOGNITION OF PLANAR OBJECTS USING MULTIRESOLUTION ANALYSIS

Object recognition is one of the biggest problems in computer vision. In order to be able to solve this problem, many methods have been tried in the past. Wavelets, which have successfully been used in such areas as image compression, and denoising, have been applied to the object recognition problem as well. When planar objects are photographed from different view points, they seem to be going through an affine transformation. Therefore, an affine invariant wavelet function has been formulated in a recent paper for recognition of planar objects.

This work aims to implement the proposed invariant function with the multiwavelet transform to improve its results, for multiwavelets are known to have superior performance compared to scalar wavelets in other signal processing fields. A comparison of the changes in the performance of the affine invariant function when it is implemented with scalar and multiwavelets is done. The function has also been implemented with the combined wavelet coefficients obtained by using two different set of filters. A classical method, Fourier descriptors have been included in the simulations for comparison purposes as well.

ÖZET

DÜZLEMSEL NESNELERİN ÇOKLU ÇÖZÜNÜRLÜK ANALİZİ KULLANILARAK TANINMASI

Nesne tanıma, bilgisayarlı görü alanındaki en büyük sorunlardan birisidir. Bu sorunu çözebilmek için geçmişte pek çok yöntem denenmiştir. İmge sıkıştırma, gürültü giderimi gibi alanlarda başarılı bir şekilde uygulanmış olan dalgacıklar, nesne tanıma problemine de uygulanmıştır. Düzlemsel nesneler farklı noktalardan görüntülendiklerinde ilgin dönüşüme uğramış gibi görünmektedirler. Bu sebeple, düzlemsel nesnelerin tanınması için yakın zamanda yayınlanmış bir makalede bir ilgin değişimsiz fonksiyon oluşturulmuştur.

Bu çalışma önerilen değişimsiz fonksiyonu çoklu dalgacık dönüşümü ile gerçekleştirerek sonuçlarını geliştirmeyi amaçlamaktadır, çünkü çoklu dalgacık dönüşümünün diğer sinyal işleme alanlarında dalgacık dönüşümüne göre daha başarılı olduğu bilinmektedir. İlgin değişimsiz fonksiyonun tekli ve çoklu dalgacıklar ile gerçekleştirildiğinde başarımında olan değişiklikler karşılaştırılmaktadır. Fonksiyon iki farklı filtre seti kullanılmasıyla oluşan birleştirilmiş dalgacık katsayılarıyla da gerçekleştirilmektedir. Klasik bir metod olan Fourier tanımlayıcıları da karşılaştırma amacıyla deneylere eklenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

A	Affine transformation matrix with elements a_1, a_2, b_1 and b_2
a	Constant
B	Translation vector with elements a_0 and b_0
b	Constant
C	Vector scaling coefficients
$\mathcal{C}$	Path along a contour
c	Scaling coefficients
D	Vector wavelet coefficients
d	Wavelet coefficients
$f_{i,j}$	Invariant function of the original curve using scales i and j
$\tilde{f}_{i,j}$	Invariant function of the affine curve using scales i and j
G	Matrix wavelet (high pass) filter
g	Input function
H	Matrix scaling (low pass) filter
h	Scaling (low pass) filter
h_1	Wavelet (high pass) filter
I	Absolute invariant function
$L^2(\mathbb{R})$	Vector space of measurable, square integrable 1D functions
l	Number of multiscaling coefficient matrices
M	Length of filter
m	Number of multiwavelet coefficient matrices
N	Number of data samples
P	Symmetric conic matrix with elements p_{11}, p_{12} and p_{22}
Q	Symmetric conic matrix with elements q_{11}, q_{12} and q_{22}
R	Number of scaling functions in a multiwavelet representation
S	Lower bound of Sobolev smoothness
T	Total area enclosed by the affine curve
T^0	Total area enclosed by the original curve
t	Time, parameter

V_j	Subspace of $L^2(\mathbb{R})$ spanned by scaling functions at scale j
W_j	Subspace of $L^2(\mathbb{R})$ spanned by wavelets at scale j
$W_j g(t)$	Wavelet coefficients of a function $g(t)$ at scale j
$W_j g_i(t)$	Multiwavelet coefficients of $g(t)$ at scale j and i^{th} row
X	Fourier transform of $x(t)$
$\vec{X}$	Coordinate vector
$x(t)$	Function containing x-coordinates of the original curve
$\tilde{x}(t)$	Function containing x-coordinates of the affine curve
Y	Fourier transform of $y(t)$
$y(t)$	Function containing y-coordinates of the original curve
$\tilde{y}(t)$	Function containing y-coordinates of the affine curve
α	Multiplying constant used in repeated row processing
Γ	Wavelet based conic matrix with elements γ_{11}, γ_{12} and γ_{22}
η	Invariant function of the original curve using six scales
$\tilde{\eta}$	Invariant function of the affine curve using six scales
η_i	i^{th} component of η
θ	Angle in degrees
ξ	Arbitrary parametrization of a curve
ϕ	Scaling function
ψ	Wavelet function
Φ	Vector scaling (multiscaling) function
Ψ	Vector wavelet (multiwavelet) function
AC	Area Center
App	Approximation order
BIGHM2	Biorthogonal multi-filter bank obtained from GHM
BIH52S	Hermite cubic multi-filter bank used in analysis
CL	Chui-Lian orthogonal symmetric multi-filter bank
det	Determinant
DWT	Discrete Wavelet Transform

FFT	Fast Fourier Transform
GHM	Geronimo-Hardin-Massopust orthogonal multi-filter bank
IFFT	Inverse Fast Fourier Transformation
max	Maximum
min	Minimum
RDWT	Redundant Discrete Wavelet Transform
RMWT	Redundant Multiwavelet Transform
SA4	Orthogonal symmetric multi-filter bank by Shen, Tan, Tham
SNR	Signal-to-Noise Ratio

1. INTRODUCTION

1.1. Problem Statement: Object Recognition

Object recognition, which is one of the most challenging problems in computer vision, is yet truly unsolved despite all of the research done in the past. Obtaining recognition system performances approximating the perfect recognizer, the human being, is still very distant. On the other hand, designing systems suiting specific needs that take into account the conditions and circumstances under which recognition is to be performed, is possible. Typically, recognition has to be done when the object can be seen from different points of view producing totally different images, which should all be recognized as portraying the same object. This difficult task of recognizing objects from their images can be transformed to finding suitable invariants, which may be, for instance, shape descriptors that remain unchanged even when the images of objects are taken from different points. Hence, it may be said that object recognition is a search for invariants [1].

The invariants that should be extracted from the object will be projective invariants in this case, which are invariant to the viewpoint only. Demanding too general invariants, which are invariant to illumination, color, etc., will complicate matters, since their extraction will be unnecessarily more difficult. Projective invariants are preferred, therefore, for the projective transformation is the smallest group that includes all possible viewpoint related changes [1]. While the projective transformation is represented geometrically with projection or a projection plus translation, when the projection is parallel, the transformation becomes affine. The relationship between images taken from different directions is modelled with affine transformation when the depth of object along the line of sight is small compared to the viewing distance [2]. Hence, planar objects for which this assumption always holds seem to go through an affine transformation when their pictures are taken. Defining the problem in this study as the recognition of planar objects in 3D space from their shapes, the invariants to be searched for becomes those of affine transformations.

Various object recognition techniques have been proposed in the literature about this issue so far. While the shape descriptors used for the techniques may be either boundary or region based, except for a few (i.e. moment invariants [3]), almost all of those have been selected using the boundaries of objects solely, for this has the advantage of diminishing the amount of data to be processed without discarding valuable information. Some of those techniques use the boundary globally as one whole entity, while some of them process it locally. The global invariants tend to weaken immensely when some part of the object is missing, which is due to occlusion most of the time while the local invariants are less immune to noise when compared to global invariants [4]. This stems from the fact that traditionally global invariants utilize the coefficients that describe the relationship between the coordinates of the shape to be recognized, hence requires all of the coordinates for a full description of the curve. On the contrary, local invariants are based on derivatives at each point of the curve making them vulnerable to noise [1, 4].

In the classical literature on object recognition, the global and local invariants described above are the algebraic and differential invariants, respectively. Both of those use spatial domain information for object recognition, which corresponds basically to the coordinates of the boundary. Recognition in the transform domain has also been introduced as in the case of Fourier descriptors [5], and wavelet transform based methods [2, 4, 6]. Since the whole set of coordinates is used in computing each of the coefficients in the transform domain, such methods will be producing global invariants, and hence they will suffer from occlusion problems as well.

The basic idea behind one of the global invariant methods, Fourier descriptors, is that a closed curve may be represented with a periodic function of a continuous parameter, or alternatively with a set of Fourier coefficients of this function [5]. The set of Fourier coefficients is called the Fourier descriptors of the curve. For the Fourier descriptors to be affine invariant, the parameterization of the curve should be linear under an affine transformation. This is required, since Fourier transformation preserves the linearity of the affine transformation if and only if the parameterization is linear. For this purpose, the area parameterization is being used, which is developed in [7].

A matrix of coefficients is obtained for each boundary, where the rows correspond to the Fourier transforms of x and y coordinates. When two frequency indices are chosen, yet another matrix, which is 2×2 is obtained. If the determinant of this matrix is taken, the number obtained is a relative invariant, since it is related to the determinant of the relevant matrix of the affine curve via a factor, which is the same for any two frequency indices. The invariant may be made absolute by dividing with another relative invariant number. By fixing one of the frequency indices in the matrix that is to form the relative invariant number, and letting the other frequency index vary, a set of invariant numbers may be obtained. To obtain invariance to the starting point, a set of linear equations has to be solved. (See [5] for a discussion.) This complicates matters and for simplicity the phase term is avoided, since it contains information about the starting point. However, in such a case, there is the possibility that two different curves may end up with the same set of numbers violating the issue of completeness. Moreover, the discrimination between objects when this method of Fourier descriptors is used, will be based on standard statistical pattern recognition techniques, which require a training phase as well. This method, therefore, is not very attractive, for it is not truely invariant and it requires a lot of processing before and after an object has to be recognized.

The method in [4] is based on wavelet transformation of the coordinate functions of the boundary. Wavelet transformation has different scales as opposed to Fourier transformation, which has frequency indices. The invariant proposed depends on the fact that the area of the triangle formed by the wavelet coefficients at three different scales will produce a relatively affine invariant function. This function may be normalized with another function composed of a different set of coefficients. A match is chosen that maximizes the cross-correlation between two signatures, where one of the signatures is that of the unknown object and the second belongs to one of the reference objects in a database. However, invariance to starting point, and obtaining point correspondence issues are not addressed fully in the paper. Instead, for each reference object, all possible shifts of its signature is cross-correlated with the signature of the unknown object to assign the maximum of these correlation values to that reference object as the solution to the estimation of the starting point problem.

The work in [6] is also based on wavelet transformation. The approximation and detail signals at an arbitrary scale j forms the matrix this time, whose determinant is taken to form the affine invariant function. Since the method uses approximation signal as well, which depends on the position of the object in an image, the centroid of the boundary of the object is used as the origin of the coordinate plane. A dissimilarity function is defined, which attains bigger values as the dissimilarity between two objects increases. This function makes use of the extrema at each scale of the unknown object rather than the whole set of coefficients by comparing their positions and magnitudes with those of the reference objects'. The scales that have larger energy are preferred when obtaining the affine invariant representation. This should hold for all of the objects in the database, which naturally decreases the number of scales used. In the paper, point correspondence is tried to be obtained between the points of the boundary of an unknown object and points on the boundaries of objects in the database by using area parametrization. The points on both types of curves are selected so that the normalized area parameter values will be increasing uniformly. However, the problem of finding a starting point for the boundary has not been addressed.

An affine invariant wavelet function has been proposed in [2]. This function, which is based on a wavelet-based conic representation, uses 7-12 wavelet scales. The invariant of two wavelet-based conics is the affine invariant wavelet function. The coefficients of the conics are time-varying, where they are found by fitting one conic to three wavelet points, which are the wavelet transform coefficients of a single point in the time domain. Since the affine invariant function thus obtained is relatively invariant, it is made absolutely invariant by normalizing it with another relative invariant function. Depending on whether the wavelet scales of the two functions are allowed to overlap or not, the total number of wavelet scales varies between 7 and 12. By using a lot of wavelet scales, the discriminating power of the invariant function is tried to be increased. However, in the paper, point correspondence problem is not taken into consideration.

For recognition of planar objects in 3D space, which can be observed from such viewing points that unoccluded images of the objects are obtained, it will be wise to

use one of the global invariants instead of a local invariant, which are doomed to have a difficult time recognizing objects in the presence of noise.

1.2. Motivation for the Thesis

During the last decade, wavelets have been applied to numerous fields ranging from pure mathematics, differential equation solving, image compression to image denoising. This widespread appearance of wavelets have encouraged researchers to use wavelets in object recognition applications as well. Promising results have been reported in [2, 4, 6]. However, among the works listed, one of them, [2] stands out by trying to fulfill the requirement that as many of the dyadic wavelet levels as possible should be used in recognition of an object to be able to accurately represent it. This is accomplished with the introduction of the wavelet-based conic equation. The method in [2] produces a single affine invariant function for each object, which makes the recognition task very easy, since the maximum of the correlations between the unknown object and the reference images in the database will identify the unknown object. The computations are simple and there is no need to resort to standard statistical pattern recognition techniques to discriminate between different objects as in [5].

It was then a wonder whether the results obtained in [2] would improve when the multiwavelet transform was used instead of the wavelet transform. Multiwavelets, which are wavelets derived from multiple scaling functions, have shown superior performance compared to wavelets in such applications as image compression, image denoising, etc. due to their intrinsic properties. However, they have not been applied to object recognition yet. Moreover, the affine invariant function in [2] requires the usage of the redundant form of wavelet transform, which makes it necessary to develop a redundant form of multiwavelet transform. This may be accomplished analogous to the wavelet transform case by leaving out the decimation stages and upsampling the filter coefficients by padding zeros as explained in Chapter 5. The idea of using the multiwavelet transform in an object recognition application for the first time; moreover, developing a redundant form of it, are all very exciting and motivating.

1.3. Scope of the Thesis

In this study, the aim was to see how the performance of the affine invariant function proposed in [2] would change when multiwavelet transform was used instead of the wavelet transform. However, when the tests were done with real data, the results of the affine invariant wavelet function turned out to be very different than the ones in the paper [2]. The experiments in the paper were done by adding some noise to the affine transformed boundaries instead of taking the boundary of some affine transformed object. This simulates the situation that if a point matching algorithm is being used, exact point correspondence between the original curve and the curve extracted from the affine transformed object will not be achieved and some noise will be left. Thus, in order to make the affine invariant wavelet function achieve acceptable levels of performance when tested with real data, some additional steps have been added to the algorithm as well.

Chapter 2 of the thesis introduces wavelets and multiwavelets. How the theory of wavelets and wavelet transformation has been developed as well as characteristics of wavelets are explained. Multiwavelets, which are an extension of wavelets, and the way multiwavelet transform is implemented covers the last part of this chapter. Chapter 3 presents the work in [2] and attempts to interpret the steps followed when proposing the affine invariant function.

In Chapter 4, the additions and changes to the affine invariant wavelet function algorithm are given. First of all, a point correspondence algorithm based on the specific case when airplanes are the objects to be recognized has been added. Airplanes have been used as objects, since in the tradition of research, aircraft silhouettes have been used in object recognition applications. This algorithm initially tries to find a starting point for each curve, which is going to be the same for the original and affine curves, beginning from which the area parametrization in [5] is computed. If points on two curves, which are related by an affine transformation have the same normalized area parameter values, they are affine images of each other. This is how point correspondence is obtained. When real data is used in the experiments, the coarser dyadic

wavelet levels should be selected, for the finer ones will be very much affected by noise. Therefore, a change of dyadic levels has been done so as to select the ones that will be the least immune to noise.

In Chapter 5, after the performance of the affine invariant function has been increased to acceptable levels, it has been implemented with the redundant multiwavelet transform. For this purpose, in the beginning, the coefficients resulting from each wavelet function in the multiwavelet representation have been handled separately to produce a number of affine invariant functions. When such an approach did not produce good results, a way to combine the coefficients has been tried by selecting the coefficients corresponding to the coarsest levels of the multiwavelet transform and using them jointly in one function.

Chapter 6 contains the simulation results. The performances of the original function, the modified affine invariant wavelet function and affine invariant multiwavelet function can be compared by looking at the tabulated results. The wavelet filters in [8] and Daubechies-4 filters have been used. The chosen multiwavelets are Geronimo-Hardin-Massopust (GHM), Chui-Lian (CL) multiwavelets, as well as an orthogonal symmetric multi-filter bank constructed by Shen, Tan, and Tham, which is abbreviated as SA4, biorthogonal Hermite cubic filter bank, BIH52S, and a biorthogonal multi-filter bank derived from GHM, BIGHM2. In order to compare the performances of all of these functions, an experiment similar to the one in the paper [2] has been done as well, where each object in a database of 20 airplanes is being viewed at 100 different points with random noise added to its boundary to simulate imperfect point matching. The recognition power of each function is shown by how many of the different poses have ended up with correct matching. Finally, a comparison with the Fourier descriptors method is done. Chapter 7 concludes the thesis with a listing of possible future work.

2. WAVELETS AND MULTIWAVELETS

2.1. What are Wavelets?

Waves are defined as oscillating functions of time (or space), such as a sinusoid [9]. Fourier analysis, which has been widely used in mathematics, science and engineering, is the analysis of signals or functions in terms of sinusoids or equivalently, complex exponentials, which are combinations of sinusoids as in the famous Euler identity. Such an analysis is very useful when the signal being analyzed is stationary, which means that its characteristics, frequency content in this case, do not change with time or space. For the analysis of nonstationary signals, whose characteristics are dependent on time, a technique that allows simultaneous time and frequency analysis is necessary to be able to know the frequency content of the signal at a certain instant of time. To accomplish this, instead of allowing waves to extend through the whole time or space axis oscillating with equal amplitudes as in the Fourier analysis case, they are restricted to a region of time or space. This defines the wavelet: Wavelet is a "small wave", which has its energy concentrated in time [9]. These concepts are illustrated in Figure 2.1, where the plots on the left and right refer to a wave and a wavelet, respectively.

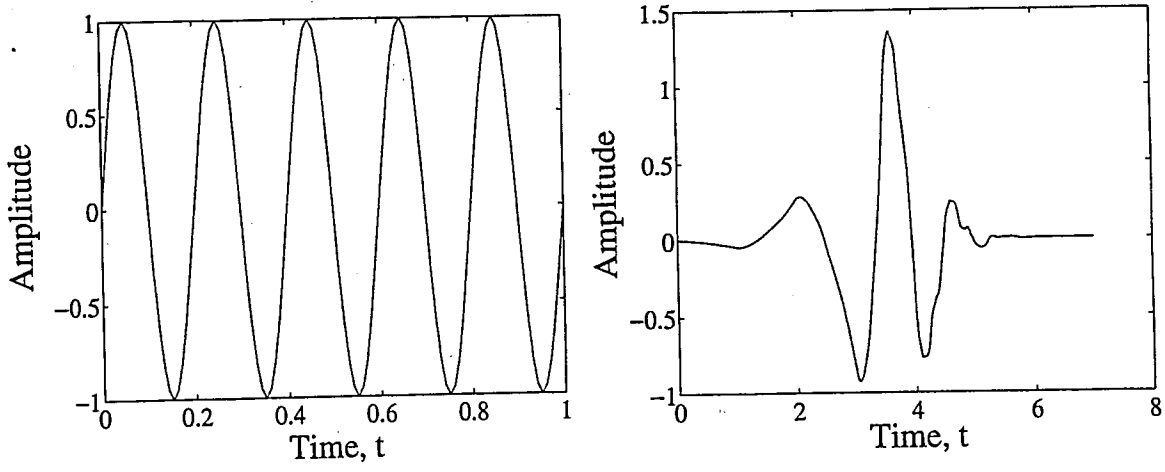

Figure 2.1. A wave and a wavelet

As can be seen above although the wavelet has finite energy concentrated around

a point, it still exhibits oscillating wave-like characteristics.

2.2. Wavelet Expansion Systems

Wavelets are used in a series expansion of signals much like the way waves are used in Fourier series. One of the main reasons why signals are described by their series expansions rather than their time domain representations is that very few coefficients may provide some useful information, which is not even so obvious in the time domain. For instance, when the signal is a sinusoid at a fixed frequency, the Fourier series is composed of one large coefficient at that frequency. Therefore, one may say that Fourier analysis is capable of frequency localization although it lacks time localization, since it uses a basis composed of waves, which extend through the whole time axis. Wavelets have appeared in the literature in late 1980s to fill this gap by providing simultaneous time and frequency localization. Wavelet representation is very much like a musical score where the location of notes tells when the tones occur and what their frequencies are [9]. Such a representation is obtained by mapping a one-dimensional function into a two dimensional array of coefficients. For the wavelet expansion of a signal $g(t)$, a two parameter system is constructed as

$$g(t) = \sum_k \sum_j d_{j,k} \psi_{j,k}(t), \tag{2.1}$$

where $\psi_{j,k}(t)$ are the wavelet expansion functions. The set of coefficients $d_{j,k}$ is called the discrete wavelet transform of $g(t)$ and Equation 2.1 is its inverse transformation.

2.2.1. Characteristics of Wavelet Systems

The wavelet expansion set, hence, the discrete wavelet transform of a signal, is not unique as in the case of Fourier transformation, where the expansion set consists solely of sinusoids or complex exponentials. Thus, there are many possible different wavelet systems depending on the set used, but they all share some common characteristics:

- Wavelets have first been introduced by Grossmann and Morlet in 1984 as functions $\psi(t)$ whose translations and dilations can be used for expansions of functions in $L^2(\mathbb{R})$, which is the vector space of measurable, square integrable one-dimensional functions [10]. In analogy with the first wavelet system, all first generation systems are generated from a single scaling or wavelet function by simple scaling and translation. The two dimensional parametrization in Equation 2.1 is achieved from the function $\psi(t)$, which is sometimes called the generating or mother wavelet, by

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \tag{2.2}$$

  where both j and k are integers. The norms of the so generated wavelets are the same as that of the generating wavelet independent of the scale j due to the factor $2^{j/2}$.

- All useful wavelet systems satisfy the multiresolution conditions, since this has been the motivating factor for the emergence of wavelet systems in the beginning. Multiresolution dictates that if a set of signals can be represented by a weighted sum of $\varphi(t-k)$, which are translates of a function $\varphi(t)$, then a larger set including the original one can be represented with a weighted sum of $\varphi(2t-k)$. What this means is that if the original expansion set signals are made half as wide and are translated in half as wide steps, they will represent a wider set of signals exactly or will be able to give a better representation of any signal.

  The multiresolution representation has been very useful in image processing by providing a simple hierarchical framework for interpreting the image information [10]. At different resolutions (i.e. scales j), the details of an image usually correspond to different physical structures of the scene in the image. While at a coarse resolution the details are those of the larger structures, as the resolution increases, details pertaining to smaller structures start to appear. Analyzing images first at coarse resolutions and then gradually increasing the resolution has been useful in pattern recognition applications.

- The lower resolution coefficients can be efficiently calculated from the higher resolution ones by a tree-structured algorithm called the filter bank. This has

first been suggested by Mallat in his famous paper [10]. The filter bank consists of low and high pass filters.

2.3. The Discrete Wavelet Transform

The concept of resolution is useful for a better understanding of the mathematics and interpretation of wavelets. In order to be able to use the ideas of multiresolution, instead of starting with the wavelet, traditionally, a scaling function is defined from which the wavelet is obtained. A basic scaling function defines a set of scaling functions, which are integer translates of it, as

$$\varphi_k(t) = \varphi(t-k) \qquad k \in \mathbf{Z} \qquad \varphi \in L^2(\mathbb{R}), \tag{2.3}$$

where $\mathbf{Z}$ represents the set of integers while $L^2(\mathbb{R})$, which was mentioned before in the text, is an important vector space in signal processing, which is the space of all functions with a well defined integral of the square of the modulus of the function [9]. Here, "L" signifies a Lebesque integral, the "2" denotes the integral of the square of the modulus of the function and $\mathbb{R}$ shows that the independent variable of the integration is a number over the real line.

If the subspace of $L^2(\mathbb{R})$ spanned by these functions is defined to be

$$V_0 = Span_k\{\varphi_k(t)\}, \tag{2.4}$$

then it can be said that

$$g(t) = \sum_k c_k \varphi_k(t) \qquad \text{for any} \qquad g(t) \in V_0. \tag{2.5}$$

The subspace spanned can be increased or decreased in size when the time scale of the scaling functions is changed. A basic scaling function defines a two dimensional

family of functions when its time scale is allowed to change as well:

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k). \tag{2.6}$$

The subspace of $L^2(\mathbb{R})$ spanned by these functions is

$$V_j = Span_k\{\varphi_k(2^j t)\} = Span_k\{\varphi_{j,k}(t)\} \qquad k \in \mathbf{Z}. \tag{2.7}$$

When $j > 0$, the spanned subspace will be larger because the $\varphi_{j,k}(t)$ are narrower and are translated in narrower steps, which will enable them to represent much more detail. The coarser information will be represented by the wider scaling functions with $j < 0$, whose span will be much smaller than those of the narrower ones. This may be summarized by the following:

$$V_j \subset V_{j+1} \qquad \text{for all} \qquad j \in \mathbf{Z} \tag{2.8}$$

with

$$V_{-\infty} = \{0\}, \qquad V_{\infty} = L^2. \tag{2.9}$$

In one of the two limiting cases, $j = -\infty$, the scaling functions extend over the whole real line with zero amplitude and hence can represent only the trivial space $\{0\}$, whereas in the other case of $j = \infty$, the scaling functions approximate delta diracs placed at all values of t and can span the whole $L^2(\mathbb{R})$.

Thus, a nesting of subspaces can be done as

$$\{0\} \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2. \tag{2.10}$$

The nesting of subspaces implies that if a function is in the subspace V_0 spanned by $\varphi(t-k)$, then it is also in V_1 spanned by $\varphi(2t-k)$, which is a larger one. Since it is known that $\varphi(t)$ is in V_0 naturally, then it can be expressed as a weighted sum of

shifted $\varphi(2t)$, as

$$\varphi(t) = \sum_n h(n)\sqrt{2}\varphi(2t-n), \qquad n \in \mathbf{Z}, \tag{2.11}$$

where the coefficients $h(n)$ are called the scaling function coefficients, or the scaling filter. Equation 2.11 is a fundamental one and it is referred to as the multiresolution analysis equation (MRA), the refinement equation, or the dilation equation depending on the point of view employed.

The important features in a signal are better analyzed by not increasing the scale from j to ∞, but looking at the details added as the scale is increased. Therefore, a set of functions, which span the differences between the subspaces at various scales is necessary. These functions are the wavelets $\psi_{j,k}(t)$.

The orthogonal complement of V_j in V_{j+1} is defined to be W_j, which requires that all of the members of V_j are orthogonal to all of the members of W_j. This is stated mathematically as

$$\int \varphi_{j,k}(t)\psi_{j,l}(t)\mathrm{d}t = 0. \tag{2.12}$$

Starting from scale $j = 0$, the subspace spanned by the wavelets $\psi(t-k)$, W_0, is defined in a way so that

$$V_1 = V_0 \oplus W_0, \tag{2.13}$$

where $\oplus$ shows that V_1 is the direct sum of the subspaces V_0 and W_0. In general, the subspaces satisfy the following relationship

$$L^2 = \underbrace{\underbrace{\underbrace{V_0 \oplus W_0}_{V_1} \oplus W_1}_{V_2} \oplus W_2}_{V_3} \oplus \cdots . \tag{2.14}$$

The starting scale j is arbitrarily chosen and when it is $-\infty$, Equation 2.14 becomes

$$L^2 = \cdots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots, \tag{2.15}$$

where no scaling subspace is necessary and a signal $g(t) \in L^2(\mathbb{R})$ can be represented with Equation 2.1.

The projection of a signal $g(t)$ onto the V_j space is an *approximation* signal of $g(t)$ at scale j, whereas the projection of it onto the W_j space produces the *detail* signal at scale j.

Since the wavelet $\psi(t)$ resides in the next subspace spanned by the narrower scaling functions $\varphi(2t-k)$, it can be said that $W_0 \subset V_1$ and $\psi(t)$ can be represented as a weighted sum of these scaled and translated scaling functions for some $h_1(n)$:

$$\psi(t) = \sum_n h_1(n)\sqrt{2}\varphi(2t-n), \qquad n \in \mathbf{Z}. \tag{2.16}$$

A set of functions $\varphi_{j,k}(t)$ and $\psi_{j,k}(t)$ have been constructed that can span the whole $L^2(\mathbb{R})$. Generalizing Equation 2.14 so that the starting scale j is arbitrary, say j_0, any function $g(t) \in L^2(\mathbb{R})$ can be written as

$$g(t) = \sum_k c_{j_0}(k)2^{j_0/2}\varphi(2^{j_0}t-k) + \sum_k \sum_{j=j_0}^{\infty} d_j(k)2^{j/2}\psi(2^j t-k) \tag{2.17}$$

or equivalently

$$g(t) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{\infty} d_j(k)\psi_{j,k}(t). \tag{2.18}$$

The coefficients $c_{j_0}(k)$ and $d_j(k)$ in Equation 2.18 are the discrete wavelet transform (DWT) coefficients of the signal $g(t)$ in its most general form. These coefficients may be used to completely describe the original signal like the Fourier coefficients. However, unlike Fourier coefficients, wavelet coefficients may be used to describe nonstationary

as well as periodic, stationary phenomena.

