

**A WAVELET BASED METHOD FOR AFFINE INVARIANT 2D
OBJECT RECOGNITION**

by
MEHMET YAĞMUR GÖK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

July 2003

A WAVELET BASED METHOD FOR AFFINE INVARIANT 2D OBJECT
RECOGNITION

APPROVED BY

Prof. Dr. Aytül Erçil
(Thesis Supervisor)

Prof. Dr. Ahmet Enis Çetin
(Thesis Co-Supervisor)

Assist. Prof. Dr. Mehmet Keskinöz

DATE OF APPROVAL:

©Mehmet Yağmur Gök 2003
All Rights Reserved

To My Family

Acknowledgments

I gratefully thank Prof. Dr Enis etin and Prof. Dr. Aytül Eril for their supervision, guidance and suggestions throughout the development of this Thesis. I also thank to Erdem Bala and İbrahim Hokelek for their helps