2.4. Filter Banks and Their Relation to the Discrete Wavelet Transform

In almost all applications, one never needs to work directly with either scaling functions or wavelets. The scaling function and wavelet coefficients in Equations 2.11 and 2.16 as well as $c_{j_0}(k)$ and $d_j(k)$ in Equation 2.18 are sufficient. To be able to work directly with the DWT coefficients, however, the lower scale coefficients and higher scale coefficients have to be defined in terms of each other. This has been achieved by using some math and such facts like scaling functions and wavelets are orthogonal to each other to produce the following:

$$c_j(k) = \sum_m h(m-2k)c_{j+1}(m) \tag{2.19}$$

$$d_j(k) = \sum_m h_1(m-2k)c_{j+1}(m). \tag{2.20}$$

where c's are scaling and d's are wavelet coefficients.

The first and second of the equations above describe how the lower order scaling and wavelet coefficients, respectively, can be obtained from the higher order scaling coefficients. These are called analysis equations. The derivation of these can be found either in [9] or [10].

Equations 2.19 and 2.20 show that when scale $j+1$ scaling coefficients are filtered (convolved) with $h(-n)$ and $h_1(-n)$ in two separate branches and decimated by two, scale j scaling and wavelet coefficients, respectively, are obtained. The implementation of these equations with filter banks has been called Mallat's algorithm [10], which is shown in Figure 2.2 for two consecutive stages.

Using two filters doubles the total number of coefficients, which is reduced to its initial value by the decimation stage. The downsampling after each filtering stage ensures that the total number of coefficients produced is the same as the number of

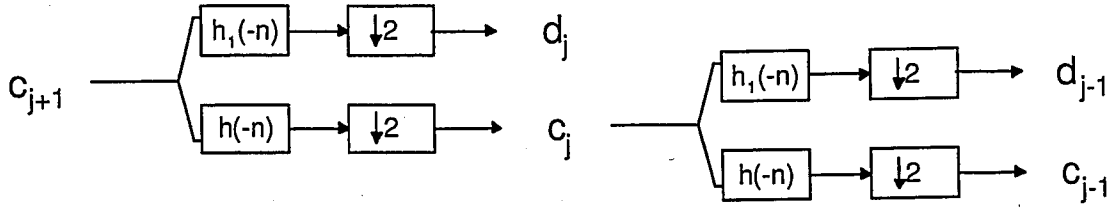

Figure 2.2. Two-stage wavelet analysis filter bank

samples in the original signal. Thus, there should be no information loss.

In the frequency domain, the first stage of two banks divides the spectrum of $c_{j+1}(k)$ into a lowpass and a highpass component, since the FIR filter implemented by $h(-n)$ is a lowpass filter and the one implemented by $h_1(-n)$ is a highpass filter. While the lowpass portion corresponds to scaling coefficients, the highpass filtered part represents wavelet coefficients. The second stage, then, divides the lowpass portion into another lowpass and a bandpass component. This is illustrated in Figure 2.3.

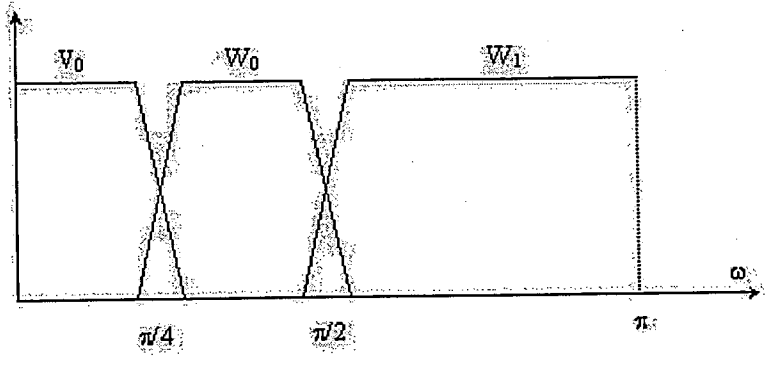

Figure 2.3. Frequency bands of the analysis tree

The input set of scaling coefficients $c_{j+1}(k)$ to be used in the analysis system in Figure 2.2 are chosen as the samples of the signal $g(t)$ [9], since at high enough scale, the scaling functions will act as delta functions, and if the samples of $g(t)$ are taken above the Nyquist rate, they will be good approximations to the scaling coefficients at that scale. Hence, no wavelet coefficients are necessary at that scale and the analysis stage can start from that scale. The signal can be analyzed until only one scaling and one wavelet coefficient are left. If the signal has 2^L samples, for instance, the analysis will include L scales at most.

The synthesis of higher scale coefficients from the lower ones, on the other hand, is done via the equation

$$c_{j+1}(k) = \sum_m c_j(m)h(k-2m) + \sum_m d_j(m)h_1(k-2m). \tag{2.21}$$

This corresponds to an upsampling of scaling and wavelet coefficients after which, they are filtered with $h(n)$ and $h_1(n)$, respectively. Finally, their sum gives the next higher scale scaling coefficients. Figure 2.4 describes this synthesis of scaling coefficients scheme.

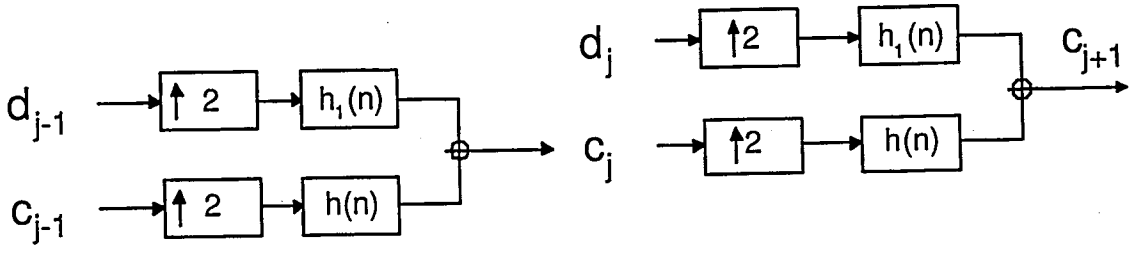

Figure 2.4. Two-stage wavelet synthesis filter bank

2.5. Multiwavelets

The multiresolution analysis for the $L^2(\mathbb{R})$ done until now have assumed a nesting of subspaces as in Equation 2.10, where each subspace is spanned by translations of scaled versions of a single scaling function. The difference between the subspaces are spanned by translations of a single wavelet. A more general multiresolution analysis setting can be assumed when several scaling functions are allowed to span each of the subspaces. Similarly, the difference subspaces are spanned by several wavelets. This idea leads to multiwavelets [11].

Multiwavelets have several advantages compared to wavelets. For example, such features as compact support, orthogonality, symmetry, and big number of vanishing moments, which are known to be important in signal processing cannot all be possessed at the same time by wavelets [11]. Multiwavelets, on the other hand, which are an extension of wavelets can possess these characteristics simultaneously, which may mean

that they are going to be more successful than wavelets in various applications.

If it is assumed that V_0 is spanned by translations of R different scaling functions $\varphi_i(t),\ i = 1, \ldots, R$, the scaling and translation of these functions produces:

$$\varphi_{i,j,k}(t) = 2^{j/2}\varphi_i(2^j t - k). \tag{2.22}$$

The multiresolution analysis forms subspaces of $L^2(\mathbb{R})$:

$$V_j = Span_k\{\varphi_{i,j,k}(t),\ i = 1, \ldots, R\}. \tag{2.23}$$

A vector scaling function is defined as

$$\Phi(t) = [\varphi_1(t), \ldots, \varphi_R(t)]^T. \tag{2.24}$$

where each row of the vector scaling function corresponds to a single scaling function. Similarly, since $V_0 \subset V_1$, it can be said that

$$\Phi(t) = \sqrt{2}\sum_n H(n)\Phi(2t - n), \tag{2.25}$$

where $H(k)$ is an $R \times R$ matrix. This is the matrix version of the scalar dilation equation, Equation 2.11, and hence may be called the matrix dilation equation as in [11].

Furthermore, it is assumed that the difference subspaces W_0 are spanned by R different wavelets. Since $W_0 \subset V_1$, there exists a unique sequence of $R \times R$ matrices such that the following holds:

$$\Psi(t) = \sqrt{2}\sum_n G(n)\Phi(2t - n). \tag{2.26}$$

Again, this is the vector version of Equation 2.16.

2.5.1. Implementation of the Multiwavelet Transform

If the expansion coefficients of the multiscaling and multiwavelet functions are defined to be $c_{i,j}(k)$ and $d_{i,j}(k)$, vectors of these coefficients may be formed as:

$$C_j(k) = [c_{1,j}(k), \ldots, c_{R,j}(k)]^T \tag{2.27}$$

$$D_j(k) = [d_{1,j}(k), \ldots, d_{R,j}(k)]^T. \tag{2.28}$$

If there exists a function $g(t)$ such that $g(t) \in V_{j_0}$, it can be written as a linear combination of the multiscaling functions and multiwavelets:

$$f(t) = \sum_k C_{j_0}(k)^T \Phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty} \sum_k D_j(k)^T \Psi_{j,k}(t). \tag{2.29}$$

Analogous to the scalar wavelet case, the calculation of the lower scale multiscaling and multiwavelet coefficients can be done by using the next higher scale multiscaling coefficients by using the set of equations [12]:

$$C_{j-1}(k) = \sqrt{2} \sum_m H(m-2k) C_j(m) \tag{2.30}$$

$$D_{j-1}(k) = \sqrt{2} \sum_m G(m-2k) C_j(m). \tag{2.31}$$

The analysis equations can be realized as a matrix filter bank, which is demonstrated in Figure 2.5.

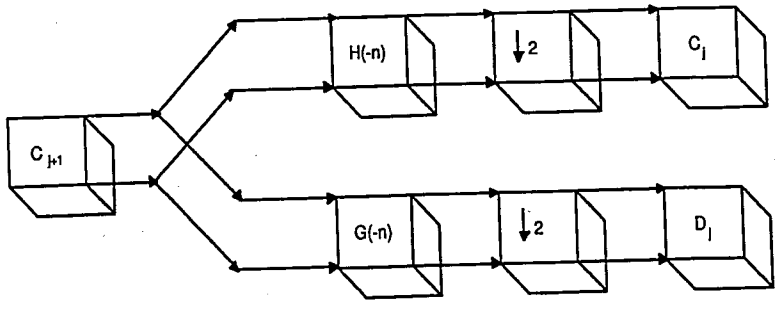

Figure 2.5. Analysis filter bank for multiwavelets

The reconstruction of higher scale coefficients is done via the formula:

$$C_{j+1}(k) = \sqrt{2} \sum_{m} H^T(k-2m)C_j(m) + G^T(k-2m)D_j(m). \tag{2.32}$$

The synthesis equation is implemented with a matrix filter bank as demonstrated in Figure 2.6.

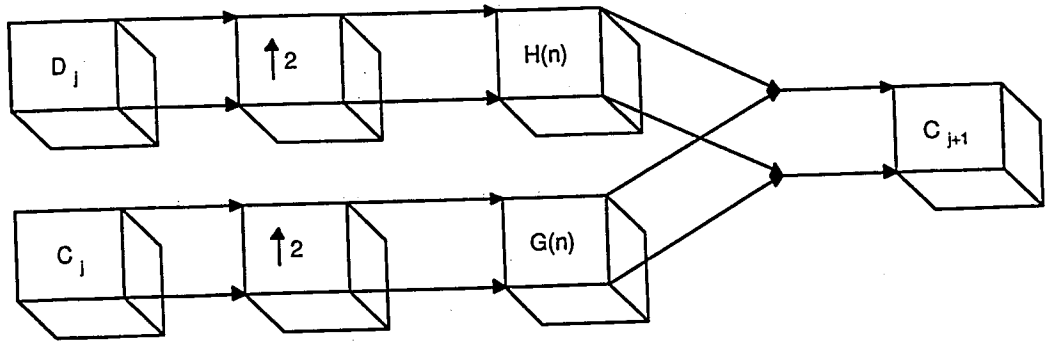

Figure 2.6. Synthesis filter bank for multiwavelets

The analysis and synthesis equations are similar in structure to their scalar counterparts. The main difference is that, this time, the filter banks operate on blocks of R inputs and all of the operations are done in terms of blocks of inputs. Therefore, to start the multiwavelet transform, initial estimates of multiscaling coefficients have to be obtained from a one dimensional signal to form a block of R one dimensional inputs. Data samples of the signal are preprocessed (prefiltered) to produce reasonable values of the expansion coefficients for the multiscaling function at the highest scale.

2.5.2. Preprocessing Methods

A number of preprocessing techniques are widely used in multiwavelet applications. One of those is the repeated row, and the other is the approximation preprocessing. While the approximation preprocessing preserves the number of data samples, the repeated row preprocessing increases the number of data samples from N to RN. A brief overview of these two schemes follows, where it is assumed that R has a value of 2.

2.5.2.1. Repeated Row Preprocessing: Oversampling. The most obvious way to get a second row of input is to repeat the first one and use two identical rows of inputs. Such a representation requires more calculations than a critically sampled one, since the amount of data has been increased to twice its original value, so when the aim is to compress data, this method, which introduces further redundancy, may not be preferred [11]. Another approach would be to produce the second row by repeating a scaled version of the first row [12] as:

$$C_j(k) = \begin{bmatrix} g(n) \\ \alpha g(n) \end{bmatrix}, \tag{2.33}$$

where $g(n)$ represent data samples and α is a constant, which is chosen so that when the input is a constant vector, the output from the highpass multifilter is zero. For the GHM multiwavelet [13], for instance, this constant is calculated to be $\alpha = 1/\sqrt{2}$.

2.5.2.2. Approximation Preprocessing: Critical Sampling. This scheme, which is based on the approximation properties of continuous-time multiwavelets as can be deduced from its name, yields a critically sampled signal representation. It has first been suggested by J. Geronimo and is naturally connected with the GHM multiwavelets. The estimated coefficients for the GHM multiwavelets are shown below, but the procedure can be repeated for other multiwavelets as well, by referring to either [11] or [12].

Supposing that the input sequence $g[n]$ contains samples of the function $g(t)$ at half integers so that

$$g[2n] = g(n), \qquad g[2n+1] = g(n+1/2). \tag{2.34}$$

Then, the estimated multiscaling coefficients at the highest scale are found from the set of equations:

$$c_1(k) = \frac{\varphi_2(1)g[2k-1] - \varphi_2(1/2)(g[2k-2] + g[2k])}{\varphi_2(1)\varphi_0(1/2)} \tag{2.35}$$

$$c_2(k) = \frac{g[2k]}{\varphi_2(1)}, \tag{2.36}$$

where $\varphi_1(t)$ and $\varphi_2(t)$ are the scaling functions of the multiwavelet system.

3. AFFINE INVARIANT WAVELET FUNCTION

Recognition of objects from their images taken at different viewing points than their reference image in a database is a very important problem in computer vision. When the objects that are to be recognized are planar in shape, the objects seem to be going through an affine transformation, which is a subgroup of the more general projective transformation.

Object recognition techniques are classified as local or global, depending on whether they use global features such as homogenous polynomial coefficients, Fourier descriptors, wavelet transform coefficients, moments, etc., or local ones such as critical points or local derivatives [4]. Local techniques have very weak performances in the presence of noise, since finding critical points or taking high order derivatives will be problematic in these cases. On the other hand, if a clear, unoccluded image of the unknown object cannot be obtained, it will be very difficult to do a correct recognition when one of the global techniques is used. While some of the techniques are region-based, which use the information in the image region corresponding to the object, such as the moment based method in [3], most of the others are boundary based. The boundary based methods are encountered more often in the literature, for the boundary contains enough information to accurately represent the object, its shape, so that most of the redundant information in the image region can be discarded.

In the following, it is assumed that the unknown objects are planar, and hence they have to be recognized from images, which contain affine transformed versions of some reference images in a database. The boundaries of objects, which can be modelled with $2D$ curves, are analyzed. When the image of an unknown object is produced, each point $(x(t), y(t))$ on the original curve obtained from the reference image in the database is mapped to a point $(\tilde{x}(t), \tilde{y}(t))$ on the affine transformed curve in the new

image via the equations:

$$\tilde{x}(t) = a_0 + a_1 x(t) + a_2 y(t) \tag{3.1}$$

$$\tilde{y}(t) = b_0 + b_1 x(t) + b_2 y(t). \tag{3.2}$$

Equations 3.1 and 3.2 can be combined into one equation:

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = A \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + B, \tag{3.3}$$

where A is a nonsingular square matrix representing the rotation, scaling and skewing in the affine transformation, and vector B represents translation.

In the rest of this chapter, the affine invariant function proposed in [2] will be introduced.

3.1. An Affine Invariant Wavelet Function Using Two Scales

The relationship between the coordinates of the boundary of the object in the database and those of the boundary of the affine transformed object is given by Equation 3.3. To be able to define an affine invariant wavelet function, which uses wavelet coefficients, the wavelet transform of both sides of Equation 3.3 has to be taken:

$$W_i\tilde{x}(t) = a_1 W_i x(t) + a_2 W_i y(t) \tag{3.4}$$

$$W_i\tilde{y}(t) = b_1 W_i x(t) + b_2 W_i y(t). \tag{3.5}$$

In matrix form Equations 3.4 and 3.5 become:

$$\begin{bmatrix} W_i\tilde{x}(t) \\ W_i\tilde{y}(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} W_i x(t) \\ W_i y(t) \end{bmatrix} = A \begin{bmatrix} W_i x(t) \\ W_i y(t) \end{bmatrix}. \tag{3.6}$$

Equation 3.6 shows the relationship between the wavelet coefficients of the coordinate functions of the two curves at a certain scale i. Using wavelet coefficients (the detail signal) has the advantage that the unknown parameters of the B vector in Equation 3.3, which carry information about the translation the affine curve has gone through, have been gotten rid of, since $W_i a_0 = W_i b_0 = 0$ due to the fact that scaling functions are capable of approximating constant functions and hence no wavelet coefficients are needed. Then, the affine invariant function using wavelet coefficients at scales i, j may be defined as:

$$f_{i,j}(t) = W_i x(t) W_j y(t) - W_i y(t) W_j x(t), \qquad i \neq j, \tag{3.7}$$

since

$$\det\left(\begin{bmatrix} W_i\tilde{x}(t) & W_j\tilde{x}(t) \\ W_i\tilde{y}(t) & W_j\tilde{y}(t) \end{bmatrix}\right) = \det\left(\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}\begin{bmatrix} W_i x(t) & W_j x(t) \\ W_i y(t) & W_j y(t) \end{bmatrix}\right) \tag{3.8}$$

$$\tilde{f}_{i,j}(t) = \det A \cdot f_{i,j}(t). \tag{3.9}$$

Due to the fact that the two functions obtained from the original and transformed curves are not exactly equal, but related with a constant term, this function is *relatively* invariant. If it had been the case that the two functions were identical, the function in Equation 3.7 would be an *absolute* invariant. By defining another function in terms of this relative invariant function, which will cancel out the unnecessary term $\det A$ that still carries the effects of the affine transformation, an absolute invariant may be obtained. In the simplest scenario, this may be done by dividing the relative function with another relative invariant function composed of different wavelet coefficients, like

$$\frac{\tilde{f}_{i,j}(t)}{\tilde{f}_{k,l}(t)} = \frac{f_{i,j}(t)}{f_{k,l}(t)} \qquad i \neq j, k \neq l. \tag{3.10}$$

The invariance of the function $f_{i,j}(t)$ can be demonstrated with the following figures. In Figure 3.1, the boundary of the object in the original image and an affine transformed version of it, which has been obtained by using Equation 3.3 directly.

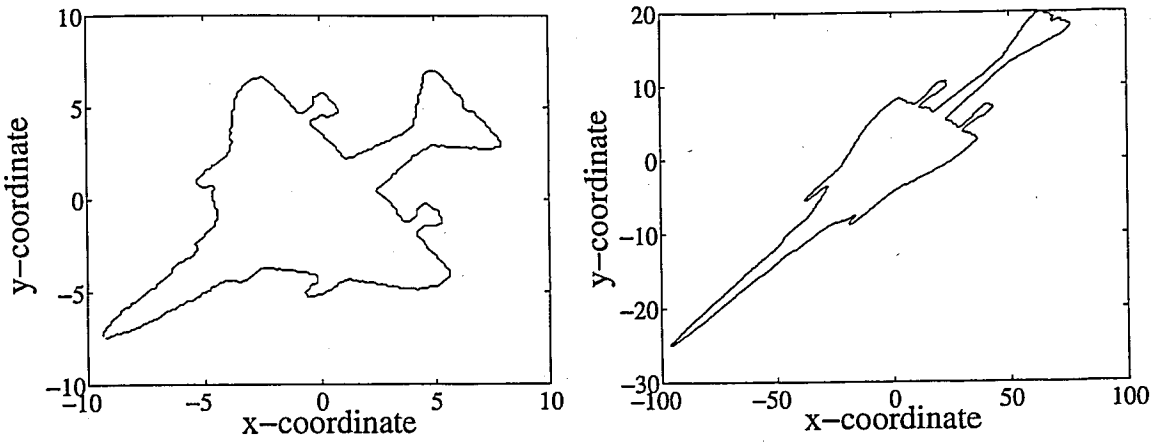

Figure 3.1. The original and affine transformed boundaries

When the relative invariant function in Equation 3.7 is calculated and plotted, Figure 3.2 is produced:

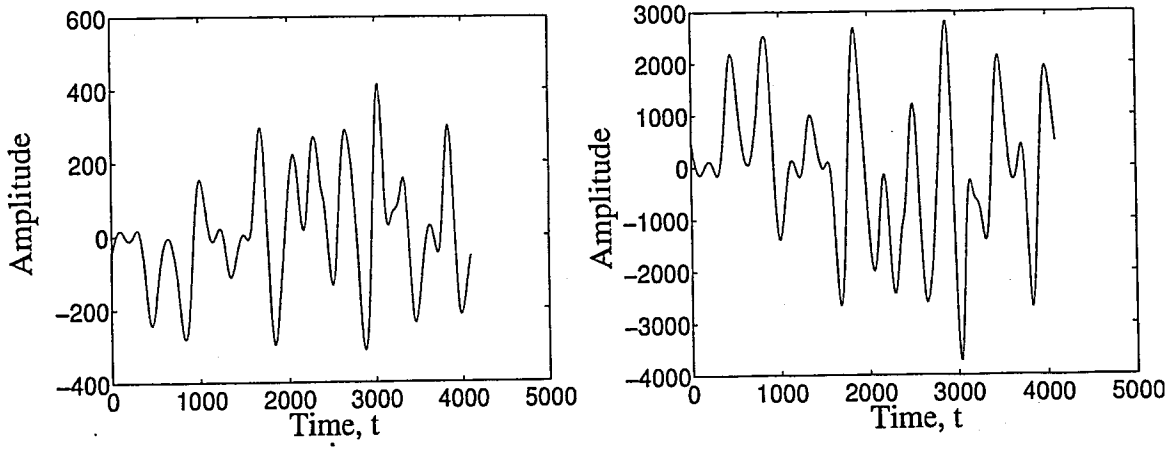

Figure 3.2. The relative invariant functions $f_{i,j}(t)$ and $\tilde{f}_{i,j}(t)$

The relative invariant function extracted from the transformed boundary is scaled and flipped with respect to the one extracted from the original one, which is due to the fact $\det A$ may have negative values.

When the discrete wavelet transform of a signal is taken, the wavelet coefficients at several scales are found. Although it is possible that different objects may have similar wavelet coefficients at a certain scale, it is impossible that this will hold for all scales. Therefore, if an affine invariant function using more of the scales in the discrete

wavelet transform is derived, it should have increased discrimination power.

3.2. An Affine Invariant Wavelet Function Using Six Scales

This function employs the idea of invariance of conics under projective transformation. Invariants of conics have been derived and widely used in the computer vision community. A listing of invariants of conics may be found in [1].

In $2D$, a conic is defined as the quadratic form:

$$\begin{bmatrix} x & y \end{bmatrix} P \begin{bmatrix} x \\ y \end{bmatrix} = a, \qquad \text{where} \qquad P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}. \tag{3.11}$$

Here a is a constant and P is a symmetric matrix. Equation 3.11 may be turned into a second order polynomial:

$$p_{11}x^2 + 2p_{12}xy + p_{22}y^2 - a = 0. \tag{3.12}$$

This form makes it clear that since there are no terms with x or y alone, it is assumed in Equation 3.11 that the center of the conic is at the origin.

The affine invariant function for a conic is defined to be $p_{11}p_{22} - p_{12}^2$ [1]. However, the joint invariant function for two conics, which have the symmetric matrices P and Q, where

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix}, \tag{3.13}$$

and P is as defined in Equation 3.11, is [14]:

$$Trace(Q^{-1}P) = Trace(P^{-1}Q) = p_{11}q_{22} + p_{22}q_{11} - 2p_{12}q_{12}. \tag{3.14}$$

An interpretation of the invariant of two conics is given in [15]. It is observed that when there is a projective transformation of coordinates, given two conics P and Q, the terms $P^{-1}Q$ and $Q^{-1}P$ go through a similarity transformation, and therefore, their eigenvalues are preserved.

The *wavelet based conic*, which uses wavelet coefficients $W_i x(t)$ and $W_i y(t)$ has been introduced, where i can be any one of the scales $\{a, b, c\}$:

$$\begin{bmatrix} W_i x(t) & W_i y(t) \end{bmatrix} \Gamma(t) \begin{bmatrix} W_i x(t) \\ W_i y(t) \end{bmatrix} = 1, \tag{3.15}$$

where

$$\Gamma(t) = \begin{bmatrix} \gamma_{11}(t) & \gamma_{12}(t) \\ \gamma_{12}(t) & \gamma_{22}(t) \end{bmatrix} \tag{3.16}$$

is a symmetric matrix with time varying coefficients. Since the wavelet coefficients are translation independent, which is due to the fact that they are zero for a constant function, this form of the conic that does not admit any translation is very well suited for wavelet analysis.

A second wavelet based conic using the matrix

$$\begin{bmatrix} \gamma_{33}(t) & \gamma_{34}(t) \\ \gamma_{34}(t) & \gamma_{44}(t) \end{bmatrix} \tag{3.17}$$

can be defined for three different scales $\{d, e, f\}$. Similar to Equation 3.14, the invariant of two wavelet based conics can be defined as [2]:

$$\eta(t) = \gamma_{11}(t)\gamma_{44}(t) + \gamma_{22}(t)\gamma_{33}(t) - 2\gamma_{12}(t)\gamma_{34}(t). \tag{3.18}$$

The affine invariant function that uses six scales is formed by solving for the elements of the matrices of the wavelet based conics. Since each matrix has three unknown

elements, three scales are necessary to find the unknowns. This is the reason why one of the conics is defined for scales $\{a, b, c\}$ and the other for $\{d, e, f\}$. If the first conic is defined for scales $\{a, b, c\}$, then the solution to a set of linear equations, which are produced by multiplying out Equation 3.6 to form second order polynomials, will give the unknown elements:

$$\gamma_{11}(t)W_a^2x(t) + 2\gamma_{12}(t)W_ax(t)W_ay(t) + \gamma_{22}(t)W_a^2y(t) = 1 \tag{3.19}$$

$$\gamma_{11}(t)W_b^2x(t) + 2\gamma_{12}(t)W_bx(t)W_by(t) + \gamma_{22}(t)W_b^2y(t) = 1 \tag{3.20}$$

$$\gamma_{11}(t)W_c^2x(t) + 2\gamma_{12}(t)W_cx(t)W_cy(t) + \gamma_{22}(t)W_c^2y(t) = 1. \tag{3.21}$$

In order to be able to use Cramer's Rule for a set of linear equations, the set of equations has to become a matrix expression:

$$\begin{bmatrix} W_a^2x(t) & 2W_ax(t)W_ay(t) & W_a^2y(t) \\ W_b^2x(t) & 2W_bx(t)W_by(t) & W_b^2y(t) \\ W_c^2x(t) & 2W_cx(t)W_cy(t) & W_c^2y(t) \end{bmatrix} \begin{bmatrix} \gamma_{11}(t) \\ \gamma_{12}(t) \\ \gamma_{22}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.22}$$

Applying Cramer rule gives:

$$\gamma_{11}(t) = \begin{vmatrix} 1 & 2W_ax(t)W_ay(t) & W_a^2y(t) \\ 1 & 2W_bx(t)W_by(t) & W_b^2y(t) \\ 1 & 2W_cx(t)W_cy(t) & W_c^2y(t) \end{vmatrix}, \tag{3.23}$$

$$\gamma_{12}(t) = \begin{vmatrix} W_a^2x(t) & 1 & W_a^2y(t) \\ W_b^2x(t) & 1 & W_b^2y(t) \\ W_c^2x(t) & 1 & W_c^2y(t) \end{vmatrix}, \tag{3.24}$$

$$\gamma_{22}(t) = \begin{vmatrix} W_a^2x(t) & 2W_ax(t)W_ay(t) & 1 \\ W_a^2x(t) & 2W_bx(t)W_by(t) & 1 \\ W_a^2x(t) & 2W_cx(t)W_cy(t) & 1 \end{vmatrix}. \tag{3.25}$$

When Cramer rule is applied, it is required that the matrix elements found in Equations 3.23, 3.24, and 3.25 are divided by the term

$$\begin{vmatrix} W_a^2x(t) & 2W_ax(t)W_ay(t) & W_a^2y(t) \\ W_b^2x(t) & 2W_bx(t)W_by(t) & W_b^2y(t) \\ W_c^2x(t) & 2W_cx(t)W_cy(t) & W_c^2y(t) \end{vmatrix}. \tag{3.26}$$

The coefficients of the second conic have to be divided with a similar term

$$\begin{vmatrix} W_d^2x(t) & 2W_dx(t)W_dy(t) & W_d^2y(t) \\ W_e^2x(t) & 2W_ex(t)W_ey(t) & W_e^2y(t) \\ W_f^2x(t) & 2W_fx(t)W_fy(t) & W_f^2y(t) \end{vmatrix}. \tag{3.27}$$

However, since $\eta_1(t)$, $\eta_2(t)$, and $\eta_3(t)$ will have to be divided by the product of these two terms, avoiding this altogether will not make a difference.

When all of the time-varying matrix elements are found and substituted at their relevant places, the affine invariant function, which is composed of six wavelet scales $\{a, b, c, d, e, f\}$ is determined:

$$\begin{aligned} \eta_{a,b,c,d,e,f}(t) &= \begin{vmatrix} 1 & 2W_ax(t)W_ay(t) & W_a^2y(t) \\ 1 & 2W_bx(t)W_by(t) & W_b^2y(t) \\ 1 & 2W_cx(t)W_cy(t) & W_c^2y(t) \end{vmatrix} \begin{vmatrix} W_d^2x(t) & 2W_dx(t)W_dy(t) & 1 \\ W_e^2x(t) & 2W_ex(t)W_ey(t) & 1 \\ W_f^2x(t) & 2W_fx(t)W_fy(t) & 1 \end{vmatrix} \\ &+ \begin{vmatrix} W_a^2x(t) & 2W_ax(t)W_ay(t) & 1 \\ W_b^2x(t) & 2W_bx(t)W_by(t) & 1 \\ W_c^2x(t) & 2W_cx(t)W_cy(t) & 1 \end{vmatrix} \begin{vmatrix} 1 & 2W_dx(t)W_dy(t) & W_d^2y(t) \\ 1 & 2W_ex(t)W_ey(t) & W_e^2y(t) \\ 1 & 2W_fx(t)W_fy(t) & W_f^2y(t) \end{vmatrix} \\ &- 2 \begin{vmatrix} W_a^2x(t) & 1 & W_a^2y(t) \\ W_b^2x(t) & 1 & W_b^2y(t) \\ W_c^2x(t) & 1 & W_c^2y(t) \end{vmatrix} \begin{vmatrix} W_d^2x(t) & 1 & W_d^2y(t) \\ W_e^2x(t) & 1 & W_e^2y(t) \\ W_f^2x(t) & 1 & W_f^2y(t) \end{vmatrix} \\ &= \eta_1(t) + \eta_2(t) - 2\eta_3(t). \end{aligned} \tag{3.28}$$

Using the definition of the relative invariant function $f_{i,j}(t)$ it can be proven that:

$$\begin{aligned}
\eta_{a,b,c,d,e,f}(t) &= 2[f_{b,c}(t)f_{e,f}(t)(f_{b,c}(t)f_{e,f}(t)+2f_{c,e}(t)f_{b,f}(t)) \\
&+ f_{b,c}(t)f_{f,d}(t)(f_{b,c}(t)f_{d,f}(t)+2f_{c,d}(t)f_{b,f}(t)) \\
&+ f_{b,c}(t)f_{d,e}(t)(f_{b,c}(t)f_{d,e}(t)+2f_{c,d}(t)f_{b,e}(t)) \\
&+ f_{a,c}(t)f_{f,e}(t)(f_{a,c}(t)f_{e,f}(t)+2f_{c,e}(t)f_{a,f}(t)) \\
&+ f_{a,c}(t)f_{d,f}(t)(f_{a,c}(t)f_{d,f}(t)+2f_{c,d}(t)f_{a,f}(t)) \\
&+ f_{a,c}(t)f_{e,d}(t)(f_{a,c}(t)f_{d,e}(t)+2f_{c,d}(t)f_{a,e}(t)) \\
&+ f_{a,b}(t)f_{e,f}(t)(f_{a,b}(t)f_{e,f}(t)+2f_{b,e}(t)f_{a,f}(t)) \\
&+ f_{a,b}(t)f_{f,d}(t)(f_{a,b}(t)f_{d,f}(t)+2f_{b,d}(t)f_{a,f}(t)) \\
&+ f_{a,b}(t)f_{d,e}(t)(f_{a,b}(t)f_{d,e}(t)+2f_{b,d}(t)f_{a,e}(t))]. \qquad (3.29)
\end{aligned}$$

The function calculated from the affine transformed boundary is:

$$\begin{aligned}
\tilde{\eta}_{a,b,c,d,e,f}(t) &= \tilde{\eta}_1(t)+\tilde{\eta}_2(t)-2\tilde{\eta}_3(t) \\
&= (\det A)^4 \eta_{a,b,c,d,e,f}(t). \qquad (3.30)
\end{aligned}$$

This fact can be seen more clearly by noticing that each of the functions $f_{i,j}(t)$ are related to their affine transformed counterparts with $\tilde{f}_{i,j}(t) = (\det A) f_{i,j}$. Equation 3.30 proves that $\eta_{a,b,c,d,e,f}(t)$ is an affine invariant function. This function, however, is a relative invariant and, hence, it has to be normalized so that the term $(\det A)^4$ is eliminated. If, for instance, the boundary functions are of length, 2^{12}, they can be decomposed into 12 scales $\{1,\ldots,12\}$. Then, the normalizing factor of the relative invariant function composed of wavelet coefficients at scales $\{1,2,3,4,5,6\}$ can be another relative invariant, which is made up of wavelet coefficients at scales $\{7,8,9,10,11,12\}$, since:

$$\frac{\tilde{\eta}_{1,2,3,4,5,6}(t)}{\tilde{\eta}_{7,8,9,10,11,12}} = \frac{\eta_{1,2,3,4,5,6}(t)}{\eta_{7,8,9,10,11,12}}. \qquad (3.31)$$

For their experiments, however, the authors in [2] have chosen to normalize the boundary functions to a length 2^{14} so that 14 scales of decomposition is performed, which

makes it possible that the finest scales of $\{1, 2\}$ can be avoided and the coarser scales $\{3, \ldots, 14\}$ can be used.

The invariance of the function $\eta_{a,b,c,d,e,f}(t)$ is demonstrated with Figure 3.2, where the invariant functions obtained from the original and affine curves are on the left and right, respectively.

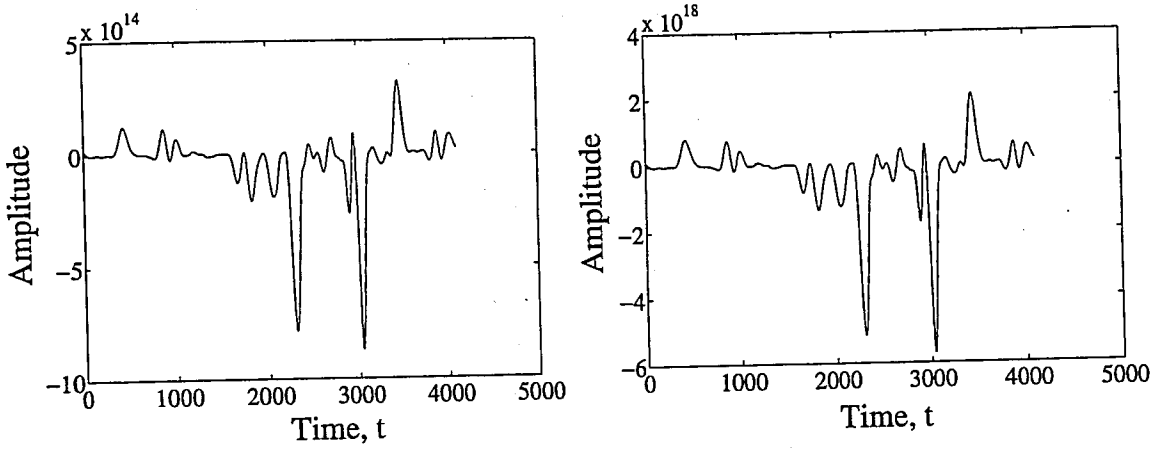

Figure 3.3. The relative invariant functions $\eta_{7,8,9,10,11,12}(t)$ and $\tilde{\eta}_{7,8,9,10,11,12}(t)$

3.2.1. Choice of Wavelet Scales

The proposed invariant function $\eta_{a,b,c,d,e,f}(t)$ is a relative invariant one and to make it an absolute invariant, it is being normalized with another relative invariant function. When doing this normalization, the scales of the normalizing relative invariant function may either be nonoverlapping with the first one, or some overlapping of scales may be allowed. If the scales of the two functions are not allowed to overlap, a total of 12 scales will be necessary and the boundary functions will have to be of length 2^{12}. Using overlapping scales, the absolute invariant function may become:

$$\frac{\eta_{1,2,3,4,5,6}(t)}{\eta_{2,3,4,5,6,7}(t)}. \tag{3.32}$$

Such a function requires the boundary to be of length 2^7. Thus, the number of scales used may vary between 7 and 12. When choosing the number of scales, a trade-off

between recognition time and accuracy has to be made, since when the boundary is of a shorter length, some of the details may be lost, but the recognition time will be much less, whereas when the boundary is longer, the noisy finer scales may be avoided, which will increase both the computation time and the recognition rate for noisy images.

3.3. The Redundant Wavelet Transform

The calculation of the affine invariant function $\eta_{a,b,c,d,e,f}(t)$ involves multiplication of wavelet coefficients at various scales, so one may pause and ask how this is possible if the number of wavelet coefficients at different scales are not the same, which is a consequence of decimation. The answer is a redundant wavelet transform, which is a modification of the well-known orthogonal discrete wavelet transform.

A redundant form of the wavelet transform was necessary because the discrete wavelet transform is not shift invariant, which means that there is no simple relationship between the wavelet coefficients of an original signal and a shifted version of it [9]. The first and most obvious way to find a shift invariant transform is to compute the wavelet transform of all possible shifts of a signal and average them. However, this is computationally very inefficient.

During the computation of the discrete wavelet transform, at each decimation stage, one may choose to keep either the odd indexed terms or the even indexed terms although conventionally the even indexed terms are kept. If the odd indexed terms had been chosen, a totally different set of wavelet coefficients would have been produced. The set of coefficients produced in such a case corresponds, however, precisely to circularly shifting the signal by an odd number of terms. Moreover, at each decimation stage, one can keep the odd indexed samples of the signal or the evens independent of the choice at the previous stage. With such a freedom, many different wavelet transforms are possible.

The stationary wavelet transform proposed in [16] contains the whole set of coefficients of discrete wavelet transform, where the coefficients are produced from all

possible combinations of choosing odd or even indexed samples during decimation at each stage. This scheme is exactly the same as the one proposed by Mallat in [17], which was actually an attempt to approximate the continuous time wavelet transform.

The idea behind the scheme is quite simple. Instead of decimating the low and high pass filtered scaling coefficients to produce the next lower scale scaling and wavelet coefficients, respectively, the decimation is omitted and the low and high pass filter coefficients are upsampled by padding them with zeros [16, 17]. In such a case, both the number of scaling and wavelet coefficients are the same as the number of original signal samples. However, the transformation is no longer orthogonal and the total number of coefficients keep increasing as the number of scales is increased.

If initially the low and high pass filter coefficients are $h(n)$ and $h_1(n)$, respectively, the low and high pass filter coefficients at a scale j lower than the initial finest scale are computed by inserting $2^j - 1$ zeros between the initial coefficients of the relevant filters.

The shift invariance of the redundant wavelet transform (RDWT) is demonstrated with Figure 3.4, where the first plot shows the signal, whose redundant wavelet transform is taken, and the second and third are the redundant wavelet transforms of the signal and a shifted version of the signal by 50 samples, respectively, at the scale three lower than the finest scale. The Daubechies wavelet of length 4 has been used to produce the plots.

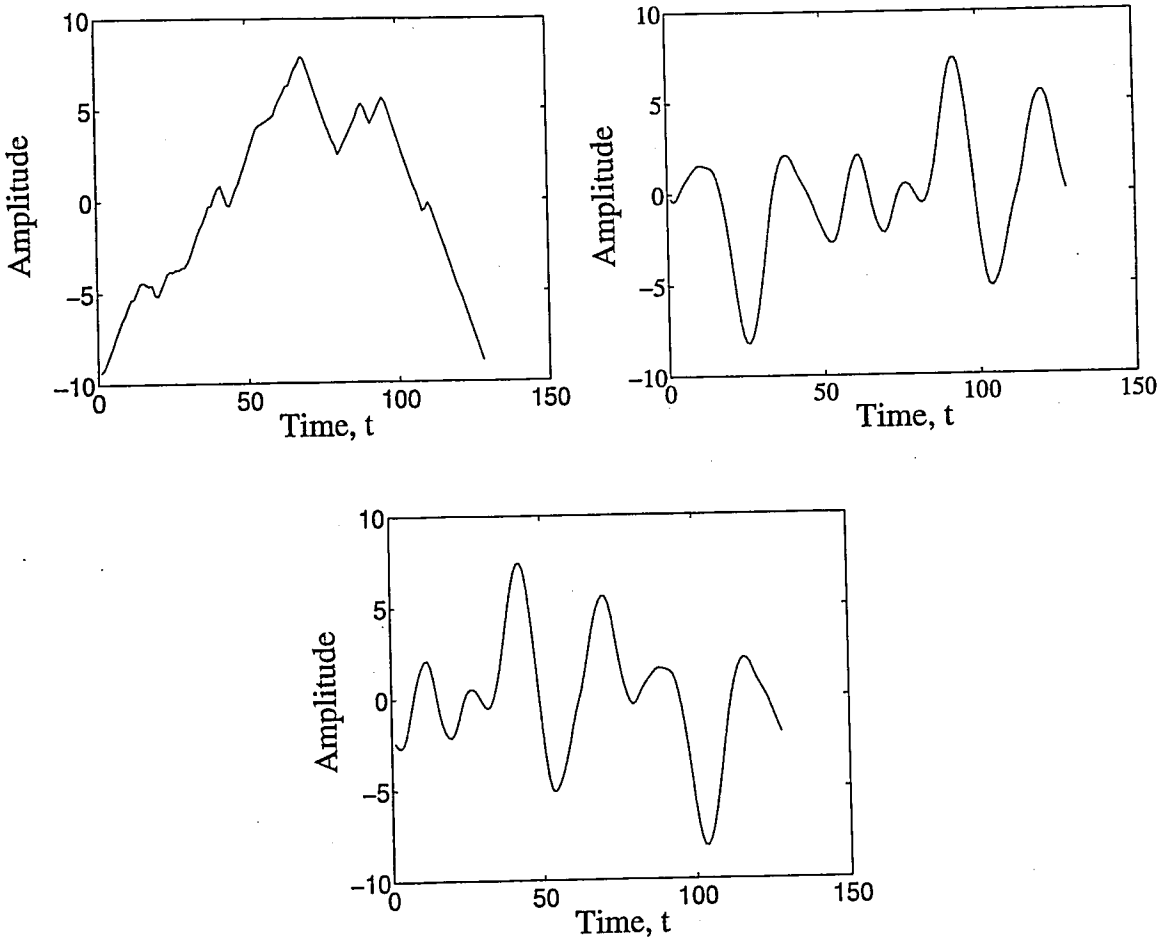

Figure 3.4. A signal, RDWT of the signal, and RDWT of the shifted signal

4. AMELIORATION OF THE AFFINE INVARIANT WAVELET FUNCTION

Although the affine invariant function proposed in [2] will theoretically have high discriminating power because it uses 12 wavelet scales, when it comes to practice, it does not produce very good results. Therefore, a number of alterations and additions to what the authors in [2] have done is necessary. For this purpose, the question of why the theory does not work in practice has to be answered.

4.1. Obtaining Point Correspondence

In practice, what a recognition system will probably do is, keep only the set of reference affine invariant functions in a database, and match that of the unknown object with those functions by means of correlation. The boundaries of the objects in the reference images are extracted in order to calculate the affine invariant functions, and later the boundary of the unknown object is found. During and after all of these processes, there is no need to establish point correspondence between the boundaries of the reference objects and the unknown object according to [2], since it is never been mentioned in the paper.

Using redundant wavelet transform, which is also shift invariant, does not guarantee that from whichever point the boundary of the unknown object starts and wherever the rest of the points are placed, an accurate object recognition can be done. When assigning the biggest correlation between all possible shifts of the invariant function of the unknown object and the invariant function of a reference object to that particular reference object, the assumption is that there is only a shift in the points of the unknown boundary. Shift invariance implies that when a signal is shifted, its wavelet transform will also shift, but the samples of the signal should be the same in the two cases although shifted. Since the boundary is sampled, there is no guarantee that even under rotation, the two boundaries will have point correspondence.

The wavelet transform of a signal is computed by assuming that the samples of the signal are equally spaced. When the boundary of an object is sampled, however, the points are not equally spaced. This fact is ignored when Equation 3.3 is used. Therefore, even when the identity of an object is known, to be able to correctly match its image with the reference image in the database obtaining a high correlation value, the points on the boundary of the object in the affine image, which correspond to the affine transforms of the points on the boundary of the object in the reference image should be found.

Interpreting the affine invariant function $\eta_{a,b,c,d,e,f}(t)$ as the invariant of two conics with time-varying coefficients makes the demand for point correspondence clear. Time-varying coefficients are calculated by fitting conics to three points at a time, which are actually the wavelet coefficients of a single point in time domain at three distinct scales. Therefore, the points having the same time index should be affine images of each other for a high correlation value between the two affine functions. Otherwise, the fitted conics will have different coefficients making the value of the affine invariant function at that time index, which is calculated from the coefficients of conics, different as well.

Registering two affine images, images related with an affine transformation, is a difficult task by itself. A simple procedure, however, that will improve the performance of the recognition system can be suggested by using the area parametrization in [5].

4.1.1. Area Parametrization

It has been mentioned above that the wavelet transform assumes the samples of the signal are obtained by uniform sampling. For the boundary of an object, which is two dimensional, this is equivalent to parameterizing the function with arc length, where the parameter is incremented in uniform amounts for consecutive samples. On the other hand, it is known that under affine transformation arc length is not linearly transformed [5], meaning the points on the affine curve, which are affine images of the points on the original curve will not have parameter values that are related to the parameter values of the points on the original curve by simple scaling and translation

but a more complicated unknown relationship. Thus, a new parametrization has been introduced such that it has the following properties [18]:

- It is derivable from the curve, independent of the initial representation of the curve.
- It must be a monotonic function of the arc length.
- It must be linear under an affine transformation.

The arc length has the first and second of these properties, but not the third due to the shear effect in affine transformation, which prevents length from being preserved. Therefore, it is not a valid parametrization.

Two candidate parameterizations satisfy these criteria: The first is the "affine length"

$$t = \int_C \sqrt[3]{\det(\frac{d\vec{X}}{d\xi}, \frac{d^2\vec{X}}{d\xi^2})}\, d\xi = \int_C \sqrt[3]{x_\xi y_{\xi\xi} - y_\xi x_{\xi\xi}}\, d\xi \tag{4.1}$$

where $\vec{X}$ is the coordinate function vector

$$\vec{X} = \begin{bmatrix} x(\xi) \\ y(\xi) \end{bmatrix}, \tag{4.2}$$

x_ξ, y_ξ are the first derivatives and $x_{\xi\xi}$, $y_{\xi\xi}$ are the second derivatives of the component functions $x(\xi)$ and $y(\xi)$, which have an arbitrary parametrization ξ, and C is the path along the curve. The affine length is problematic, since it involves second derivatives, which are very susceptible to noise. Instead, a first order form, the area parametrization has been developed [18]:

$$\begin{aligned} t &= \frac{1}{2}\int_C \left|\det(\vec{X}(\xi), \vec{X}_\xi)\right|\, d\xi \\ &= \frac{1}{2}\int_C |x(\xi)y_\xi - y(\xi)x_\xi|\, d\xi. \end{aligned} \tag{4.3}$$

This parametrization will not be a valid parametrization, either, when $B \neq 0$ in Equation 3.3. The influence of B may be avoided if the origin of the coordinate system is moved to the area center (AC), which is defined as:

$$\vec{X}_{AC} = \frac{1}{2} \frac{\int_C \vec{X}(\xi) \det(\vec{X}(\xi), \vec{X}_\xi)\, d\xi}{\int_C \det(\vec{X}(\xi), \vec{X}_\xi)\, d\xi}. \tag{4.4}$$

The area center of the affine curve is the affine transform of the area center of the original curve, which has been proven in [7]. Intuitively, this holds because the affine transformation transforms areas with a constant scale, $\det(A)$. The total areas enclosed by the original and affine transformed objects, T_0 and T, are related by

$$T = |\det A|\, T_0 \tag{4.5}$$

The integrand in Equation 4.3 is twice the area covered by a line from AC to the curve while it is moved by the amount of $d\xi$ along the curve. Therefore, the well known fact that under an affine transformation area ratios stay the same is used here. The determinant is zero at all points where the "tracing line" and the tangent to the curve at that point is collinear [18].

The affine area parametrization, which has initially been developed for affine Fourier descriptors in [7] to fulfill the requirement that Fourier transformation is linear only when the functions, whose Fourier transforms are taken are linearly parameterized under affine transformation, is being used in this study to obtain point correspondence between two curves, which are related by an affine transformation. This is accomplished by a procedure, which is outlined below:

- Starting points for the original and affine curves, which are affine images of each other are found, each of which is assigned a parameter value of 0. This is an essential step, since the area parameter values will shift if the starting points are not found correctly.
- The area parameter values of the curve, which is the boundary of the object in the

reference image that is going to stay in the database are calculated, beginning with the point found before. The area parameter values are normalized by dividing them with the total area the object occupies so that the total area becomes 1.

- The affine curve, which is the boundary of the unknown object, is resampled at a higher rate to obtain more samples, some of whose normalized area parameter values may match those of the original curve. Among the samples of the resampled affine curve, the one whose normalized area parameter value is the same as one of the area parameter values of the original curve is selected as the image of the point to whom this value belongs. Doing this consecutively for each point on the original curve, affine images of all of the points are found.

In this procedure, instead of making the area parameterizations of both of the curves uniform, one of the curves is preserved as it is, and the second one is altered. By this way, both the computation time is decreased, and the drawback of this procedure, which is the fact that the resampling will probably not be perfect, is limited.

The success of such a procedure is strongly correlated with whether the starting points are found accurately. Feature matching is an active area of research and the algorithms suggested in the literature are computationally very exhausting such as [19]. Therefore, an alternative, which works only in the case of airplane silhouettes has been suggested.

4.1.1.1. Starting Point Finding. The basic idea behind this procedure is to find the nose of the airplane. The reason is simple: All of the airplanes possess a nose and, therefore, when the unknown object is being matched with any one of the objects in the database, the rest of the point finding algorithm may go on. Due to the shape of airplanes, the noses of all airplanes lie along the major axis in their canonical (top) view. To restore the object along its major axis, Hotelling transformation is used, which may make it possible to find the nose, since it is the farthest point of the boundary along the major axis. However, some problems are also present:

- First of all, Hotelling transformation in practice is used when the object undergoes a Euclidean transformation, which is rotation, and translation complemented with scaling. When the shear effect of affine transformation is included, the major axis may not align exactly with the line connecting the nose and tail of the airplane. The nose still may be chosen as the one that has the biggest x-coordinate magnitude. The word magnitude underlines the fact that the transformation matrix Hotelling transformation produces may place either the nose or the tail of the airplane on the positive portion of the x-axis. When the shear is taken into account, the nose may be the point that has the biggest or the smallest x-coordinate. These are illustrated with Figure 4.1 below, where the airplane on the left is aligned along its major axis as seen on the right.
- Some of the shear effects make the line along the wings, the major axis. In such cases, the wings are along the x-axis. This fact should also be taken into account when designing a starting point finding algorithm for an airplane silhouette.

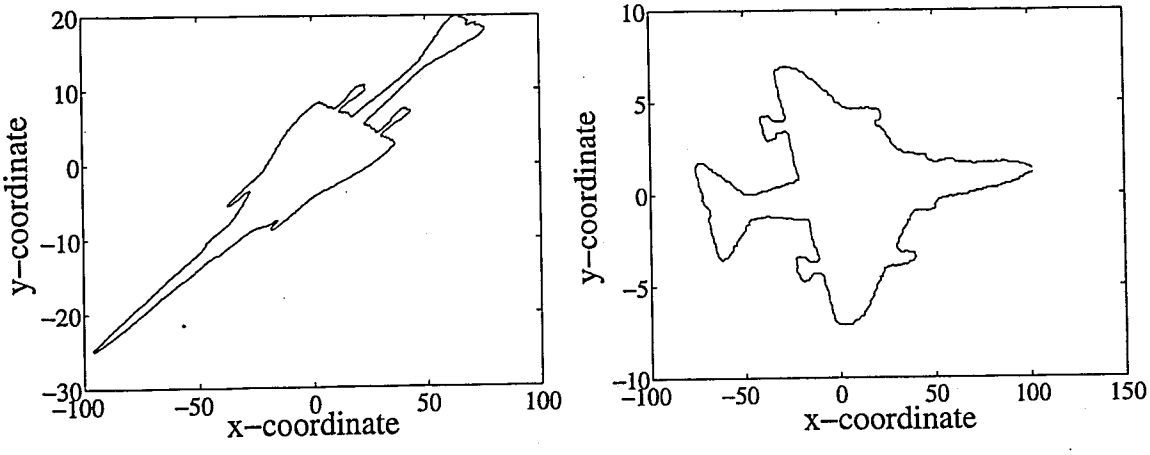

Figure 4.1. An airplane boundary, the boundary aligned along its major axis

In its final form, the procedure depends on the facts:

- The centroid of the airplane is closer to the nose than to its tail.
- The wings are at approximately same distances from the centroid.

To check whether the wings or the nose and tail are along the major axis, the four distances from the centroid to all of the extrema should be found. These are:

- Centroid-to-Point with the biggest x value (1)
- Centroid-to-Point with the smallest x value (2)
- Centroid-to-Point with the biggest y value (3)
- Centroid-to-Point with the smallest y value (4)

The bigger of the ratio of distances $max(1,2)$-to-$min(1,2)$ and $max(3,4)$-to-$min(3,4)$ will tell whether the nose is along the x or y direction. When the direction is found, the point that is more distant to the centroid is selected as the nose of the airplane.

The starting point finding algorithm, although airplane dependent, works in almost all cases correctly. In some cases, it may fail because it is based on distances, which are lengths, and lengths are not preserved under affine transformation.

4.2. The Chosen Wavelet Scales

The relative affine invariant function $\eta_{a,b,c,d,e,f}(t)$ uses six different scales, and to make it an absolute invariant another six scales are needed, which means that a total of 12 scales are being employed at the same time as in Equation 3.31. To avoid using the finer scales, which may contain a lot of noise, for the simulations in [2], the following invariant function, which allows some overlapping of scales, has been used:

$$I(t) = \frac{\eta_{3,4,5,6,7,8}(t)}{\eta_{7,8,9,10,11,12}(t)}. \tag{4.6}$$

For a decomposition of the signal to 12 scales, it should be of length a 2^{12}, hence a resampling of the signal will be necessary. When the redundant wavelet transform of the signal is calculated, and the relative affine invariant functions, $\eta_{3,4,5,6,7,8}(t)$ and $\eta_{7,8,9,10,11,12}(t)$ are plotted, Figure 4.2 is obtained. One may say that the function $\eta_{3,4,5,6,7,8}(t)$, which is composed of finer scales is a high frequency signal, whereas the function $\eta_{7,8,9,10,11,12}(t)$ varies more smoothly. Moreover, the maximum amplitude of the former is less than 10 while the latter one has maxima well over 10^{14}. Since bigger magnitude means higher energy, the information carried by $\eta_{3,4,5,6,7,8}(t)$ is quite negligible and using it as the invariant function will result in a very poor performance.

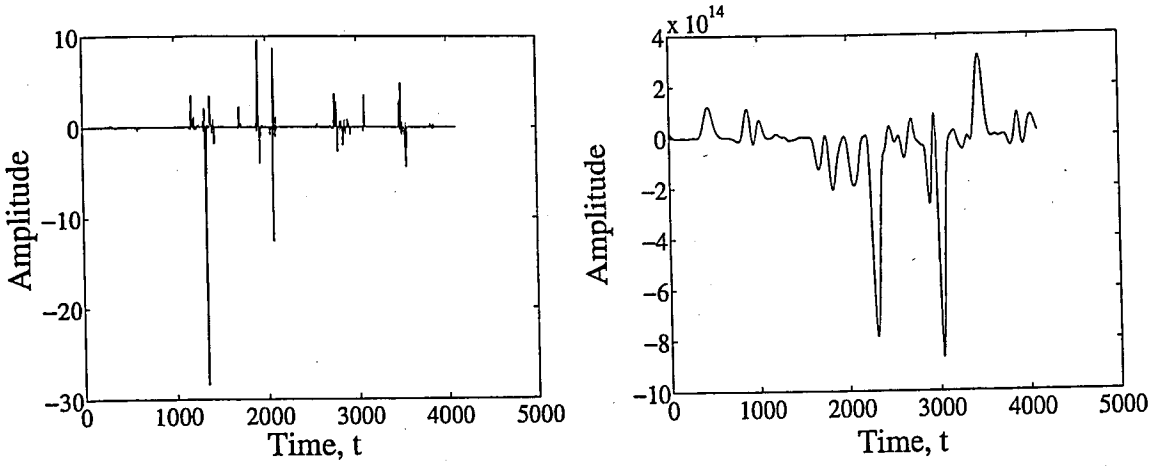

Figure 4.2. Relative invariant functions, $\eta_{3,4,5,6,7,8}(t)$ and $\eta_{7,8,9,10,11,12}(t)$

Displaying a concatenation of the wavelet coefficients of the x-coordinate function of the original curve at various scales $\{1, \ldots, 12\}$, the difference between the finer scale coefficients and the coarser ones is shown.

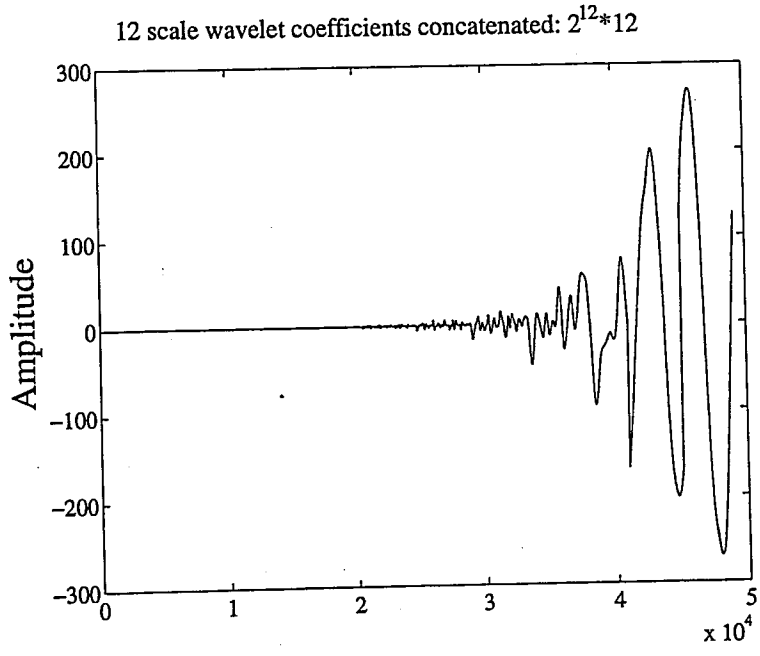

Figure 4.3. Wavelet coefficients at scales $\{1, \ldots, 12\}$

Therefore, when the finer scales are selected to be used in the formation of the affine invariant function, it is very unlikely that a correct recognition will be done due to the errors that occur during sampling, and when obtaining point correspondence.

Using another relative affine invariant function to make the relative affine invariant function $\eta_{a,b,c,d,e,f}(t)$ absolute is not very appropriate in this case. Another term, which is also relatively invariant under affine transformation may be used for normalization. One candidate for this is the total area an object occupies, since this area in the reference image, T_0 is transformed as $T = \det A\ T_0$, where T is the object area in the new image. Therefore, cancelling the $(\det A)^4$ in Equation 3.30 can be done via:

$$\frac{\tilde{\eta}_{a,b,c,d,e,f}(t)}{T^4} = \frac{\eta_{a,b,c,d,e,f}(t)}{T_0^4}. \tag{4.7}$$

Since only six scales are sufficient to produce the invariant function, instead of making the length of the boundary 2^{12}, a length of 2^7 may be chosen. A longer signal will increase both the computation time and the probability of making errors during the point matching step. The reason for not choosing 2^6 is explained with the aid of Figure 4.4, where the first plot shows the wavelet coefficients of the x-coordinate function of the original curve at scales $\{1, \ldots, 7\}$, and the second and third plots are the relative invariant functions $\eta_{1,2,3,4,5,6}(t)$ and $\eta_{2,3,4,5,6,7}(t)$, respectively.

The relative invariant function, which is composed of finer scales, has magnitudes on the order of 10^5, whereas the other one has magnitudes exceeding 10^8. The function, which has more energy is chosen in this case.

The final form of the absolute function is:

$$\frac{\tilde{\eta}_{2,3,4,5,6}(t)}{T} = \frac{\eta_{2,3,4,5,6}(t)}{T_0}. \tag{4.8}$$

Actually, when the matching of two functions is done via correlation analysis, if one of the functions is a scaled version of the other, the correlation will still be 1. If there is a function $f(t)$ and it has to be correlated with $af(t)$, this is done by

$$\frac{f(t) \cdot af(t)}{\sqrt{f^2(t)}\sqrt{a^2 f^2(t)}} = \frac{af^2(t)}{f(t) \cdot af(t)} = 1. \tag{4.9}$$

Therefore, it is not very important whether the function is relative or absolute.

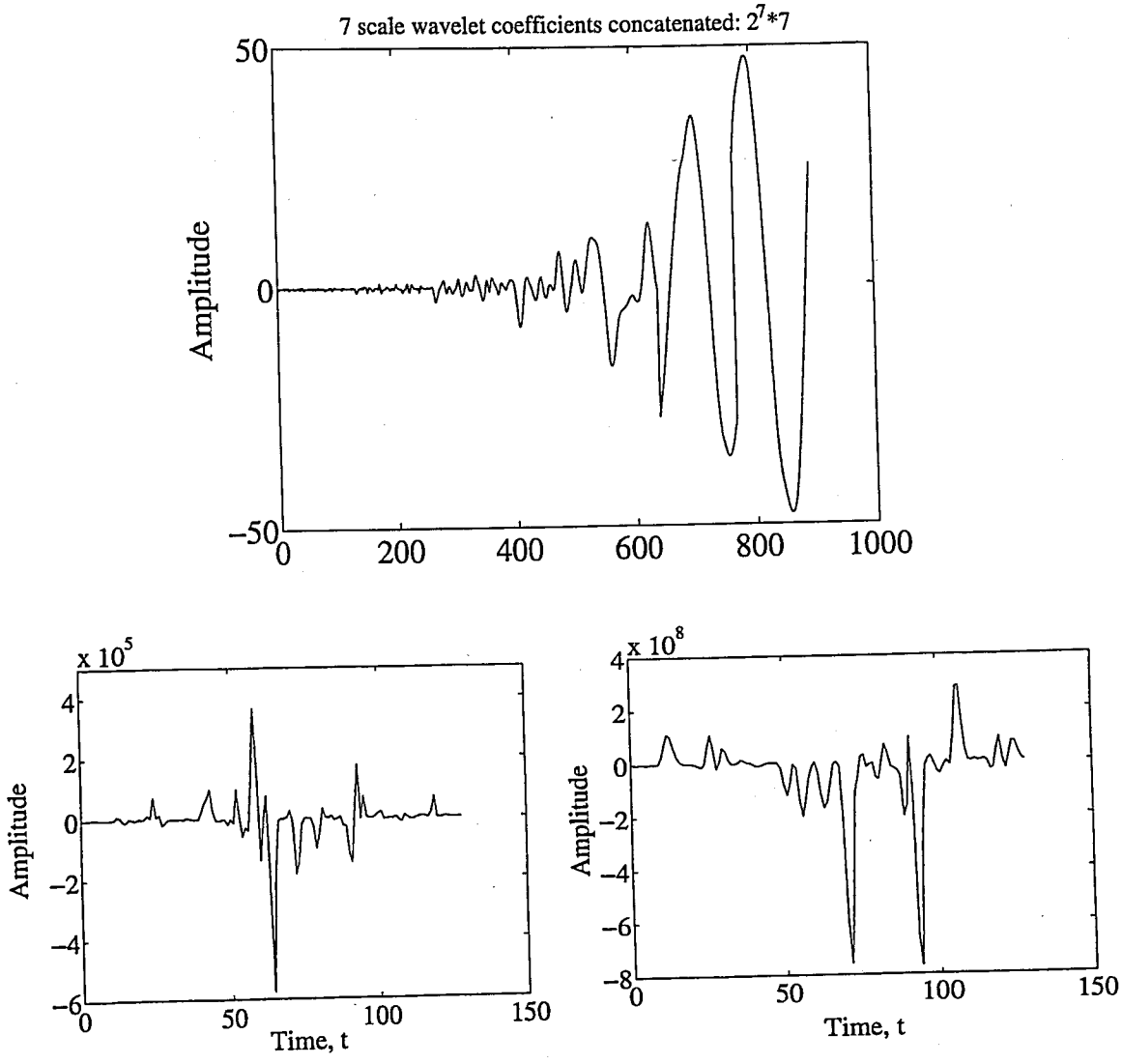

Figure 4.4. Wavelet coefficients at scales $\{1, \ldots, 7\}$, relative invariant functions $\eta_{1,2,3,4,5,6}(t)$ and $\eta_{2,3,4,5,6,7}(t)$

The affine invariant function obtained from the reference image, $I_1(t)$ is correlated with the function obtained from the affine image, $I_2(t)$,

$$\frac{\int I_1(t) I_2(t) dt}{\|I_1(t)\| \, \|I_2(t)\|}, \tag{4.10}$$

which will be the criterion for the selection of the identity of the object: The reference object to which the maximum correlation value is assigned is chosen as the identity of the unknown object.

This completes the explanation of steps and procedures necessary to obtain a more effective recognition system by elaborating on the work in [2]. The functionality of each step is explained with an example. The reference image of an object and an affine transformed version of it are as shown:

Figure 4.5. The reference image of an object and an affine transformed version

When the algorithm in [2] is used as it is, the boundary is resampled so that it has a length of 2^{12} and Figure 4.6, where the affine invariant function found from the affine curve has been shifted so that the correlation value between the two affine invariant functions is maximized, is produced.

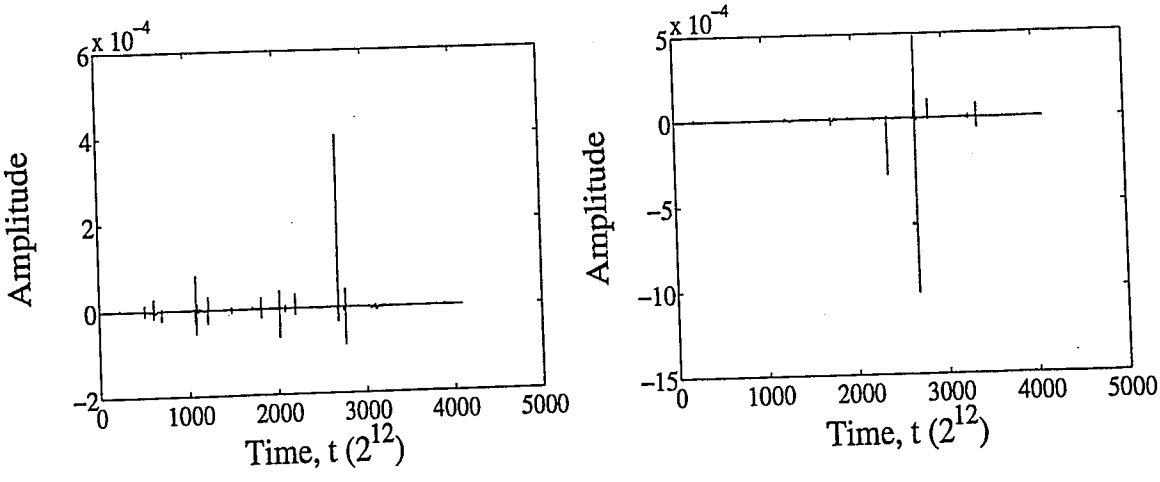

Figure 4.6. The absolute invariant functions from the reference and affine images

The maximum correlation value for these two functions is 0.3457. Looking at the plots, however, it is no wonder why the correlation is so low. When the length of the boundary is dropped to 2^7, but the point correspondence step is avoided, the correlation value is 0.7221. In Figure 4.7, the plot on the left is the affine curve, the

boundary of the affine object, and the one on the right contains the two invariant functions superimposed for the highest correlation value.

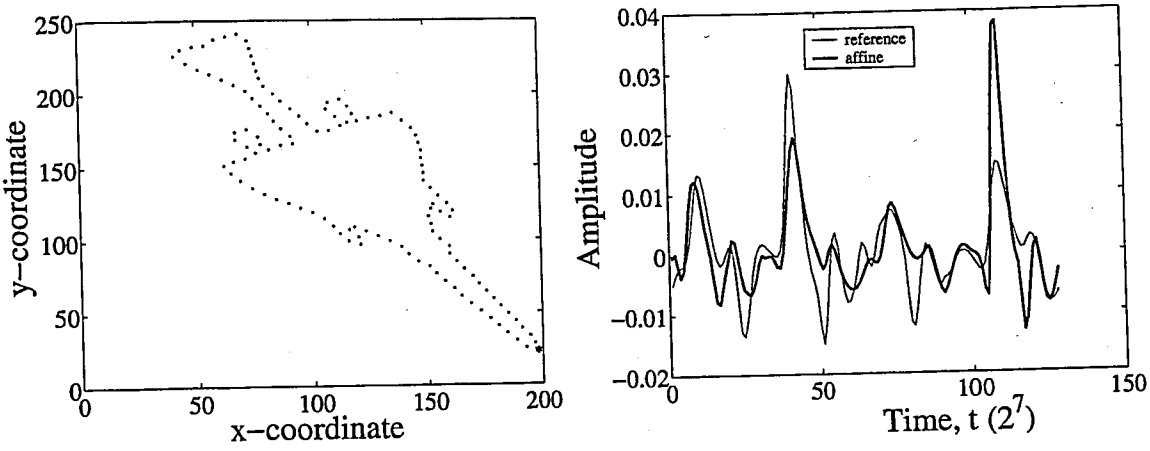

Figure 4.7. The affine curve without any point matching, the invariant functions superimposed

When point matching is done, Figure 4.7 is changed to Figure 4.8 and the correlation value is 0.9779:

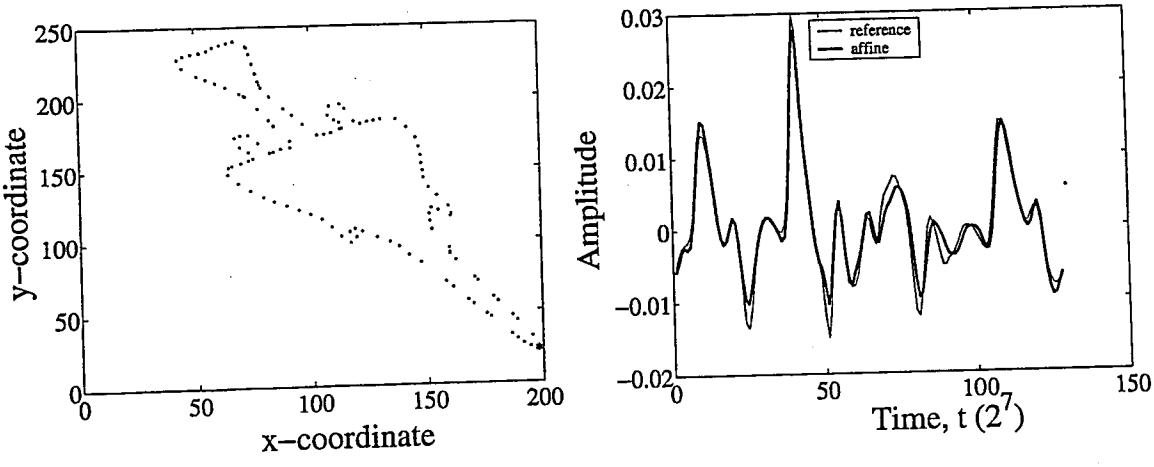

Figure 4.8. The affine curve modified by the point matching step, the invariant functions superimposed

Similar results have been obtained for different objects as well.

5. AFFINE INVARIANT MULTIWAVELET FUNCTION

Wavelets have proven to be a useful tool in many fields including image compression, denoising, and lastly object recognition. Until recently, almost exclusively scalar wavelets have been used. However, some properties of multiwavelets make them very attractive for signal processing applications. For instance, such features as compact support, orthogonality, symmetry, and vanishing moments are known to be important in signal processing [11]. A scalar wavelet cannot posses all of these simultaneously although multiwavelets can. This implies that multiwavelets can provide perfect reconstruction (orthogonality), have a small number of filter coefficients (compact support), good performance at the boundaries (symmetry), and high order of approximation (vanishing moments) at the same time, so they could outperform scalar wavelets. Multiwavelet theory, which has been largely developed with compression properties in mind [12] will be used in an object recognition application for the first time. The affine invariant function proposed in [2] and improved in this study is implemented with multiwavelet transform.

5.1. Choice of Multiwavelet Scales

In order to select the multiwavelet coefficients that are going to be used in the invariant function $\eta_{a,b,c,d,e,f}(t)$, it is important to understand the relationship between the multiwavelet coefficients of the original and affine curves. When the multiwavelet transform of Equation 3.3, which shows how the points on the affine curve are produced from the points on the original curve, is taken,

$$\begin{bmatrix} W_i\tilde{x}_1(t) \\ \vdots \\ W_i\tilde{x}_R(t) \\ \hline W_i\tilde{y}_1(t) \\ \vdots \\ W_i\tilde{y}_R(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} W_ix_1(t) \\ \vdots \\ W_ix_R(t) \\ \hline W_iy_1(t) \\ \vdots \\ W_iy_R(t) \end{bmatrix} \tag{5.1}$$

where i is an arbitrary scale, R, which is mostly selected as 2 in the literature of multiwavelets, is the number of scaling and wavelet functions that span the subspaces V_i and W_i of $L^2(\mathbb{R})$, is found.

Looking at Equation 5.1, it is seen that the multiwavelet coefficients at identical rows, satisfy an equation similar to the scalar wavelet case

$$\begin{bmatrix} W_i\tilde{x}_j(t) \\ W_i\tilde{y}_j(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} W_i x_j(t) \\ W_i y_j(t) \end{bmatrix} \tag{5.2}$$

where i is an arbitrary scale and j can be any one of $\{1, \ldots, R\}$. Therefore, by viewing each element of a multiwavelet as a single wavelet, an affine invariant function for each of these single wavelet function coefficients can be formed following the same procedure for scalar wavelets. After the correlation values of each of the affine invariant functions have been found, either of the maximum, the mean, the median or the minimum of the correlations can be selected as the final correlation value.

The simulations done with these in mind, however, turned out to be very ineffective with very low recognition rates. Affine invariant functions produced with GHM multiwavelets, which have an R of 2, are shown in Figure 5.1.

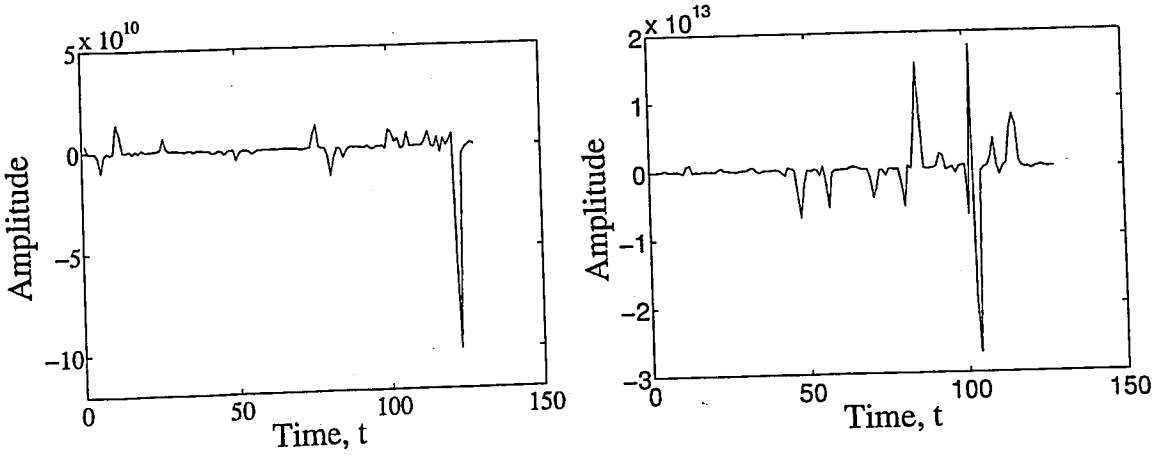

Figure 5.1. Affine invariant functions of the two single wavelets

Since the functions have both high-magnitude sharp peaks and portions with

vanishing magnitude, this suggests that some of the scales may not be very appropriate to use, and hence should be avoided. When the multiwavelet transform of the x-coordinate function of an object is plotted, it is seen that at the finer 4 scales, the coefficients have very low magnitudes and therefore do not posses much energy. This is apparent in Figure 5.2, where the concatenated coefficients at different rows are separately shown. Beginning with the fifth scale, bigger magnitudes are encountered. Since the signal whose redundant multiwavelet transform is taken is of length 2^7, the fifth scale starts with the $128 \times 4 + 1 = 513^{th}$ sample of the concatenated coefficients.

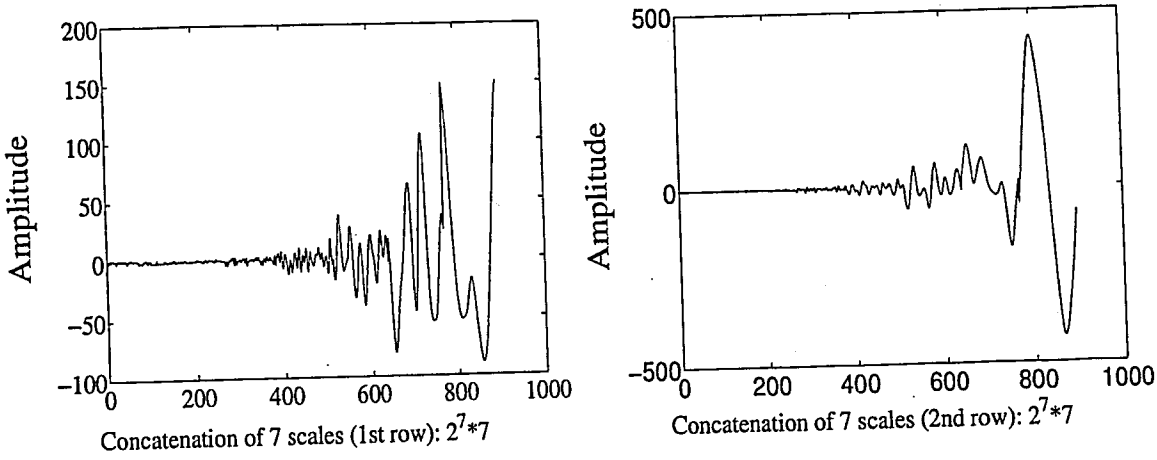

Figure 5.2. Multiwavelet transform coefficients at scales $\{1, \ldots, 7\}$

This suggests using the three coarser scales and forming a single affine invariant function instead of R functions. Since in the design of multiwavelet systems R has almost always been chosen as 2, there are a total of 6 sets of coefficients as required to form a single affine invariant function. When a single invariant function is formed, it will have bigger magnitudes than two separate invariant functions with smoother, wider peaks as can be seen in Figure 5.3.

The final form of the affine invariant multiwavelet function uses the 3 coarsest scales of the multiwavelet transform coefficients. This corresponds to fitting one conic with time varying coefficients to each row of the multiwavelet transform coefficients.

Since calculation of the multiwavelet transform is computationally more complex, it will not be very correct to compare the performances of scalar and multi wavelets

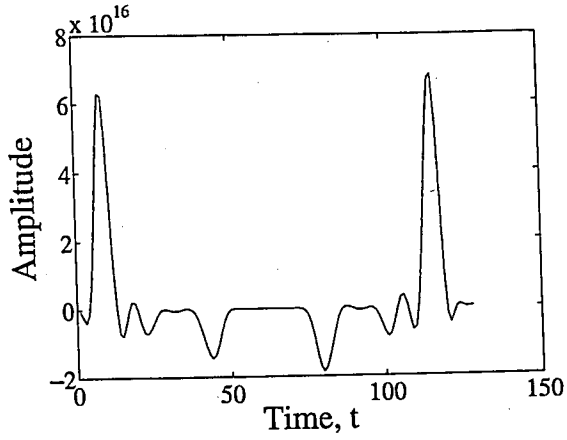

Figure 5.3. Affine invariant function produced by combining multiwavelet coefficients at different rows

in this setting. Therefore, wavelet coefficients from two different wavelets are used to form a single affine invariant function, where the 3 coarsest scales are selected from each wavelet. This function may be expected to outperform the function formed from the coefficients of a single wavelet.

When the scales to be used in the affine invariant function are chosen, the ones that are less susceptible to noise and have bigger energy are preferred, since otherwise any scale will have an equal chance due to the fact that the wavelet coefficients of two different objects cannot be the same for all scales.

5.2. Redundant Multiwavelet Transform

During the implementation of orthogonal multiwavelet transform, the decimation, which is done after filtering the multiscaling coefficient vector at a finer scale will reduce the length of the coefficients at coarser scales. Since calculation of the affine invariant multiwavelet function requires the multiplication of coefficients at different scales, however, their length should be the same, which can only be accomplished with a redundant representation. This turns out to be one of the biggest problems encountered during the making of the thesis, for there is yet no application of a redundant multiwavelet transform in the literature.

A translation invariant multiwavelet transform has been introduced in [20], which is an attempt to improve the results obtained in a previous article [21] on image denoising that was based on a redundant form of the wavelet transform. The latter article, forms a translation invariant representation by averaging over all possible shifts of a signal, which is an idea introduced by Coifman and named by him as "Cycle Spinning" by him [22]. The translation invariant representation, which is obtained by averaging over all possible shifts shows that the autocorrelation functions of wavelet and scaling functions satisfy refinement equations, where the filter coefficients turn out to be the famous "a-trous" filters [22].

The *algorithme a-trous* has been originally devised as a computationally efficient algorithm, which is viewed as a nonorthogonal multiresolution algorithm with no decimation. The filters are called a-trous, meaning with holes because the even coefficients of the relevant filter with the exception of the center are zero. It has been shown that the Lagrange a-trous filters are in one to one correspondence with the Daubechies filters for orthonormal wavelets of compact support [23]. They are actually convolutional squares of Daubechies filters. A completely different derivation and point of view, which is based on averaging the discrete wavelet transforms of shifted versions of the same signal, arrive at the a-trous filters. The algorithme a-trous is said to be a good approximation of the continuous time wavelet transform [23].

When the same averaging approach is used for multiwavelets, the autocorrelation and cross-correlation functions of the multiscaling and multiwavelet coefficients satisfy a refinement equation themselves. The filters are found as the convolution of the orthogonal filters with themselves. Therefore, when the orthogonal filters are designed for $R = 2$, the coefficient matrices in the new system will be 4×4. The computational complexity will increase with increasing R and it will be harder to choose the scales to be used in the affine invariant function.

The formulation of a redundant multiwavelet transform, which is also shift invariant, can also be done by following Mallat's approach by avoiding decimation and using the designed coefficients as they are. This is a simple idea, but it works fine.

In the scalar wavelet case when the decimation step is avoided and the filter coefficients are upsampled instead, a redundant wavelet transform is obtained. Filtering a signal $c_{i+1}(n)$ with $h(-n)$ means convolving it so that

$$c_i(n) = \sum_k h(k-n)c_{i+1}(n), \tag{5.3}$$

where i is an arbitrary scale. In matrix form, this equation becomes

$$\begin{bmatrix} c_i(1) \\ c_i(2) \\ \vdots \\ \vdots \\ c_i(N) \end{bmatrix} = \begin{bmatrix} h(1) & h(2) & \dots & h(M) & 0 & 0 \\ 0 & h(1) & \dots & \dots & h(M) & 0 \\ & & & \vdots & & \\ h(2) & \dots & h(M) & \dots & 0 & h(1) \\ & \vdots & & & \vdots & \end{bmatrix} \begin{bmatrix} c_{i+1}(1) \\ c_{i+1}(2) \\ \vdots \\ \vdots \\ c_{i+1}(N) \end{bmatrix}, \tag{5.4}$$

where N is the length of the signal and M is the length of the filter.

The multiwavelet transform computation is done by replacing the filter coefficients by matrices and signal coefficients by vectors. In matrix form this may be visualized as:

$$\begin{bmatrix} C_{i,1}(1) \\ C_{i,2}(1) \\ C_{i,1}(2) \\ C_{i,2}(2) \\ \vdots \\ C_{i,1}(N) \\ C_{i,2}(N) \end{bmatrix} = \begin{bmatrix} H_1(1) & H_2(1) & H_1(2) & H_2(2) & \dots \\ H_3(1) & H_4(1) & H_3(2) & H_4(2) & \dots \\ & & \vdots & & \end{bmatrix} \begin{bmatrix} C_{i+1,1}(1) \\ C_{i+1,2}(1) \\ C_{i+1,1}(2) \\ C_{i+1,2}(2) \\ \vdots \\ C_{i+1,1}(N) \\ C_{i+1,2}(N) \end{bmatrix}. \tag{5.5}$$

In Equation 5.5, R is 2 so that the signal, whose multiwavelet transform is taken is $2 \times N$ and there are M matrices as coefficients, where each is 2×2. Each element of a coefficient matrix may be taken as a scalar filter, where $H_1(-n)$ and $H_3(-n)$ filter the

first and $H_2(-n)$ and $H_4(-n)$ filter the second row of the coefficients. Moreover,

$$\begin{aligned} C_{i,1}(n) &= C_{i+1,1}(n) * H_1(-n) + C_{i+1,2}(n) * H_2(-n) \\ C_{i,2}(n) &= C_{i+1,1}(n) * H_3(-n) + C_{i+1,2}(n) * H_4(-n), \end{aligned} \tag{5.6}$$

where $*$ means convolution. Both rows of the multiwavelet and multiscaling coefficients at a coarser scale depend on both rows of the multiscaling coefficients at a higher scale. A similar interpretation of the multiwavelet transform has been done in [24], where the multiwavelet filter bank is seen as a time varying filter bank.

The redundant multiwavelet transform is obtained by padding each of the scalar filters with zeros when using Equation 5.6, which is the same as padding the matrix coefficients with zero matrices of size 2×2.

To demonstrate that such a procedure produces a translation invariant redundant multiwavelet transform (RMWT), Figure 5.4 is used, where the first plot is that of a signal, the second and third are the plots of the first rows of the RMWT's of the signal and a shifted version of it at an arbitrary scale obtained with the GHM multiwavelet.

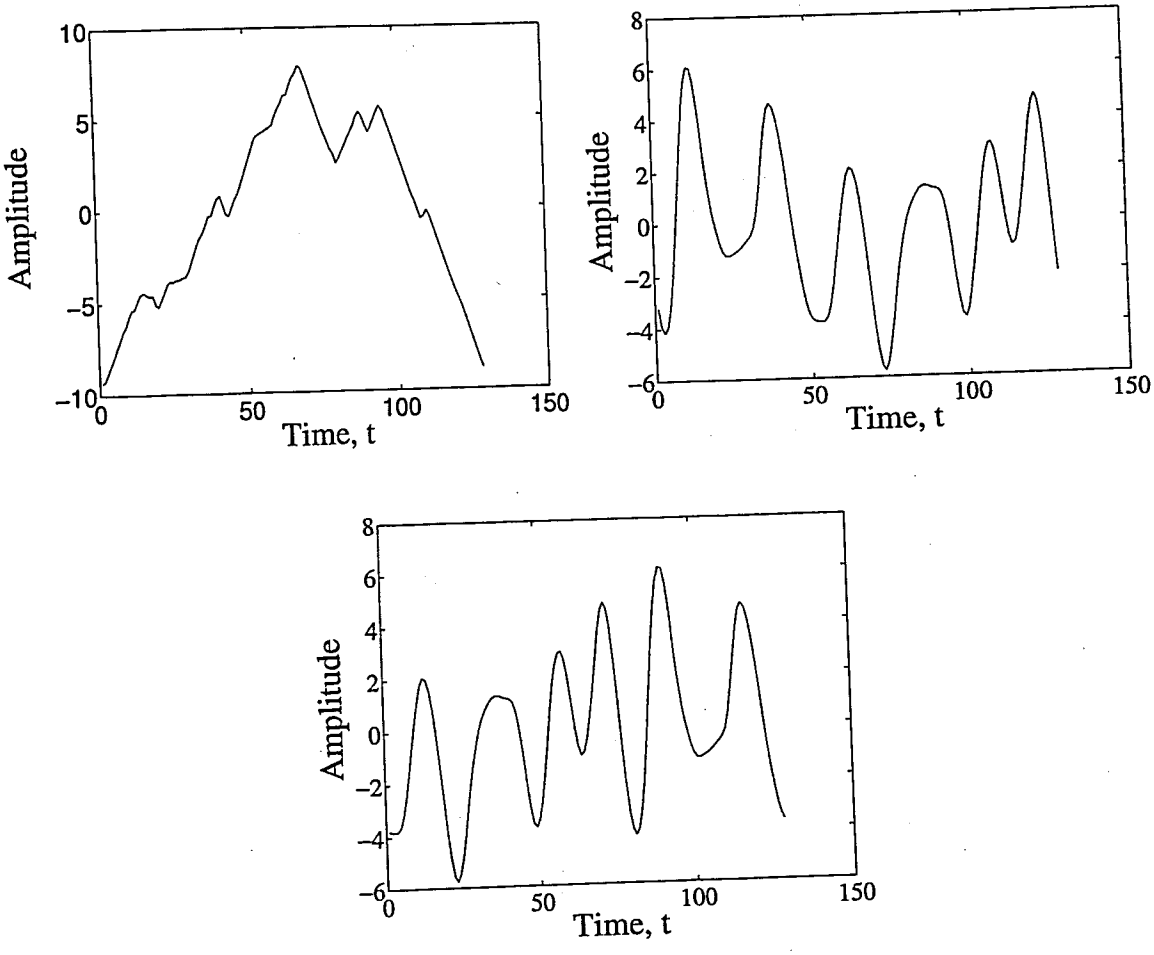

Figure 5.4. A signal, RMWT of the signal, RMWT of the shifted signal

6. EXPERIMENTAL RESULTS

In order to test the validity of what has been claimed until this point, some experiments have to be done, which are going to be computer simulations for recognition of objects using wavelet and multiwavelet transforms. Before presenting what the outcomes of the experiments are, the test set used in the simulations are introduced.

6.1. Test Set

The test set consists of images of 20 airplanes in their canonical (top) view as can be seen in Figure 6.1 and affine transformed versions of these images as in Figure 6.2. The top view images are obtained from [2], as some of the affine images. The rest has been produced by the author. Some of these objects are totally different while some of them are similar, like planes (7) and (20) or (18) and (19). The different objects will show the discriminating power of the invariant functions and the similar ones will show their ability to discriminate between small variations. The affine transformed versions of the images are the unknown objects, where an affine transform of each object in Figure 6.1 exists.

6.2. The Original Affine Invariant Wavelet Function

The simulations have first been done for the proposed affine invariant function in [2]. The function is reproduced here for convenience:

$$I(t) = \frac{\eta_{3,4,5,6,7,8}(t)}{\eta_{7,8,9,10,11,12}(t)}. \tag{6.1}$$

Since the total number of scales for which the boundary functions have to be decomposed is 12, they need to be resampled to have lengths 2^{12}. After the affine invariant functions for the reference and unknown images have been obtained, which can be denoted as $I_1(t)$ and $I_2(t)$, respectively, they are correlated using Equation 4.10, which

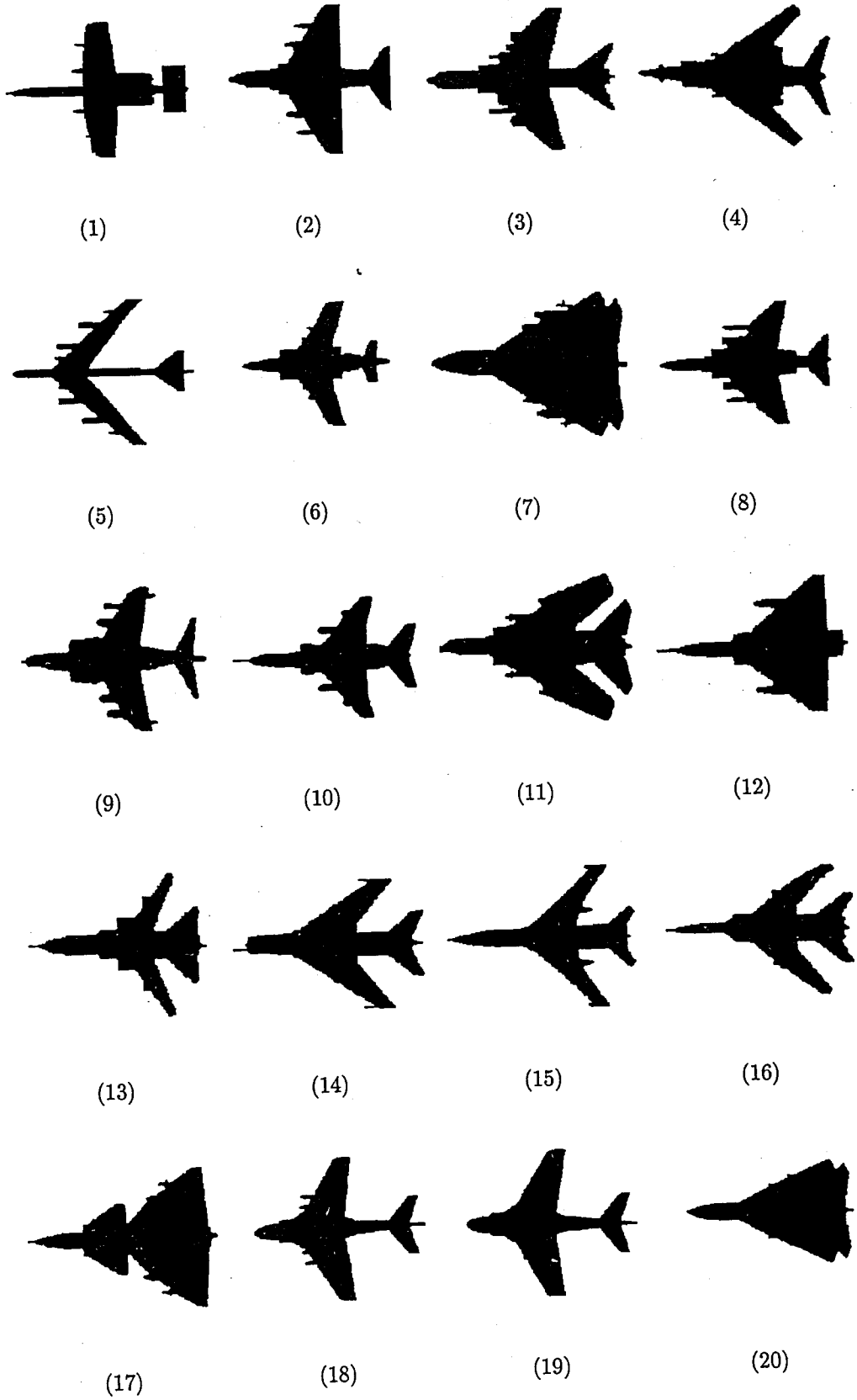

Figure 6.1. The reference images

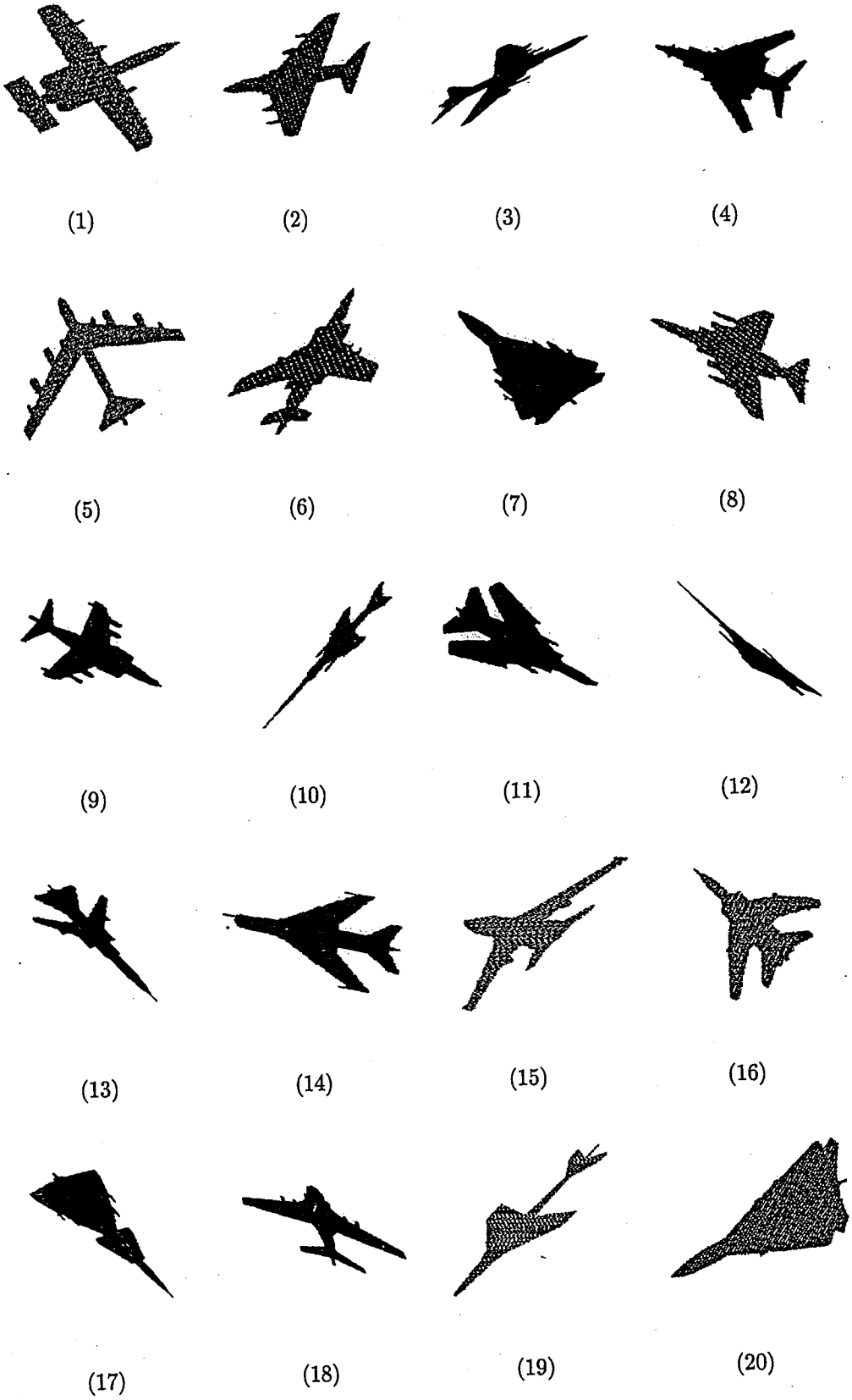

Figure 6.2. The affine images

is reproduced here:

$$\frac{\int I_1(t)I_2(t)dt}{\|I_1(t)\|\ \|I_2(t)\|}. \tag{6.2}$$

The reference invariant function, which has the maximum correlation with the invariant function of the unknown object will determine the identity of the object. The authors in [2] have used the filters proposed by Mallat and Zhong in [8] when performing their experiments.

While the results were being tabulated, the best five matches have been listed in five columns, where the first column shows the best match. For each unknown object, the correlation value between the invariant function of the object and that of a reference object is given in each column. The number in parentheses shows which of the reference objects have been used.

During the simulations, no noise has been added to the boundary obtained from the images, since the quantization noise that is present due to sampling will be enough.

By looking at Table 6.1, it can be said that the performance of the original affine invariant function is very poor. Only one of the objects, (7), has been recognized correctly. For 12 of the objects, the correct match is not even included among the best five matches. These results are in compliance with expectations, for the finer scale coefficients are very much affected by sampling of the boundary of the object and quantization noise. Hence, they should not be used in an object recognition system. To overcome the drawbacks of a system using this affine invariant function, it has been altered slightly as in Chapter 4 so that the new function is

$$I(t) = \frac{\eta_{2,3,4,5,6,7}(t)}{T_0^4}, \tag{6.3}$$

where T_0 is the total area occupied by the object in question. This function uses only six scales of the redundant wavelet transform.

Table 6.1. The best five matches for the original affine invariant wavelet function

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.8960 (7)**	0.8842 (1)	0.8700 (6)	0.8621 (14)	0.8515 (20)
(2)	**0.7422 (8)**	0.7313 (11)	0.6252 (5)	0.5808 (4)	0.5369 (5)
(3)	**0.9694 (7)**	0.9464 (1)	0.9302 (6)	0.9175 (14)	0.9134 (20)
(4)	**0.4560 (20)**	0.4503 (7)	0.4458 (1)	0.4397 (6)	0.4349 (14)
(5)	**0.8462 (20)**	0.8347 (7)	0.7719 (1)	0.7709 (18)	0.7608 (6)
(6)	**0.6338 (8)**	0.6264 (11)	0.5373 (5)	0.5098 (14)	0.5076 (6)
(7)	**0.9363 (7)**	0.9198 (1)	0.9050 (6)	0.8938 (14)	0.8867 (20)
(8)	**0.8450 (11)**	0.8286 (8)	0.7078 (5)	0.6816 (4)	0.6530 (17)
(9)	**0.5109 (19)**	0.5105 (9)	0.4867 (18)	0.4806 (8)	0.4679 (11)
(10)	**0.9004 (8)**	0.8791 (11)	0.7575 (5)	0.6602 (9)	0.6571 (17)
(11)	**0.7169 (7)**	0.6834 (1)	0.6718 (6)	0.6635 (20)	0.6539 (14)
(12)	**0.8462 (8)**	0.8053 (11)	0.7029 (5)	0.6028 (13)	0.5885 (17)
(13)	**0.7832 (19)**	0.7374 (18)	0.7256 (9)	0.6339 (7)	0.5958 (4)
(14)	**0.9671 (7)**	0.9508 (1)	0.9340 (6)	0.9235 (14)	0.9160 (20)
(15)	**0.5369 (7)**	0.5271 (1)	0.5257 (11)	0.5167 (8)	0.5120 (14)
(16)	**0.5792 (14)**	0.5636 (6)	0.5600 (1)	0.5491 (3)	0.5359 (8)
(17)	**0.7540 (8)**	0.7005 (11)	0.6177 (5)	0.5425 (13)	0.4913 (17)
(18)	**0.8437 (7)**	0.7899 (18)	0.7841 (19)	0.7756 (1)	0.7595 (5)
(19)	**0.7492 (11)**	0.7411 (8)	0.6366 (3)	0.6077 (9)	0.5784 (17)
(20)	**0.8687 (8)**	0.7880 (11)	0.7036 (5)	0.6344 (13)	0.5383 (17)

6.3. The Modified Affine Invariant Wavelet Function

When the modified affine invariant function is used in the simulations, the boundary is resampled so that it is of length 2^7, since this length is sufficient to decompose the component boundary functions, $x(t)$ and $y(t)$ for 7 scales. Even if the length is longer, the coarsest 6 scales should be selected. However, a longer length means the point matching stage, which aims to restore point correspondence, will be computationally more demanding. The accuracy will decrease as well, since the point matching done will not be perfect and the more the points are matched incorrectly, the more will be the drop in the correlation value between the correct reference function and the unknown function.

In order to show how the point matching algorithm has worked, the boundaries of the reference objects in the database, the boundaries of affine transformed versions of these objects, which have been processed are shown in Figures 6.3-6.22 along with their superimposed invariant functions obtained by using Mallat-Zhong filters in [8].

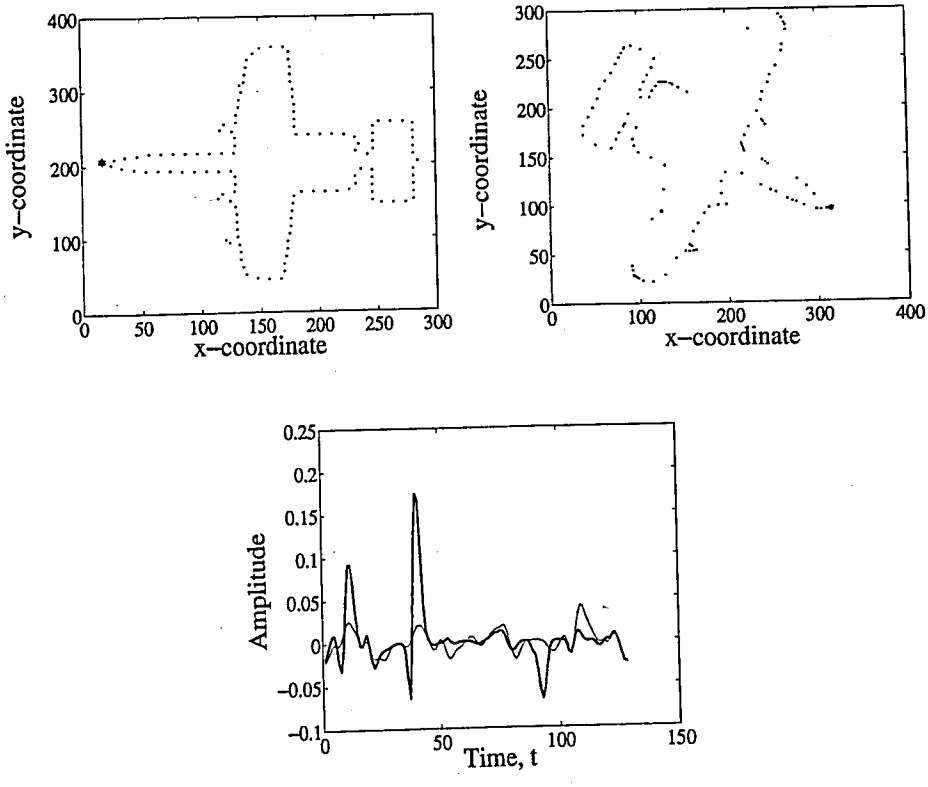

Figure 6.3. Plane 1, correlation is 0.4987

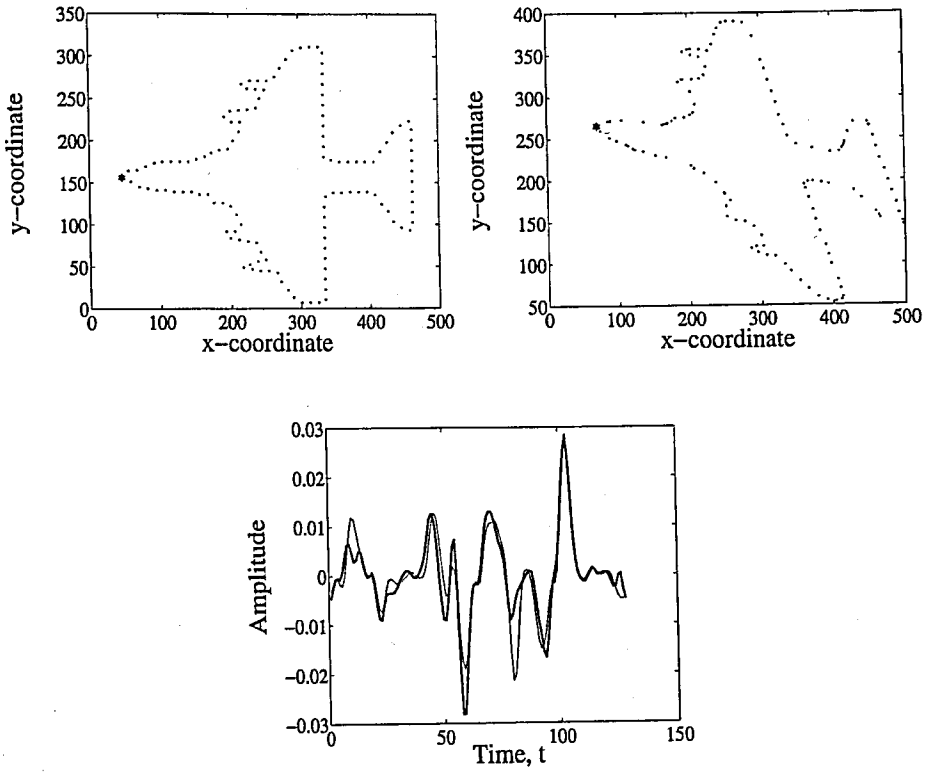

Figure 6.4. Plane 2, correlation is 0.9000

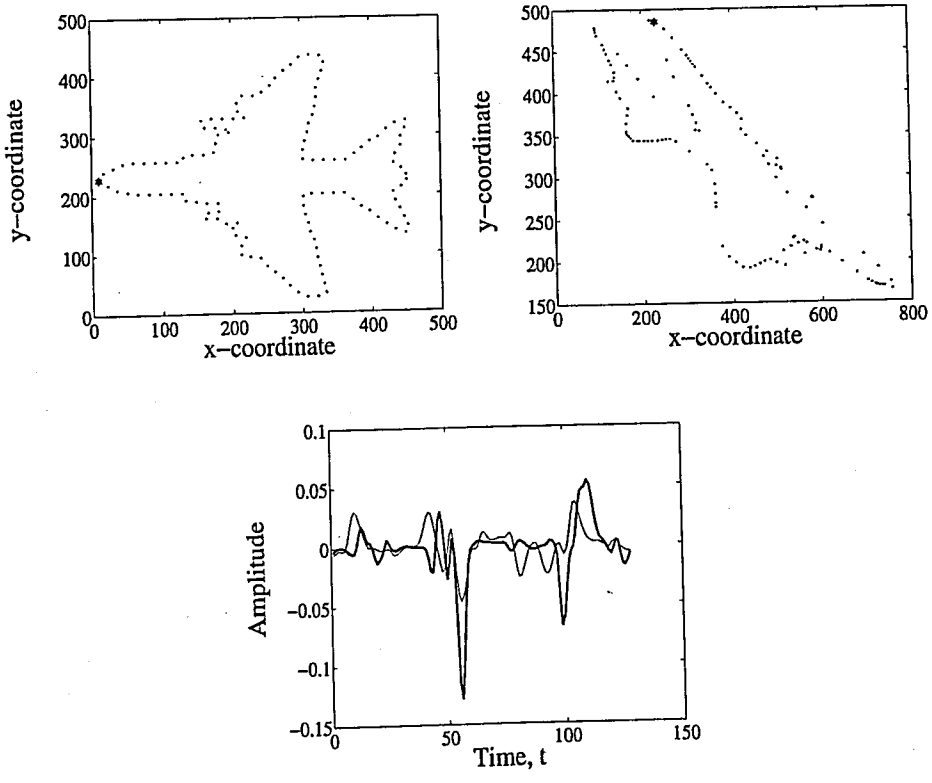

Figure 6.5. Plane 3, correlation is 0.4602

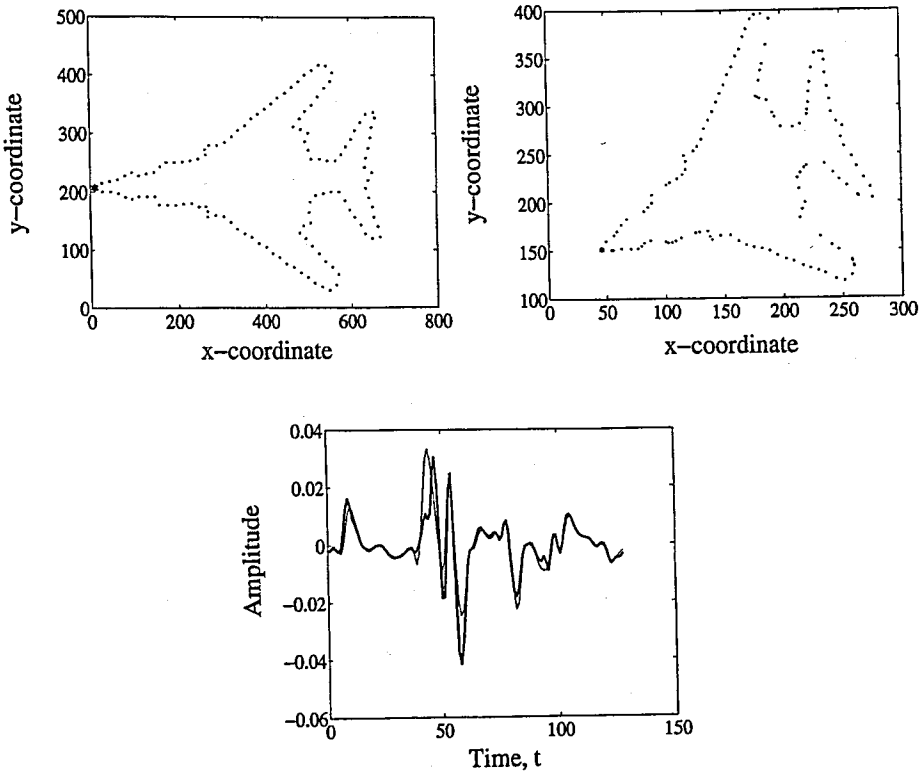

Figure 6.6. Plane 4, correlation is 0.8487

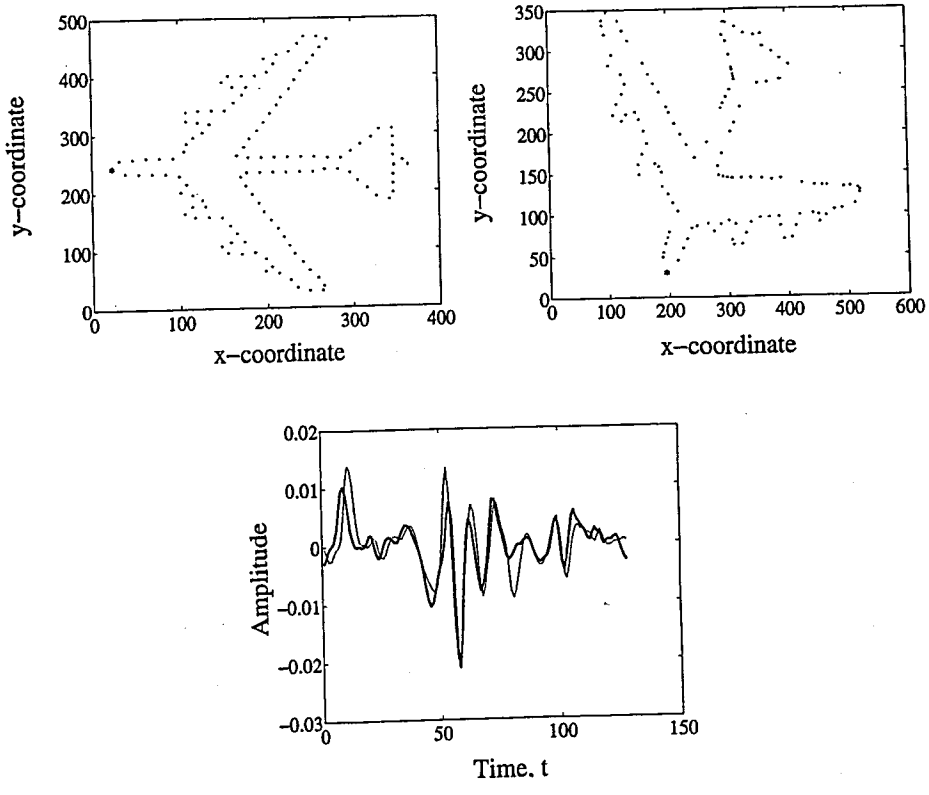

Figure 6.7. Plane 5, correlation is 0.7998

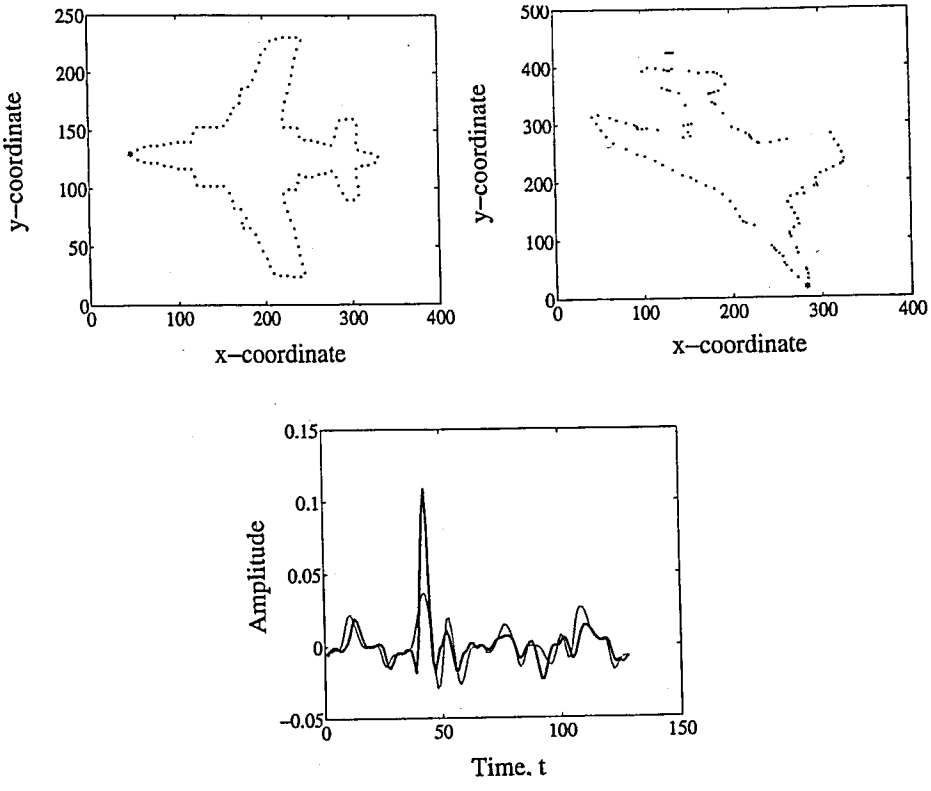

Figure 6.8. Plane 6, correlation is 0.6773

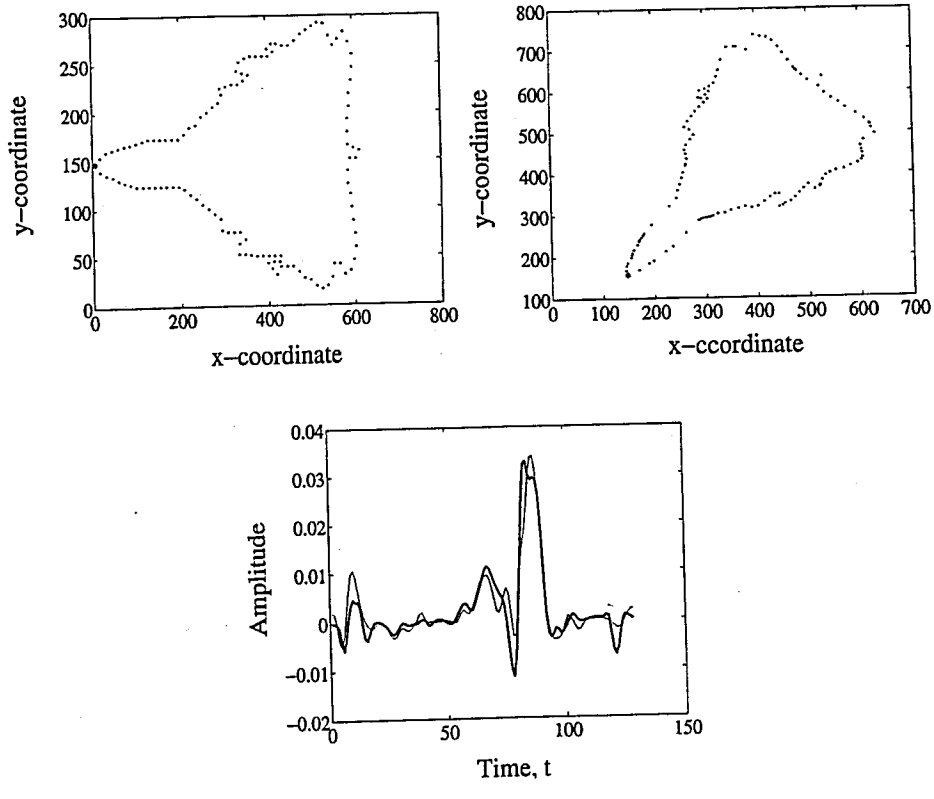

Figure 6.9. Plane 7, correlation is 0.9328

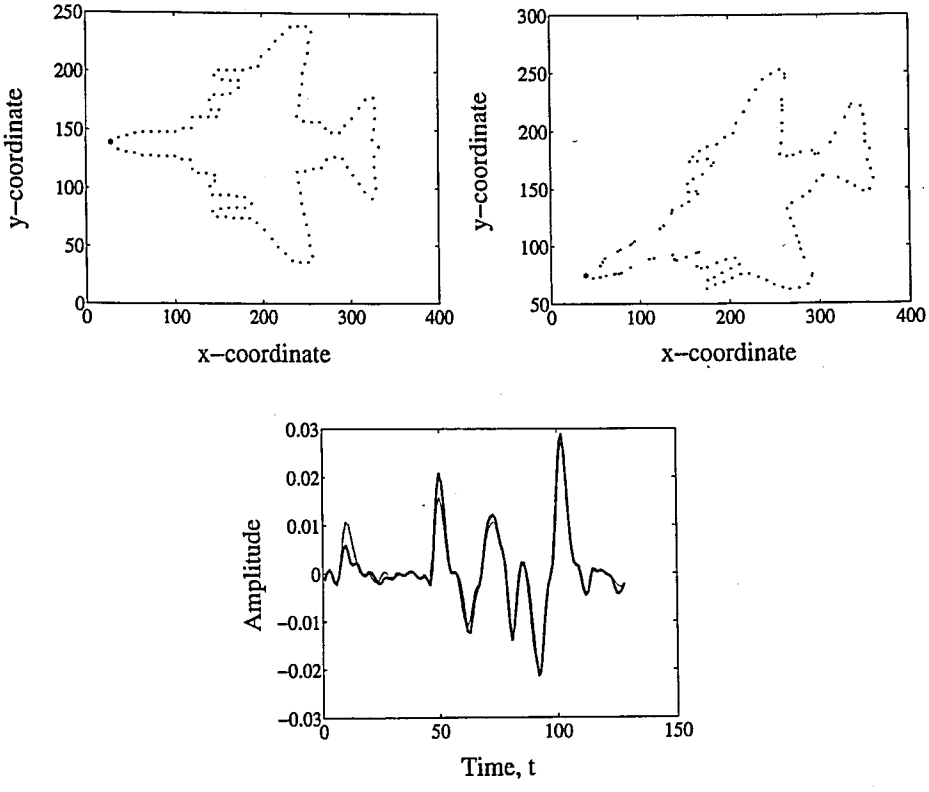

Figure 6.10. Plane 8, correlation is 0.9805

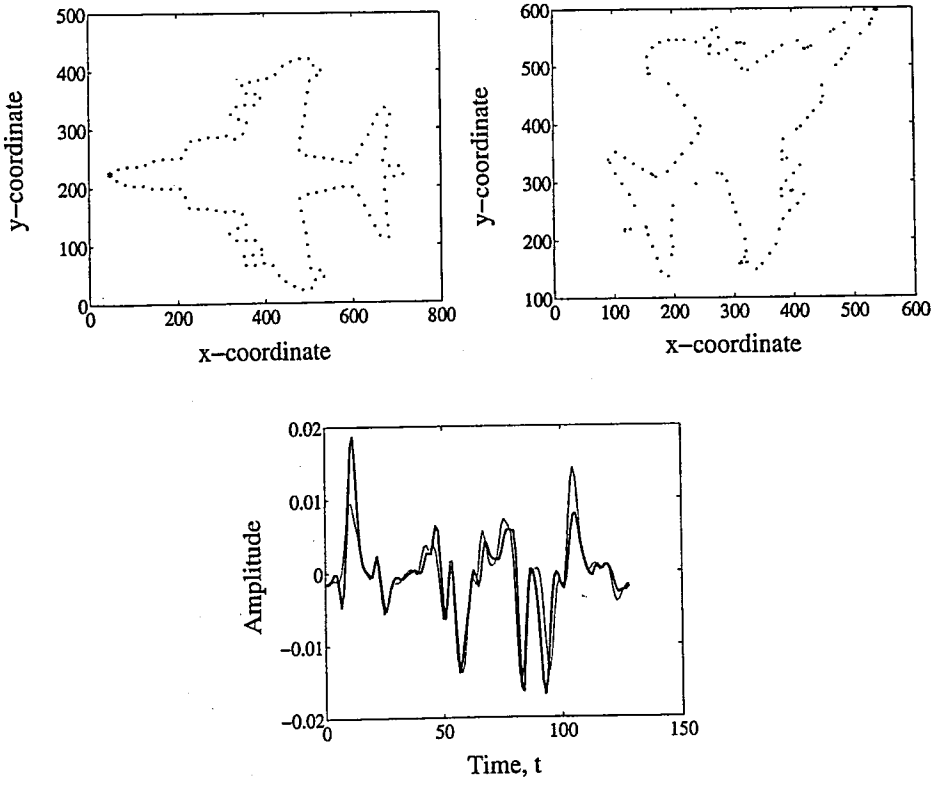

Figure 6.11. Plane 9, correlation is 0.8408

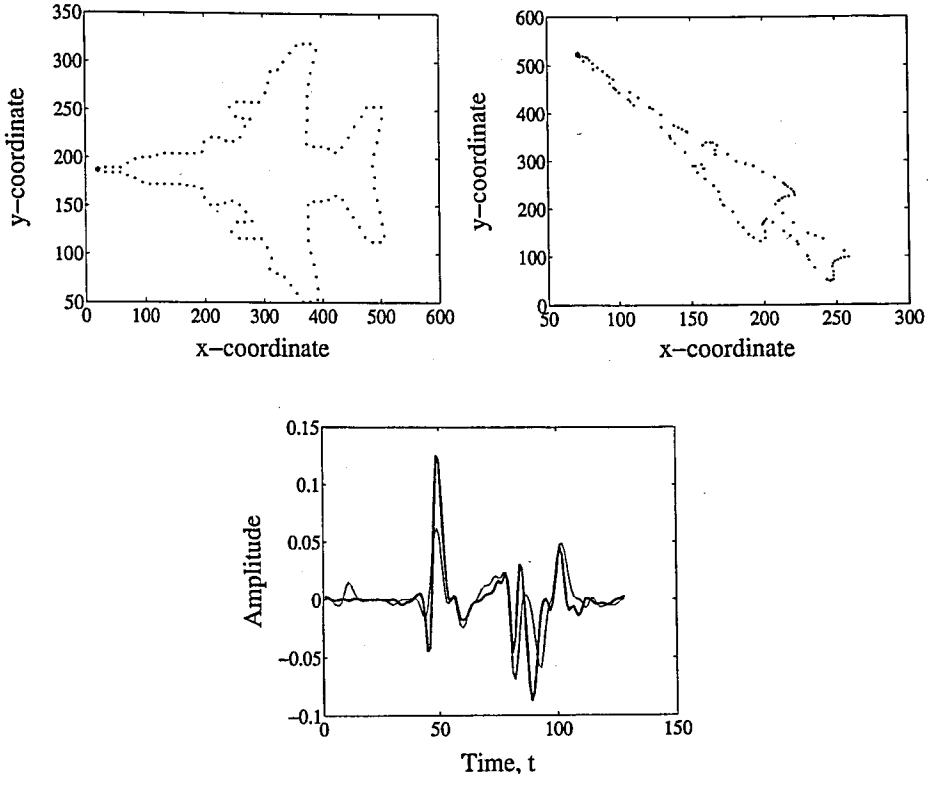

Figure 6.12. Plane 10, correlation is 0.6497

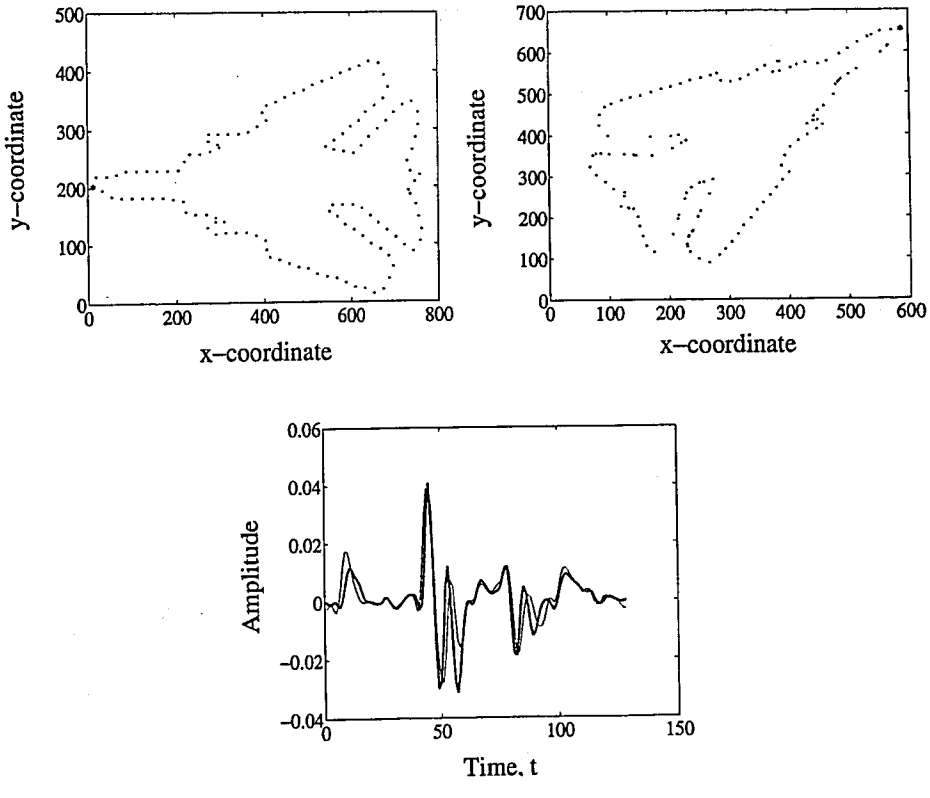

Figure 6.13. Plane 11, correlation is 0.8035

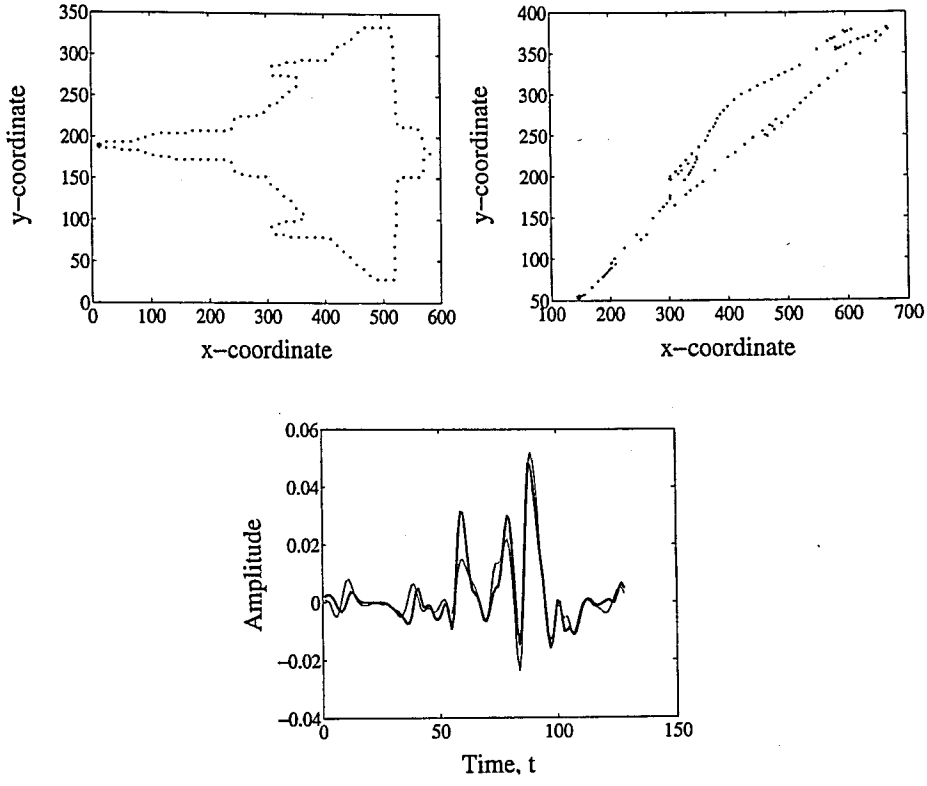

Figure 6.14. Plane 12, correlation is 0.9162

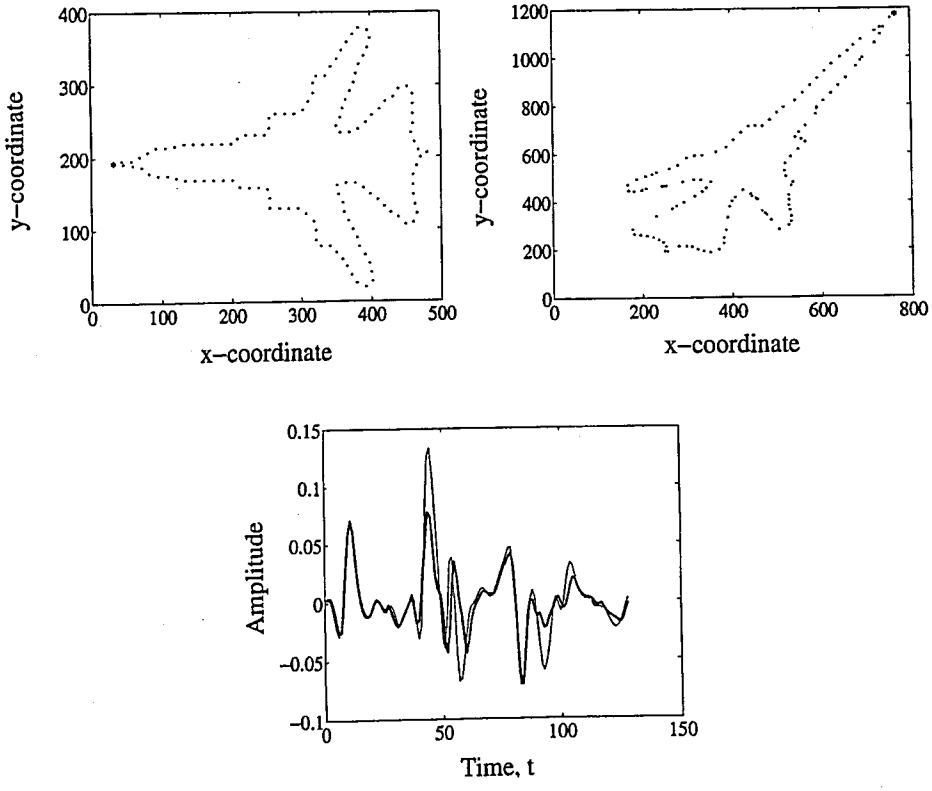

Figure 6.15. Plane 13, correlation is 0.8435

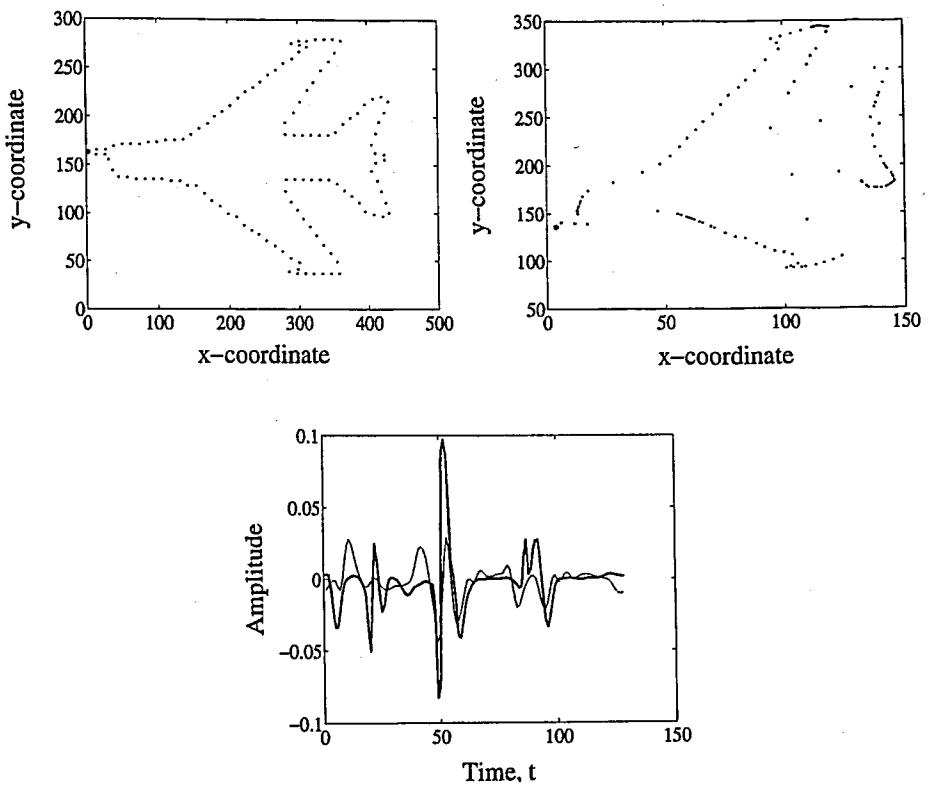

Figure 6.16. Plane 14, correlation is 0.4989

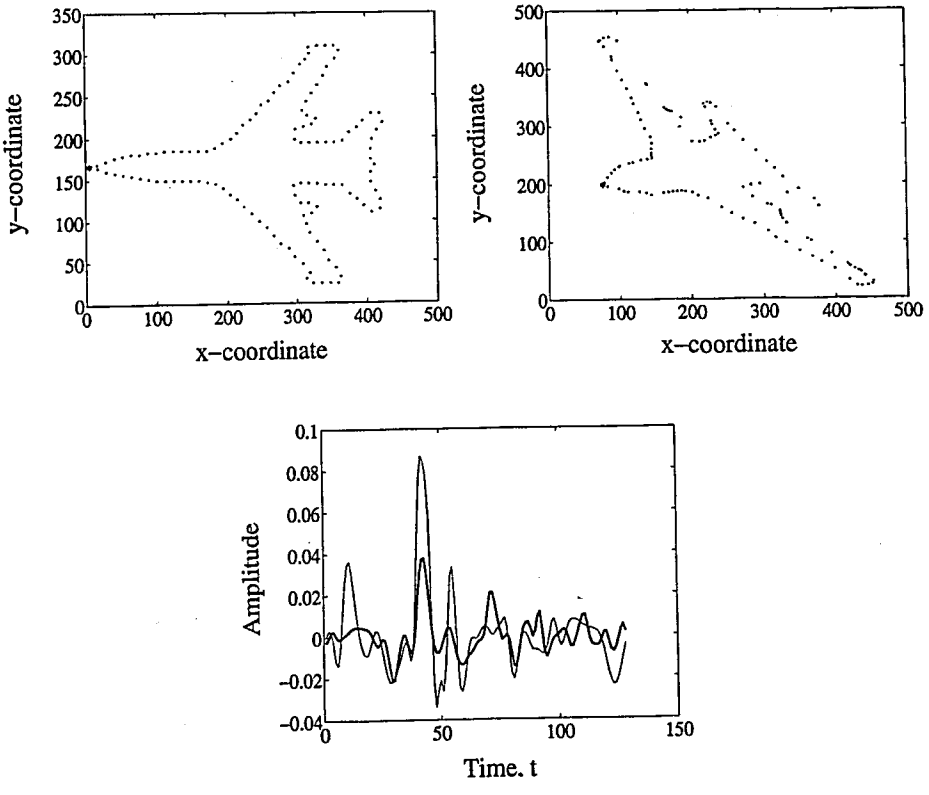

Figure 6.17. Plane 15, correlation is 0.7454

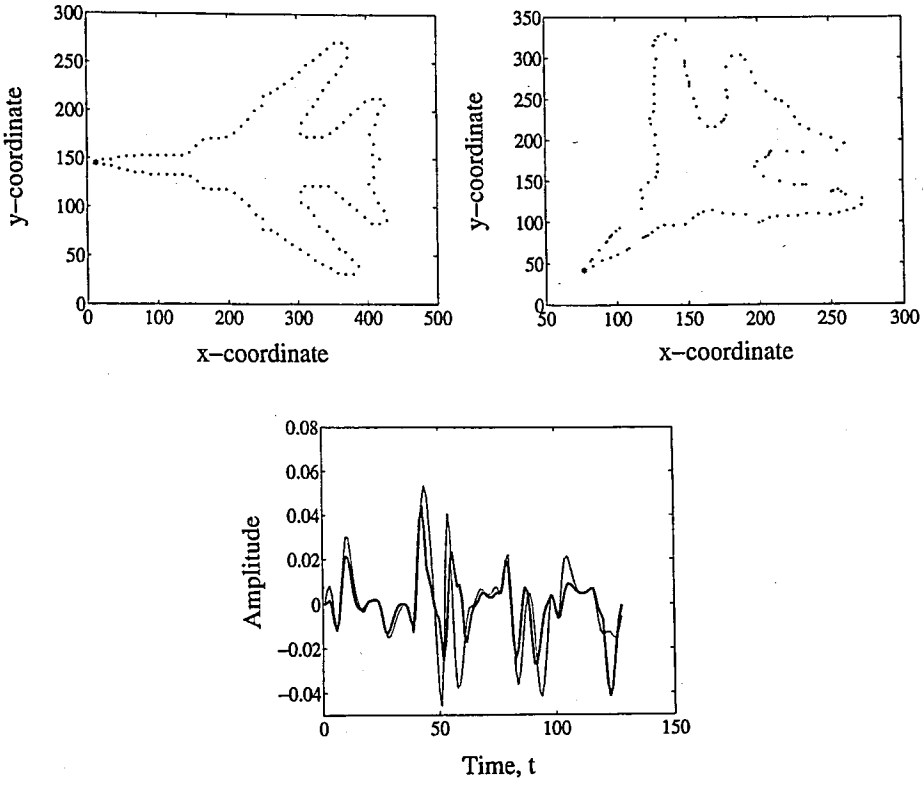

Figure 6.18. Plane 16, correlation is 0.6749

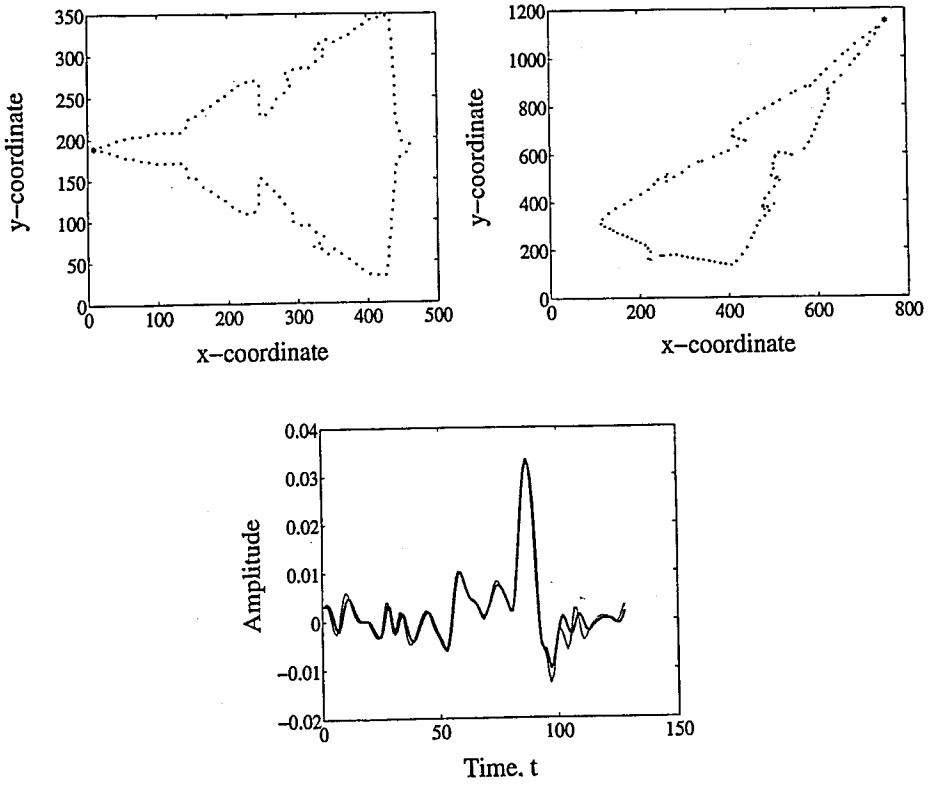

Figure 6.19. Plane 17, correlation is 0.9776

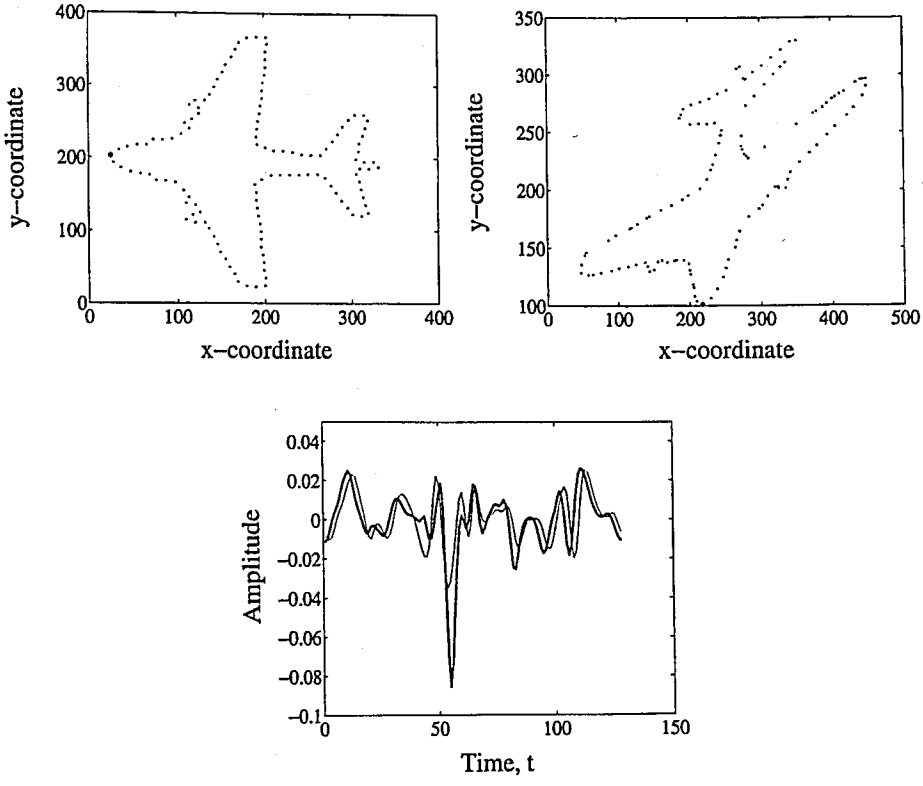

Figure 6.20. Plane 18, correlation is 0.6780

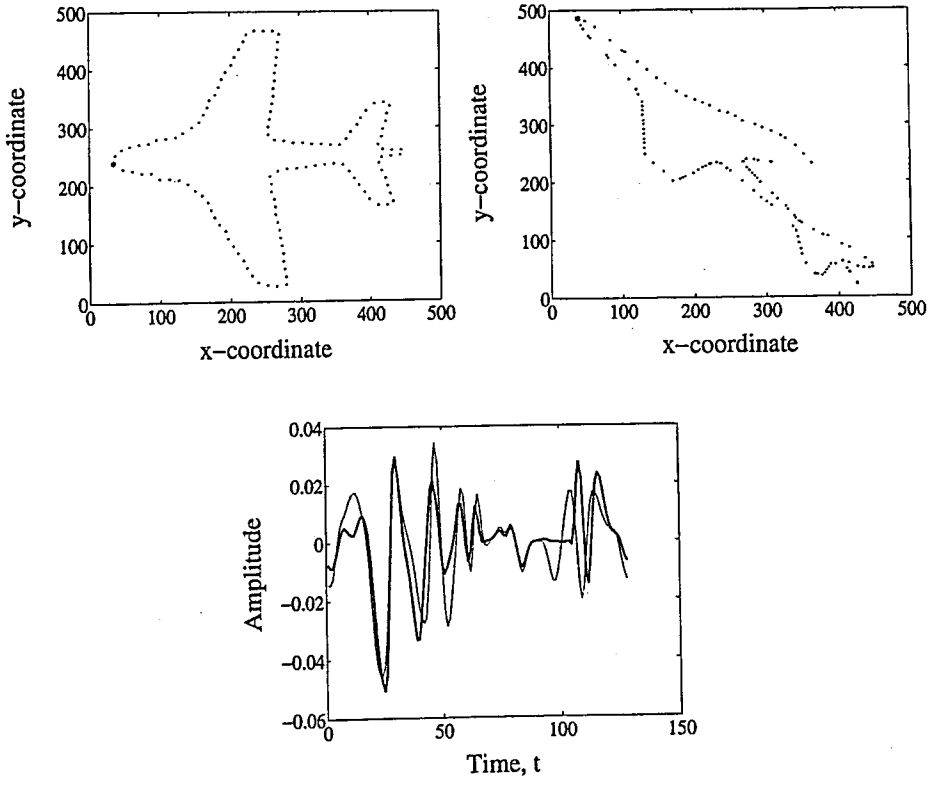

Figure 6.21. Plane 19, correlation is 0.7450

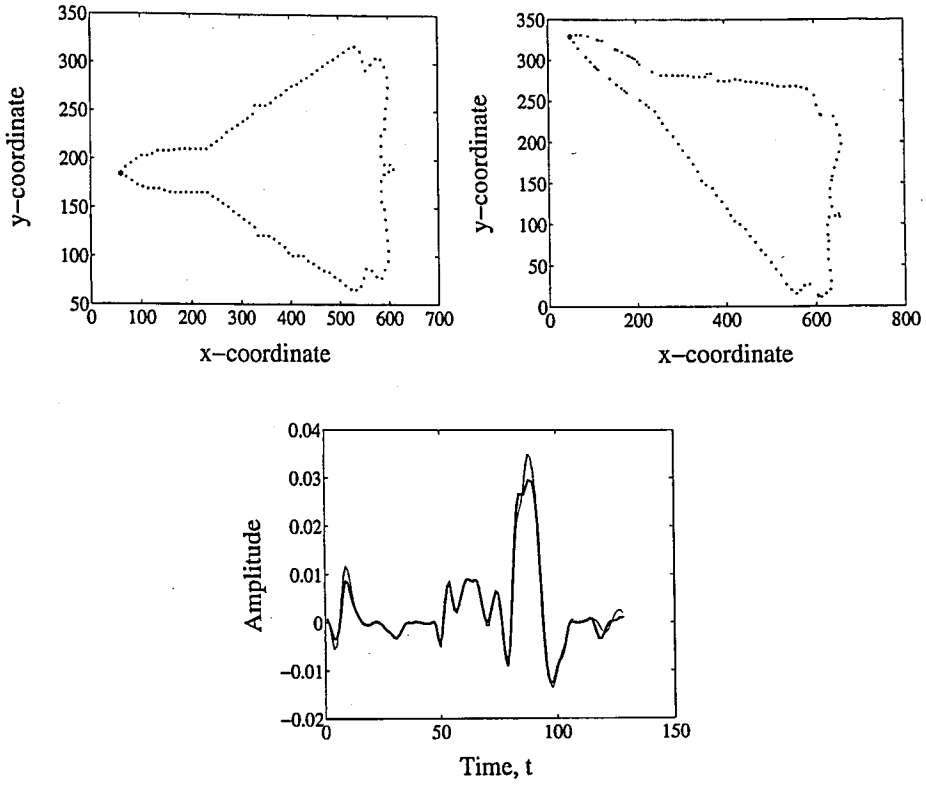

Figure 6.22. Plane 20, correlation is 0.9896

By looking at Figures 6.3-6.22, it may be said that the point matching algorithm does not work very well for some of the objects. For Planes 1 and 14 this fact is very obvious. The reason, why the algorithm did not work in these two cases, is that along the wings the tracing and tangent lines are collinear, making the calculation of the area parameter values difficult. The fact that the affine transforms of the points on the reference curve are not found very accurately, does not mean an incorrect recognition will be done. This may be a means to learn the robustness of the wavelet or multiwavelet coefficients to noise, which is nonuniform in this case. The nose of Plane 3 could not be found, and hence point correspondence is not obtained, which should make it impossible to find the correct match. In the figures, the nose is shown with a hexagon.

To evaluate the performance of the modified affine invariant wavelet function, simulations have been performed for each unknown object in Figure 6.2. Two different sets of filters, Mallat-Zhong, which are the ones used by the authors in [2] and Daubechies-4, which are orthogonal, nonsymmetric filters of length 4, have been used.

Table 6.2. The best five matches for the modified affine invariant wavelet function using Mallat-Zhong filters

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.5957 (13)**	0.5708 (19)	0.5632 (7)	0.5558 (8)	0.5066 (4)
(2)	**0.9000 (2)**	0.8502 (8)	0.6770 (13)	0.6498 (3)	0.5466 (6)
(3)	**0.5661 (4)**	0.5350 (14)	0.4783 (2)	0.4602 (3)	0.4584 (6)
(4)	**0.8487 (4)**	0.6967 (10)	0.6776 (15)	0.6305 (11)	0.6297 (13)
(5)	**0.7998 (5)**	0.5448 (19)	0.5176 (2)	0.5168 (11)	0.4853 (15)
(6)	**0.6844 (15)**	0.6773 (6)	0.6006 (13)	0.5879 (11)	0.5592 (3)
(7)	**0.9328 (7)**	0.8837 (20)	0.8565 (17)	0.7417 (12)	0.5906 (15)
(8)	**0.9805 (8)**	0.8370 (2)	0.6594 (13)	0.5859 (10)	0.5764 (11)
(9)	**0.8408 (9)**	0.7866 (10)	0.6852 (6)	0.6524 (13)	0.6432 (4)
(10)	**0.7391 (11)**	0.7266 (15)	0.6497 (10)	0.5963 (13)	0.5617 (8)
(11)	**0.8035 (11)**	0.6395 (13)	0.6012 (6)	0.5576 (17)	0.5523 (3)
(12)	**0.9162 (12)**	0.6781 (17)	0.6014 (7)	0.5928 (13)	0.5492 (10)
(13)	**0.8435 (13)**	0.7133 (15)	0.6128 (11)	0.5685 (12)	0.5511 (5)
(14)	**0.6293 (3)**	0.6092 (11)	0.5999 (9)	0.5233 (10)	0.5199 (4)
(15)	**0.7454 (15)**	0.6538 (13)	0.5566 (8)	0.5433 (4)	0.4882 (16)
(16)	**0.6749 (16)**	0.6675 (8)	0.6415 (13)	0.6246 (12)	0.5793 (15)
(17)	**0.9776 (17)**	0.8982 (7)	0.8503 (20)	0.7081 (12)	0.5114 (3)
(18)	**0.7241 (9)**	0.6780 (18)	0.6272 (10)	0.5482 (13)	0.5149 (6)
(19)	**0.7450 (19)**	0.6055 (15)	0.4973 (16)	0.4973 (16)	0.4575 (14)
(20)	**0.9896 (20)**	0.9175 (7)	0.8517 (17)	0.7066 (12)	0.7045 (15)

Table 6.3. The best five matches for the modified affine invariant wavelet function using Daubechies-4 filters

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.6551 (8)**	0.6503 (12)	6371 (13)	0.5331 (17)	0.5231 (15)
(2)	**0.7172 (2)**	0.7049 (8)	0.5123 (19)	0.5121 (1)	0.4406 (17)
(3)	**0.6167 (1)**	0.5574 (4)	0.5216 (19)	0.5099 (5)	0.4885 (18)
(4)	**0.9781 (4)**	0.7739 (9)	0.7256 (10)	0.6619 (17)	0.6246 (8)
(5)	**0.5558 (5)**	0.5408 (11)	0.5385 (16)	0.4945 (2)	0.4702 (15)
(6)	**0.8340 (6)**	0.7071 (10)	0.6872 (3)	0.6830 (4)	0.6606 (9)
(7)	**0.6019 (17)**	0.5414 (12)	0.4863 (7)	0.4660 (6)	0.4617 (3)
(8)	**0.9811 (8)**	0.6136 (2)	0.5895 (18)	0.4929 (17)	0.4783 (1)
(9)	**0.8655 (9)**	0.6968 (4)	0.6739 (10)	0.6133 (4)	0.5319 (17)
(10)	**0.8014 (4)**	0.7673 (10)	0.5368 (5)	0.4570 (8)	0.4173 (11)
(11)	**0.5639 (3)**	0.5557 (9)	0.5166 (1)	0.5006 (11)	0.4283 (15)
(12)	**0.9390 (12)**	0.5539 (6)	0.5407 (18)	0.5385 (17)	0.5292 (8)
(13)	**0.7726 (13)**	0.6087 (19)	0.5155 (3)	0.5135 (15)	0.5028 (6)
(14)	**0.6339 (1)**	0.6153 (3)	0.5649 (11)	0.5531 (5)	0.5339 (18)
(15)	**0.5600 (15)**	0.5312 (19)	0.4982 (5)	0.4922 (18)	0.4751 (10)
(16)	**0.7760 (16)**	0.6410 (13)	0.4947 (5)	0.4678 (6)	0.4552 (14)
(17)	**0.8746 (17)**	0.6616 (8)	0.5948 (9)	0.5857 (11)	0.5670 (7)
(18)	**0.6973 (9)**	0.6805 (10)	0.6566 (4)	0.6260 (6)	0.6051 (18)
(19)	**0.7690 (4)**	0.7132 (19)	0.5996 (8)	0.5679 (10)	0.5636 (14)
(20)	**0.8838 (20)**	0.5773 (17)	0.5139 (6)	0.4979 (9)	0.4919 (7)

According to the results displayed in Tables 6.2 and 6.3, there is a great improvement compared to the original affine invariant wavelet function, which was able to recognize only one of the airplanes, Plane 7, correctly. For Mallat-Zhong filters, this number is 14, and when Daubechies-4 filters are used, 12 of the airplanes are identified correctly. As expected, Planes 1 and 14, whose area parametrization steps were not very successful, could not be recognized with either filter. Plane 3 is not recognized, either, due to the fact that the starting point, which is to have the area parameter value 0, could not be found resulting in erroneous parametrization and point matching. The point from which Plane 10 has been viewed prevents it from being recognized, which probably makes point matching harder. Although there is an overlap between the objects that could not be recognized by either of the filters, such objects also tend to vary depending on the individual characteristics of filters.

On the average, the maximum correlation values between the invariant functions when Daubechies-4 filters are used, are lower than the case when Mallat-Zhong filters are used. For instance, Plane 15 is successfully recognized although the correlation between the reference image for Plane 15 and the unknown image is only 0.5600.

6.4. The Multiwavelet Affine Invariant Function

Multiwavelets have been used in an object recognition application, since they have shown superior performance compared to wavelets in some applications like image compression and denoising. Hence, they may be expected to outperform wavelets in this field as well. The aim of this section is to explore this.

When the multiwavelet coefficients at different rows are used to form separate invariant functions, the individual functions are not successful. This is demonstrated with Tables 6.4 and 6.5, where GHM multiwavelets with repeated row preprocessing have been used. Selection has been done according to either the mean or the maximum value. The number of correct matches turn out to be 4 and 8 when the criterion is the mean and the maximum correlation, respectively.

Table 6.4. The best five matches with GHM multiwavelet for two separate functions-sorting according to mean correlation

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.6338 (9)**	0.5874 (1)	0.5160 (19)	0.5082 (5)	0.4994 (20)
(2)	**0.6361 (3)**	0.5813 (2)	0.5572 (20)	0.5162 (6)	0.5029 (9)
(3)	**0.6401 (10)**	0.6235 (4)	0.5633 (20)	0.5453 (13)	0.5331 (7)
(4)	**0.6278 (20)**	0.6122 (13)	0.5895 (6)	0.5770 (4)	0.5337 (10)
(5)	**0.5959 (10)**	0.5734 (4)	0.5248 (3)	0.5065 (5)	0.4924 (13)
(6)	**0.6332 (1)**	0.5259 (6)	0.4733 (3)	0.4449 (9)	0.4215 (13)
(7)	**0.5795 (13)**	0.5222 (10)	0.5177 (4)	0.5135 (19)	0.4348 (16)
(8)	**0.7287 (8)**	0.5865 (3)	0.5636 (20)	0.4936 (13)	0.4501 (9)
(9)	**0.6174 (9)**	0.5744 (7)	0.5051 (20)	0.5024 (1)	0.4939 (12)
(10)	**0.6263 (20)**	0.5619 (7)	0.5175 (19)	0.4956 (16)	0.4716 (6)
(11)	**0.6082 (7)**	0.5788 (20)	0.5671 (11)	0.5646 (10)	0.4943 (14)
(12)	**0.5189 (12)**	0.5005 (5)	0.4757 (4)	0.4721 (16)	0.4682 (20)
(13)	**0.6289 (13)**	0.5853 (4)	0.5338 (7)	0.5196 (5)	0.4915 (10)
(14)	**0.6410 (10)**	0.5819 (4)	0.5395 (20)	0.4895 (14)	0.4705 (12)
(15)	**0.5186 (13)**	0.5050 (9)	0.5019 (4)	0.4717 (3)	0.4694 (10)
(16)	**0.6509 (7)**	0.6469 (20)	0.6352 (13)	0.5417 (10)	0.5274 (3)
(17)	**0.6518 (4)**	0.6398 (17)	0.5633 (13)	0.4969 (16)	0.4964 (14)
(18)	**0.6371 (9)**	0.6181 (18)	0.6098 (17)	0.5961 (7)	0.5123 (10)
(19)	**0.6103 (20)**	0.5957 (10)	0.5205 (19)	0.5089 (18)	0.5067 (6)
(20)	**0.6073 (10)**	0.5103 (14)	0.4680 (19)	0.4663 (20)	0.4396 (7)

Table 6.5. The best five matches with GHM multiwavelet for two separate functions-sorting according to maximum correlation

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.7852 (9)**	0.7189 (7)	0.7074 (19)	0.6804 (1)	0.6413 (11)
(2)	**0.8951 (2)**	0.7505 (3)	0.6980 (6)	0.6777 (20)	0.6700 (9)
(3)	**0.8273 (20)**	0.7833 (10)	0.7499 (4)	0.5946 (7)	0.5877 (13)
(4)	**0.8885 (4)**	0.8233 (20)	0.7716 (10)	0.6654 (6)	0.6265 (13)
(5)	**0.7696 (4)**	0.7541 (5)	0.7257 (3)	0.6843 (10)	0.6519 (20)
(6)	**0.7715 (6)**	0.7229 (1)	0.6219 (3)	0.5774 (9)	0.5337 (13)
(7)	**0.7817 (13)**	0.7046 (3)	0.6313 (10)	0.6038 (14)	0.5615 (4)
(8)	**0.9316 (8)**	0.7443 (3)	0.6970 (20)	0.5895 (2)	0.5795 (16)
(9)	**0.8630 (9)**	0.6553 (20)	0.6481 (7)	0.6475 (8)	0.5975 (12)
(10)	**0.9068 (20)**	0.6694 (13)	0.6525 (7)	0.6510 (12)	0.5782 (16)
(11)	**0.8205 (11)**	0.7919 (20)	0.7498 (10)	0.7431 (7)	0.6482 (5)
(12)	**0.6975 (12)**	0.6608 (5)	0.6346 (4)	0.5841 (18)	0.4838 (20)
(13)	**0.7605 (13)**	0.6871 (10)	0.6564 (4)	0.6039 (5)	0.5594 (7)
(14)	**0.7018 (4)**	0.7015 (10)	0.6931 (20)	0.6595 (17)	0.5938 (12)
(15)	**0.7519 (4)**	0.6382 (9)	0.5974 (13)	0.5321 (10)	0.5188 (12)
(16)	**0.8118 (20)**	0.7967 (13)	0.7421 (10)	0.6620 (7)	0.6265 (19)
(17)	**0.9035 (17)**	0.7492 (4)	0.6228 (13)	0.6103 (15)	0.5809 (8)
(18)	**0.7642 (9)**	0.7165 (17)	0.6994 (4)	0.6982 (18)	0.6652 (1)
(19)	**0.8477 (20)**	0.7305 (10)	0.5984 (5)	0.5928 (18)	0.5840 (4)
(20)	**0.6649 (10)**	0.6308 (20)	0.5788 (7)	0.5339 (14)	0.5078 (19)

As a result, the multiwavelet coefficients that have the bigger magnitudes are selected to be used in one affine invariant function. Simulation results turned out to be better in such a case. The multiwavelets used in the simulations along with some of their properties are listed below, where R is the number of scaling and wavelet functions, l and m are the number of multiscaling and multiwavelet coefficient matrices, respectively, and *App* is the approximation order. When a system has an approximation order of 2, for instance, it is capable of representing exactly by linear combinations of translates of $\phi_1(t-k)$, $\phi_2(t-k)$, $k \in \mathbf{Z}$, constant and linear functions, which are functions of order 0 and 1, respectively.

GHM Geronimo-Hardin-Massopust orthogonal symmetric multi-filter bank

$R: 2, \quad l: 4, \quad m: 4, \quad App: 2$

CL Chui-Lian orthogonal symmetric multi-filter bank

$R: 2, \quad l: 3, \quad m: 3, \quad App: 2$

SA4 orthogonal symmetric multi-filter bank constructed by Shen, Tan, and Tham

$R: 2, \quad l: 4, \quad m: 4, \quad App: 1$

BIH52S most smooth biorthogonal symmetric multi-filter bank, dual to Hermite cubic filter bank BIH32S

$R: 2, \quad l: 5, \quad m: 3, \quad App: 2$

BIGHM2 biorthogonal multi-filter bank obtained from the GHM multi-filter bank by factoring out one approximation order, dual to BIGHM6

$R: 2, \quad l: 2, \quad m: 6, \quad App: 1$

Sometimes it is helpful to replace one orthogonal basis by two biorthogonal ones: One for decomposition (analysis) and one for reconstruction (synthesis) [12]. The biorthogonal multi-filter banks above are the analysis filters naturally.

When repeated row preprocessing is employed better results are obtained compared to the case when approximation preprocessing is used. This is not surprising, since it is known that oversampled data representations are useful for feature extraction [11]. To exemplify this, both repeated row and approximation preprocessing have been used for GHM whereas only repeated row preprocessing has been used for the rest.

Table 6.6. The best five matches with GHM multiwavelet, repeated row preprocessing

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.6298 (1)**	0.5763 (4)	0.4954 (10)	0.4724 (6)	0.4550 (9)
(2)	**0.8805 (2)**	0.7059 (19)	0.5983 (12)	0.5920 (7)	0.5328 (16)
(3)	**0.8451 (6)**	0.7836 (14)	0.7413 (16)	0.7135 (15)	0.6306 (11)
(4)	**0.9814 (4)**	0.8798 (14)	0.8775 (6)	0.8437 (2)	0.8393 (10)
(5)	**0.8494 (5)**	0.7872 (17)	0.7750 (19)	0.7441 (7)	0.7330 (20)
(6)	**0.9252 (6)**	0.8321 (14)	0.7394 (3)	0.7270 (16)	0.7224 (4)
(7)	**0.9492 (7)**	0.8516 (14)	0.7758 (12)	0.7287 (20)	0.6803 (17)
(8)	**0.9402 (8)**	0.7829 (16)	0.7014 (3)	0.6585 (2)	0.6109 (1)
(9)	**0.9072 (4)**	0.8831 (10)	0.8425 (19)	0.7686 (18)	0.7661 (9)
(10)	**0.8990 (14)**	0.7180 (13)	0.7173 (11)	0.6837 (10)	0.6666 (4)
(11)	**0.9381 (11)**	0.9010 (14)	0.7715 (16)	0.7671 (4)	0.5984 (6)
(12)	**0.9071 (12)**	0.8877 (7)	0.7825 (14)	0.6962 (17)	0.6436 (20)
(13)	**0.9633 (13)**	0.9416 (14)	0.8982 (15)	0.7956 (16)	0.7389 (10)
(14)	**0.8691 (14)**	0.7910 (13)	0.7046 (11)	0.6930 (16)	0.6289 (1)
(15)	**0.9533 (15)**	0.8717 (16)	0.8713 (14)	0.8628 (6)	0.8293 (4)
(16)	**0.9561 (16)**	0.9409 (14)	0.7907 (13)	0.6610 (12)	0.6547 (6)
(17)	**0.9741 (17)**	0.7848 (15)	0.6420 (16)	0.5439 (6)	0.5417 (8)
(18)	**0.7666 (1)**	0.6318 (4)	0.4401 (12)	0.4309 (5)	0.4286 (18)
(19)	**0.7649 (19)**	0.6269 (1)	0.5311 (12)	0.5073 (3)	0.4399 (4)
(20)	**0.9761 (20)**	0.8412 (14)	0.8074 (17)	0.7997 (15)	0.7494 (3)

Table 6.7. The best five matches with GHM multiwavelet, approximation preprocessing

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.7578 (8)**	0.7045 (10)	0.6027 (2)	0.5779 (14)	0.5595 (4)
(2)	**0.9121 (2)**	0.8658 (8)	0.6500 (10)	0.6100 (13)	0.5325 (11)
(3)	**0.7954 (17)**	0.6983 (12)	0.6521 (11)	0.6095 (15)	0.5888 (16)
(4)	**0.9056 (4)**	0.6535 (8)	0.4950 (2)	0.4851 (10)	0.4630 (18)
(5)	**0.7124 (5)**	0.6615 (11)	0.5774 (3)	0.5686 (10)	0.5481 (18)
(6)	**0.6308 (10)**	0.5987 (8)	0.5858 (2)	0.5431 (6)	0.5312 (4)
(7)	**0.7672 (7)**	0.6542 (16)	0.5912 (11)	0.5793 (17)	0.5124 (6)
(8)	**0.9954 (8)**	0.7699 (2)	0.6409 (11)	0.6171 (5)	0.5853 (4)
(9)	**0.8598 (8)**	0.7950 (9)	0.7568 (4)	0.7276 (6)	0.6623 (10)
(10)	**0.7139 (17)**	0.6365 (2)	0.5189 (9)	0.4911 (3)	0.4570 (4)
(11)	**0.7560 (14)**	0.7063 (11)	0.6837 (17)	0.6140 (15)	0.5121 (12)
(12)	**0.7262 (12)**	0.6184 (6)	0.6089 (16)	0.5910 (7)	0.5466 (18)
(13)	**0.8466 (13)**	0.6171 (17)	0.5707 (18)	0.5222 (2)	0.5110 (3)
(14)	**0.6052 (8)**	0.6015 (10)	0.5544 (4)	0.5343 (14)	0.4442 (18)
(15)	**0.6889 (11)**	0.6230 (15)	0.5780 (18)	0.5412 (16)	0.5365 (6)
(16)	**0.6103 (17)**	0.5779 (14)	0.5513 (11)	0.5318 (16)	0.4641 (12)
(17)	**0.9775 (17)**	0.6910 (7)	0.5703 (11)	0.5184 (15)	0.4875 (16)
(18)	**0.9111 (10)**	0.7270 (2)	0.7061 (18)	0.6331 (8)	0.6168 (6)
(19)	**0.7510 (10)**	0.7017 (2)	0.6445 (14)	0.6291 (19)	0.6227 (13)
(20)	**0.9723 (20)**	0.6321 (7)	0.5112 (17)	0.5057 (11)	0.4852 (3)

Table 6.8. The best five matches with CL multiwavelet, repeated row preprocessing

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.6402 (8)**	0.6128 (18)	0.6109 (19)	0.5478 (1)	0.6128 (18)
(2)	**0.9024 (2)**	0.7361 (8)	0.6816 (7)	0.6779 (17)	0.6515 (16)
(3)	**0.7241 (12)**	0.5810 (1)	0.5005 (16)	0.4829 (2)	0.4814 (19)
(4)	**0.9835 (4)**	0.6442 (10)	0.6394 (5)	0.5780 (19)	0.5747 (20)
(5)	**0.9639 (5)**	0.5511 (11)	0.5110 (14)	0.4723 (1)	0.4626 (9)
(6)	**0.6927 (6)**	0.6165 (1)	0.5652 (5)	0.5584 (8)	0.5327 (13)
(7)	**0.7519 (7)**	0.7512 (12)	0.6194 (16)	0.5795 (14)	0.5592 (15)
(8)	**0.9669 (8)**	0.7321 (2)	0.6357 (9)	0.6043 (12)	0.5478 (7)
(9)	**0.7916 (9)**	0.7729 (12)	0.7527 (8)	0.6834 (5)	0.6581 (6)
(10)	**0.6897 (7)**	0.6496 (14)	0.6299 (3)	0.5906 (16)	0.5841 (10)
(11)	**0.8237 (11)**	0.7881 (14)	0.7839 (20)	0.7330 (16)	0.5956 (12)
(12)	**0.8517 (12)**	0.7435 (7)	0.5566 (15)	0.5538 (16)	0.5200 (10)
(13)	**0.9382 (13)**	0.6665 (16)	0.6599 (10)	0.6001 (15)	0.5899 (20)
(14)	**0.6687 (15)**	0.6021 (8)	0.5663 (14)	0.5379 (11)	0.5139 (16)
(15)	**0.7779 (10)**	0.7234 (15)	0.5876 (16)	0.5750 (4)	0.5406 (14)
(16)	**0.8815 (16)**	0.7041 (4)	0.6895 (14)	0.6863 (20)	0.5934 (10)
(17)	**0.9702 (17)**	0.6775 (8)	0.6025 (1)	0.4768 (18)	0.4688 (9)
(18)	**0.7441 (18)**	0.6348 (5)	0.5991 (9)	0.5696 (8)	0.5420 (2)
(19)	**0.7577 (19)**	0.6378 (8)	0.6082 (18)	0.5877 (5)	0.5467 (2)
(20)	**0.9837 (20)**	0.6566 (16)	0.5682 (17)	0.4897 (14)	0.4836 (1)

Table 6.9. The best five matches with SA4 multiwavelet, repeated row preprocessing

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.7055 (18)**	0.5770 (1)	0.5688 (3)	0.5061 (2)	0.4583 (5)
(2)	**0.9828 (2)**	0.8690 (8)	0.7832 (18)	0.7293 (1)	0.6309 (17)
(3)	**0.7449 (1)**	0.7038 (12)	0.6335 (14)	0.6071 (8)	0.5615 (9)
(4)	**0.9630 (4)**	0.6364 (13)	0.6212 (14)	0.5885 (17)	0.5824 (15)
(5)	**0.7746 (5)**	0.6395 (1)	0.6123 (8)	0.6050 (2)	0.5631 (12)
(6)	**0.7657 (1)**	0.6339 (12)	0.6081 (15)	0.6002 (17)	0.5786 (14)
(7)	**0.8901 (7)**	0.8866 (20)	0.7360 (12)	0.6219 (18)	0.6183 (17)
(8)	**0.9423 (8)**	0.8161 (2)	0.7508 (1)	0.6269 (10)	0.6197 (12)
(9)	**0.9489 (9)**	0.8071 (1)	0.6569 (17)	0.6542 (12)	0.5922 (18)
(10)	**0.8119 (10)**	0.7993 (1)	0.7366 (2)	0.7024 (17)	0.6665 (14)
(11)	**0.7239 (11)**	0.6726 (12)	0.6613 (14)	0.6597 (1)	0.6025 (2)
(12)	**0.8410 (2)**	0.8199 (7)	0.6946 (12)	0.6389 (9)	0.6032 (3)
(13)	**0.9366 (13)**	0.7498 (14)	0.7094 (10)	0.6529 (12)	0.6355 (3)
(14)	**0.8052 (1)**	0.7646 (10)	0.6502 (12)	0.6249 (17)	0.6068 (14)
(15)	**0.8461 (15)**	0.6849 (4)	0.6559 (16)	0.6529 (13)	0.6504 (14)
(16)	**0.9520 (16)**	0.7920 (15)	0.6921 (12)	0.6295 (14)	0.6087 (11)
(17)	**0.9797 (17)**	0.8265 (12)	0.7540 (7)	0.7355 (15)	0.7315 (20)
(18)	**0.9421 (18)**	0.6978 (1)	0.6415 (17)	0.5868 (7)	0.4983 (5)
(19)	**0.7111 (18)**	0.6015 (12)	0.5505 (19)	0.5372 (10)	0.4896 (5)
(20)	**0.9862 (20)**	0.8091 (17)	0.8009 (7)	0.6967 (14)	0.6348 (12)

Table 6.10. The best five matches with BIH52S multiwavelet, repeated row preprocessing

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.8182 (3)**	0.7562 (6)	0.6179 (15)	0.5754 (12)	0.5647 (14)
(2)	**0.8147 (2)**	0.7191 (4)	0.6970 (9)	0.6818 (10)	0.6289 (1)
(3)	**0.6906 (12)**	0.6839 (11)	0.6782 (15)	0.6653 (3)	0.5953 (13)
(4)	**0.9625 (4)**	0.7245 (3)	0.6614 (12)	0.6465 (5)	0.6464 (1)
(5)	**0.7888 (5)**	0.6941 (12)	0.6109 (7)	0.6023 (10)	0.5984 (20)
(6)	**0.6630 (3)**	0.6078 (14)	0.6075 (20)	0.5832 (15)	0.5216 (8)
(7)	**0.9356 (7)**	0.8948 (20)	0.8682 (4)	0.8435 (6)	0.7549 (9)
(8)	**0.8896 (8)**	0.7972 (9)	0.7908 (4)	0.6828 (10)	0.6730 (1)
(9)	**0.7613 (3)**	0.6398 (16)	0.5889 (15)	0.5770 (10)	0.5631 (6)
(10)	**0.9044 (6)**	0.7968 (15)	0.7904 (10)	0.7384 (9)	0.7160 (3)
(11)	**0.7888 (11)**	0.7304 (9)	0.6618 (6)	0.6571 (10)	0.6567 (4)
(12)	**0.8566 (6)**	0.7934 (4)	0.7321 (12)	0.7267 (3)	0.7252 (7)
(13)	**0.8390 (13)**	0.7853 (9)	0.7795 (6)	0.7240 (1)	0.7020 (4)
(14)	**0.8787 (4)**	0.7053 (1)	0.5952 (6)	0.5715 (10)	0.5714 (8)
(15)	**0.9228 (15)**	0.7647 (3)	0.7564 (6)	0.7093 (7)	0.6894 (14)
(16)	**0.9014 (16)**	0.8558 (14)	0.6407 (7)	0.6145 (1)	0.6121 (9)
(17)	**0.9868 (17)**	0.8004 (4)	0.7841 (1)	0.6480 (6)	0.5925 (8)
(18)	**0.7955 (18)**	0.7428 (9)	0.6923 (6)	0.6333 (3)	0.6330 (20)
(19)	**0.7339 (14)**	0.6873 (1)	0.6749 (8)	0.6577 (19)	0.6420 (4)
(20)	**0.9632 (20)**	0.8870 (4)	0.7904 (7)	0.7718 (10)	0.7636 (1)

Table 6.11. The best five matches with BIGHM2 multiwavelet, repeated row preprocessing

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.7742 (17)**	0.7087 (1)	0.6605 (11)	0.5785 (12)	0.5507 (9)
(2)	**0.9043 (2)**	0.6893 (8)	0.6351 (1)	0.6167 (16)	0.5944 (9)
(3)	**0.7316 (5)**	0.6706 (3)	0.6346 (14)	0.6314 (12)	0.6160 (10)
(4)	**0.8691 (4)**	0.7571 (6)	0.7553 (7)	0.7433 (20)	0.7157 (18)
(5)	**0.7162 (19)**	0.6389 (3)	0.6186 (15)	0.5381 (10)	0.5063 (18)
(6)	**0.6931 (17)**	0.6088 (11)	0.5489 (7)	0.5247 (20)	0.5068 (12)
(7)	**0.9459 (20)**	0.9369 (7)	0.8929 (5)	0.8427 (12)	0.8149 (17)
(8)	**0.8542 (8)**	0.8265 (2)	0.7077 (1)	0.6912 (6)	0.6701 (15)
(9)	**0.8377 (7)**	0.7688 (20)	0.7542 (1)	0.6896 (6)	0.6540 (6)
(10)	**0.7208 (19)**	0.6207 (12)	0.5677 (10)	0.5487 (13)	0.4647 (11)
(11)	**0.8202 (12)**	0.7971 (5)	0.6997 (18)	0.6716 (1)	0.6237 (20)
(12)	**0.7906 (12)**	0.5649 (20)	0.5333 (8)	0.5200 (10)	0.4933 (14)
(13)	**0.9251 (5)**	0.6667 (7)	0.5906 (17)	0.5894 (18)	0.5865 (20)
(14)	**0.8131 (1)**	0.8076 (20)	0.7989 (7)	0.7907 (5)	0.7802 (9)
(15)	**0.8488 (15)**	0.7679 (6)	0.6788 (18)	0.6765 (5)	0.6711 (10)
(16)	**0.6676 (5)**	0.5438 (10)	0.5355 (14)	0.5284 (17)	0.5173 (7)
(17)	**0.9512 (17)**	0.9254 (12)	0.9039 (7)	0.9023 (20)	0.7403 (5)
(18)	**0.8072 (17)**	0.7927 (20)	0.7766 (1)	0.7110 (7)	0.6897 (12)
(19)	**0.8003 (17)**	0.6713 (19)	0.5701 (14)	0.5644 (5)	0.4821 (11)
(20)	**0.9787 (20)**	0.9440 (7)	0.9431 (17)	0.9118 (12)	0.7003 (19)

GHM multiwavelet has the maximum number of correct matches, which is 16. The two other orthogonal multi-filter banks, CL and SA4 have correctly identified 15 and 14 of the objects, respectively. On the other hand, biorthogonal multi-filter banks BIH52S and BIGHM2 have displayed poorer performances with 12 and 7 correctly identified objects.

Plane 3, for which, the starting point could not be found, is never recognized as expected showing how important the point matching step is. Plane 1, which is hard to recognize, since the point matching step is not very successful in finding the affine transforms of the points on the original curve, has been identified successfully by GHM multiwavelets. SA4 multiwavelets although less successful on the average than the GHM and CL multiwavelets, has recognized Plane 10. Recognition of Plane 10 is difficult due to the fact that the point from which it is viewed masks its distinguishing traits, such as small protruding parts. (See Figure 6.12.) The rest of the incorrect matches is due to the specific characteristics of each multiwavelet system.

Multi-filter banks used in the simulations possess one more characteristic, which is their smoothness. When S is the lower bound of Sobolev smoothness, the multiwavelets GHM, CL, SA4, BIH52S and BIGHM2 have S values of 1.4999, 1.0545, 0.9920, 0.8280, and 0.4999, respectively. Looking at these values, it may be said that the multiwavelet, which is the smoothest, has the highest number of correct matches. Such an observation is actually in compliance with the idea of trying to avoid the finer wavelet and multiwavelet scales when obtaining the affine invariant function, since those scales are not very smooth.

6.5. The Combined Wavelet Affine Invariant Function

Although the GHM multiwavelet identifies 16 of the objects correctly and this is a better performance than that of the Mallat-Zhong wavelet, which could recognize 14 of the objects, since the computational complexity of the multiwavelet transform is greater compared to the wavelet transform, they should not be rated according to these numbers. Instead, a new affine invariant function, which uses wavelet coefficients

of both Mallat-Zhong and Daubechies-4 wavelets is formed. For this purpose, the coarsest three scales of both wavelet coefficient sets have been used. Since there is not a complete overlap between the objects that could not be recognized by the two filters, their combined performance is expected to be better than when they are used on their own.

Looking at Table 6.12, it may be said that most of the objects that could not be recognized were not identified with the individual wavelets, either. These are Planes 1, 3, 10, and 14. Plane 11, which could not be correctly matched when Daubechies-4 wavelet was solely used, is not matched correctly with the wavelet combination. There has been an incorrect matching of Plane 12, which did not exist before. On the other hand, wavelet combination has identified Planes 6, 7, 18, and 19, which could not be recognized with either one of the individual wavelets.

6.6. Statistical Comparison of All Methods

The performance of the original affine invariant function has been tested by using 20 test images, which are affine transforms of images in a database of 20. By comparing the affine invariant function obtained from the unknown object with the reference ones via correlation, the identity of the unknown object is found. However, this type of a test does not show the real recognition performance of the function, since only one view of each object is not sufficient to delineate what may happen when different poses of the objects are encountered.

For this purpose, the boundary of each of object in Figure 6.1 has been subjected to the transformation

$$\begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} 2 & b \\ 0 & 1/2 \end{bmatrix}, \tag{6.4}$$

where $\theta \in \{0^o, 36^o, 72^o, \ldots, 324^o\}$ and $b \in \{-2, -3/2, -1, \ldots, 3/2, 2, 5/2\}$. This makes a total of 100 potential views for each object.

Table 6.12. The best five matches with wavelet combination

Affine image	Best Match				
	1	2	3	4	5
(1)	**0.8575 (10)**	0.8552 (8)	0.8484 (18)	0.8294 (19)	0.7959 (20)
(2)	**0.9602 (2)**	0.9119 (8)	0.8738 (10)	0.8608 (9)	0.8539 (6)
(3)	**0.9314 (13)**	0.9143 (16)	0.8665 (15)	0.8660 (4)	0.8309 (14)
(4)	**0.9931 (4)**	0.9373 (11)	0.9238 (16)	0.8978 (10)	0.8978 (10)
(5)	**0.9657 (5)**	0.8986 (14)	0.8692 (13)	0.8652 (11)	0.8618 (10)
(6)	**0.9666 (6)**	0.9195 (10)	0.8488 (3)	0.8409 (2)	0.8268 (8)
(7)	**0.9977 (7)**	0.9909 (20)	0.9855 (12)	0.9744 (17)	0.9034 (16)
(8)	**0.9876 (8)**	0.9423 (10)	0.9025 (2)	0.8849 (13)	0.8249 (13)
(9)	**0.9523 (9)**	0.9221 (10)	0.8825 (6)	0.8526 (12)	0.8390 (8)
(10)	**0.9531 (7)**	0.9387 (12)	0.9387 (20)	0.9240 (11)	0.9083 (16)
(11)	**0.9844 (20)**	0.9717 (11)	0.9625 (12)	0.9440 (17)	0.9357 (16)
(12)	**0.9857 (7)**	0.9826 (12)	0.9736 (17)	0.9634 (20)	0.8973 (11)
(13)	**0.9911 (13)**	0.9591 (20)	0.9533 (7)	0.9382 (12)	0.9248 (11)
(14)	**0.9630 (12)**	0.9623 (13)	0.9593 (7)	0.9432 (20)	0.9213 (11)
(15)	**0.9512 (15)**	0.9354 (14)	0.9268 (4)	0.9266 (13)	0.9264 (11)
(16)	**0.9872 (16)**	0.9649 (12)	0.9606 (20)	0.9593 (7)	0.9511 (11)
(17)	**0.9976 (17)**	0.9785 (20)	0.9772 (7)	0.9668 (12)	0.9220 (16)
(18)	**0.9724 (18)**	0.9568 (19)	0.7948 (8)	0.7882 (9)	0.7828 (1)
(19)	**0.9476 (19)**	0.9286 (18)	0.8128 (8)	0.7800 (9)	0.7115 (1)
(20)	**0.9984 (20)**	0.9898 (7)	0.9857 (17)	0.9633 (12)	0.9052 (13)

To simulate the fact that the point matching step is not perfect, noise, which makes Signal-to-Noise Ratio (SNR) 40dB, has been added to the affine transformed boundaries. This is the maximum amount of noise that can be tolerated, since if the noise power increases any further, the shape cannot be recognized as that of an airplane anymore. Figure 6.23 demonstrates this fact, where the plot on the left is the affine transformed boundary of an object and the one on the right is the same boundary having an SNR of 30dB.

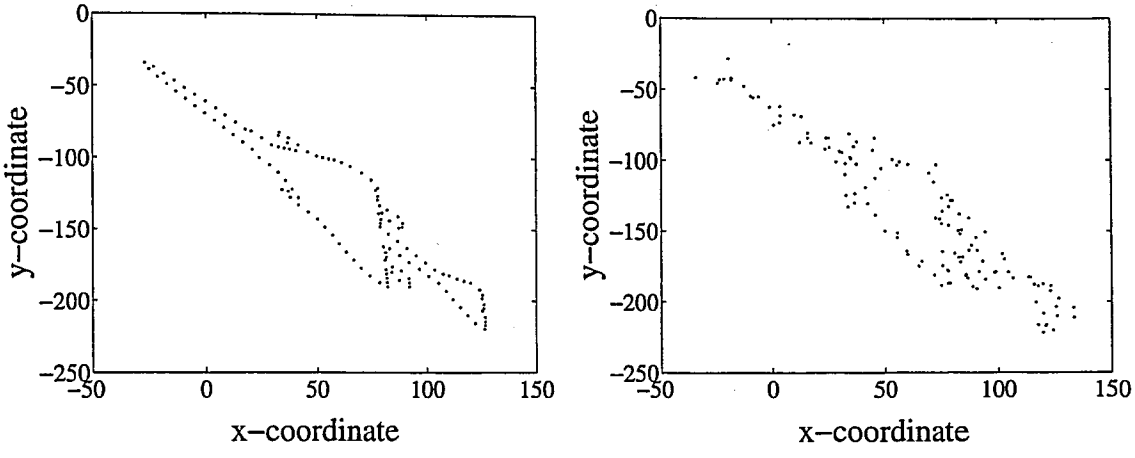

Figure 6.23. Boundary of an object, the same boundary with noise added

The definition of SNR used in the simulations is:

$$SNR = 10 \log_{10} \frac{\text{var}\{signal\}}{\text{var}\{noise\}}. \tag{6.5}$$

The methods tested in this fashion are:

- Original affine invariant function in [2],
- Modified affine invariant function with Mallat-Zhong wavelet,
- Modified affine invariant function with Daubechies-4 wavelet,
- Affine invariant multiwavelet function with GHM,
- Affine invariant multiwavelet function with CL,
- Affine invariant multiwavelet function with SA4,
- Affine invariant function using a combination of wavelet coefficients resulting from Mallat-Zhong and Daubechies-4 filters.

Table 6.13. Number of correctly matched poses of objects for each method

Plane	Number of correct matches						
	Original	Mallat-Zhong	Daubechies-4	GHM	CL	SA4	Combination
1	7	100	94	100	100	99	100
2	0	100	73	100	100	100	99
3	1	95	57	98	99	98	94
4	0	58	38	79	80	72	84
5	0	99	88	99	100	99	100
6	3	100	97	100	100	98	100
7	36	71	4	85	78	74	93
8	27	100	96	100	100	100	100
9	1	87	66	71	96	87	82
10	0	96	73	97	95	91	100
11	11	62	22	67	86	60	90
12	0	77	74	99	93	90	98
13	0	98	50	96	98	100	99
14	7	97	56	97	99	99	96
15	0	92	76	100	99	100	100
16	0	67	34	92	92	99	99
17	0	77	25	97	100	98	98
18	0	100	84	99	99	100	100
19	10	98	75	100	99	97	99
20	0	53	13	84	54	89	100

Among the seven methods tested, the method combining wavelet coefficients obtained by using different filters has shown the best performance when looked at the least number of poses recognized. Its poorest recognition rate is 82 poses out of a hundred for Plane 9, whose 96 poses have been correctly identified with the CL multiwavelet.

The affine invariant function in its original form as proposed in [2] is unsuccessful by having recognized none of the poses for 11 of the objects. For the rest of the objects, the maximum number of poses identified correctly is 36 for Plane 7. Ironically, the other methods have shown poor performance for this object.

When the modified affine invariant function is used with Mallat-Zhong filters, the minimum number of poses identified correctly is as low as 53 for Plane 20. The modified invariant function using Daubechies-4 filters has a much poorer performance compared to the other methods except the original invariant function method. It is capable of recognizing only 4 poses of Plane 7.

The multiwavelet affine invariant function has been using the GHM, CL and SA4 multiwavelets in this experiment. Although the lowest number of poses identified correctly is 67, 54, and 60 for the GHM, CL and SA4 multiwavelets, their overall performance was acceptable.

Table 6.14. Average number of correctly matched poses for each method

	Number of correct matches						
	Original	Mallat-Zhong	Daubechies-4	GHM	CL	SA4	Combination
Average	5	86	60	93	93	93	97

The performances of different methods may be compared by looking at the average number of poses that have been identified correctly. This information is given in Table 6.14. The wavelet combination method outperforms the other ones, and the

original method in [2] has recognized on the average 5 of the poses for each object. Among the scalar wavelets, Mallat-Zhong filters should be preferred over Daubechies-4 filters. The overall performance of Mallat-Zhong filters is only a little worse than those of multiwavelets, which have very similar performances.

6.7. Computational Complexity

Another way to compare the different methods is to look at their computational complexities. When implementing the scalar and multi wavelet transforms, Fast Fourier Transform (FFT) has been used instead of convolution. An N-point FFT requires $Nlog_2N$ operations, where N is the length of the signal whose FFT is computed. The convolution operation corresponds to a multiplication in the frequency domain of the FFT's of the signals that are convolved. Therefore, to implement convolution, the FFT's of signals are found, multiplied, and the product is transformed back to the time domain via Inverse Fast Fourier Transform (IFFT), which also has a complexity of $Nlog_2N$. Since N-point FFT's of the low and high pass filters have to be obtained as well, the computational complexity when computing scaling-wavelet and multiscaling-multiwavelet coefficients at any one scale depending on the signal length N is found as

Scalar wavelets $Nlog_2N \times 5$

$Nlog_2N \rightarrow$ FFT of the signal

$2 \times Nlog_2N \rightarrow$ FFT of the low and high pass filters

$2 \times Nlog_2N \rightarrow$ IFFT to compute the scaling and wavelet coefficients

Multiwavelets $3 \times Nlog_2N + 2 \times Nlog_2N \times R^2$

$Nlog_2N \times R \rightarrow$ FFT of the signal

$2 \times Nlog_2N \times R^2 \rightarrow$ FFT of the low and high pass filters

$2 \times Nlog_2N \times R \rightarrow$ IFFT to compute the multiscaling and multiwavelet coefficients

When $R = 2$, the computational complexity of the multiwavelet transform is $14 \times Nlog_2N$. This is much bigger than $5 \times Nlog_2N$ for the scalar wavelet case. However, this should be expected, since the amount of data has doubled and a matrix

filter of size 2×2 is composed of 4 scalar filters. The wavelet combination method has a computational complexity of $10 \times Nlog_2N$, for it uses two different scalar wavelets.

Taking the performances of various methods into account, it may be wise to use the wavelet combination method in an object recognition application, which has the medium level of computational complexity and the best performance among the methods tested.

6.8. Comparison with a Classical Method: Fourier Descriptors

For the sake of completeness, a comparison between the affine invariant wavelet function method and a classical method has to be done. Fourier descriptors have first been developed in the late 1960s for representation of curves. The work in [5] introduces affine invariant Fourier descriptors for recognition of objects, the methodology of which was explained in Chapter 1 of the thesis. An invariant set of coefficients may be obtained after the boundary of the object is parametrized by area if the Fourier transform of the curve is taken as in [5]. If the Fourier coefficients of the coordinate functions in such a case can be denoted by X_k and Y_k, a relative invariant is

$$det \begin{pmatrix} X_k & X_p \\ Y_k & Y_p \end{pmatrix}, \tag{6.6}$$

since this will be related to its affine counterpart with a scale factor of $\det A$ [5]. It can be made absolute by dividing with another relative function composed of different frequency indexed terms.

As has been mentioned before, this method is originally based on statistical pattern recognition techniques so that a clustering of the invariant set is necessary. If, for instance, the absolute invariant number calculated from the affine invariant Fourier coefficients of an unknown object falls into a predetermined region, its identity may be determined as the one to whom this region belongs. To be able to compare this method with the affine invariant wavelet function, however, instead of this, a correlation based

matching is done, where sets of absolute affine invariant numbers are calculated from the unknown and reference objects, which are correlated. The first 40 Fourier coefficients except the DC term is used in computation of the set and the one with the biggest magnitude is employed in normalization of the set.

Fourier descriptors when used in this fashion recognize 9 of the objects in Figure 6.2, which are Planes 2, 3, 5, 7, 9, 11, 14, 15, 18. All of the correlation values are well above 0.95, so the correlation values are not very discriminating in this case. This may be expected, since the function composed of a set of Fourier coefficients is a decreasing function of the frequency index for all objects. Thus, it has a similar shape for all. When a correlation analysis is employed, Fourier descriptors show a poorer performance compared to the affine invariant wavelet function. Each method should in fact be tested in its own setting.

7. CONCLUSIONS

Implementation of the affine invariant wavelet function proposed in [2] for recognition of planar objects in 3D with the multiwavelet transform has been the basic goal of this study. Before accomplishing this, however, the affine invariant function had to be modified in some ways so that it became suitable for a recognition system that uses real data. The affine invariant function when using multiwavelet coefficients has a better performance compared to the case when scalar wavelets are employed. If the wavelet coefficients obtained by using two different set of scalar filters are combined, the performance improves further.

The original function proposed in [2] assumes that point correspondence between the boundaries of shapes extracted from two different images of the same object, which are taken from different view points has been established before the computation of the functions. Obtaining point correspondence is a difficult task by itself. To be able to test the function with real data, point correspondence is obtained by using the area parametrization in [5] as explained in Chapter 4.

A database of 20 airplane images and affine transformed versions of these 20 images constitute the test set. The original affine invariant function, the modified affine invariant function using both Mallat-Zhong and Daubechies-4 filters, the multiwavelet affine invariant function using GHM, CL, SA4, BIH52S, and BIGHM2 multiwavelets have been tested. In this setting, the orthogonal multiwavelets; GHM, CL, and SA4, outperform the scalar wavelets while the original function does not work very well with real data recognizing only one of the objects. The scalar Mallat-Zhong wavelet, on the other hand, may be preferred over the biorthogonal multiwavelets, BIH52S, and BIGHM2. A classical method, Fourier descriptors, too, has been tested in this setting, where it has been outperformed by both scalar and multiwavelets.

Using only one pose of each object in a database of 20 may not be sufficient when comparing performances of different methods. For this purpose, 100 poses of each

object shown in its canonical view are produced by affine transforming its boundary. Random noise is added to the boundaries to simulate imperfect point matching. When the affine invariant functions are tested in this fashion, the invariant function composed of combined wavelet coefficients has very high recognition rates for all objects. The orthogonal multiwavelets have similar performances, which can be ranked the second best among all of the methods. Daubechies-4 orthogonal wavelets are outperformed by the biorthogonal filters Mallat-Zhong, whose performance is very little worse than those of the multiwavelets. When the computational complexities are also taken into account, the combination of wavelet coefficients method seems to be the best one.

7.1. Future Work

As a future work, an algorithm from the computer vision literature that restores point correspondence between two curves may be incorporated to the recognition system using the affine invariant function for matching objects. In addition, instead of using detail signals, which are the wavelet coefficients, wavelet and multiwavelet packets may be employed to choose the most discriminating features of curves.

REFERENCES

1. Weiss, I., "Geometric Invariants and Object Recognition", *International Journal of Computer Vision*, Vol.10, no.3, pp. 207-231, 1993.

2. Khalil, M. I. and M. M. Bayoumi, "A Dyadic Wavelet Affine Invariant Function for 2D Shape Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 10, pp. 1152-1164, October 2001.

3. Reiss T. H., "The Revised Fundamental Theorem of Moment Invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 8, pp. 830-834, August 1991.

4. Alferez R. and Y. F. Wang, "Geometric and Illumination Invariants for Object Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 6, pp. 505-536, June 1999.

5. Arbter K., W. E. Synder, H. Burkhardt and G. Hirzinger, "Application of Affine-Invariant Fourier Descriptors to 3-D Objects ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, pp. 640-647, July 1990.

6. Tieng Q. M. and W. W. Boles, "Wavelet Based Affine Invariant Representation: A Tool for Recognizing Planar Objects in 3D Space ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 8, pp. 846-857, August 1997.

7. Arbter K. , *Affinvariante Fourierdeskriptoren ebener Kurven*, Ph.D. Thesis, Technische Universität Hamburg, 1990.

8. Mallat S. and S. Zhong, "Characterization of Signals from Multiscale Edges", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 7, July 1992.

9. Burrus C. S., R. A. Gopinath and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentice-Hall, New Jersey, 1998.

10. Mallat S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, July 1989.

11. Strela V. , *Multiwavelets: Theory and Applications*, Ph.D. Thesis, Massachusetts Institute of Technology, 1996.

12. Strela V., W. F. Fitzgerald, R. L. Smith, A. T. Walden and P. C. Young, "Signal and Image Denoising via Wavelet Thresholding: Orthogonal and Biorthogonal, Scalar and Multiple Wavelet Transforms", *Nonlinear and Nonstationary Signal Processing*, Cambridge University Press, 2001.

13. Geronimo J., D. Hardin and P. R. Massopust, "Fractal functions and wavelet expansions based on several functions", *Journal of Approximation Theory*, Vol. 78, pp. 373-401, 1994.

14. Springer C. E., *Geometry and Analysis of Projective Spaces*, W. H. Freeman and Company, San Francisco, 1964.

15. Forsyth D., J. L. Mundy, A. Zisserman, C. Coelho, A. Heller and C. Rothwell, "Invariant Descriptors for 3-D Object Recogntion and Pose", *IEEE Transactins on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 10, October 1991.

16. Nason G. P. and B. W. Silverman, "The Stationary Wavelet Transform and some Statistical Applications", in A. Antoniadis and G. Oppenheim (eds.), *Lecture Notes in Statistics*, Vol. 103, pp. 281-300, 1995.

17. Mallat S. G., "Zero-crossings of a Wavelet Transform", *IEEE Transactions on Information Theory*, Vol. 37, No. 4, pp. 1019-1033, July 1991.

18. Arbter K., "Affine-invariant Fourier descriptors", in *From Pixels to Features*, pp.

153-164, The Netherlands: Elsevier Science, Amsterdam, 1989.

19. Mikolajczyk K. and C. Schmid, "An Interest Point Detector", *European Conference on Computer Vision (ECCV'2002)*, Copenhagen, Denmark, 28-31 May 2002.

20. Berkner K. and P. R. Massopust, "Translation Invariant Multiwavelet Transform", *Technical Report CML TR 98-06*, Rice University, 1998.

21. Berkner K. and R. O. Wells, Jr., "Smoothness Estimates for soft-threshold denoising via translation invariant wavelet transforms", *Technical Report CML TR 98-01*, Rice University, 1998.

22. Coifman R. R. and D. L. Donoho, "Translation Invariant Denoising", in A. Antoniadis (eds.), *Wavelets and Statistics*, Springer-Verlag Lecture Notes, New York, 1995.

23. Shensa M. J., "The Discrete Wavelet Transform: Wedding the A Trous and Mallat Algorithms", *IEEE Transactions on Signal Processing*, Vol. 40, No. 10, October 1992.

24. Lebrun J. and M. Vetterli, "Balanced Multiwavelets Theory and Design", *IEEE Transactions on Signal Processing*, Vol. 46, No. 4, April 1994.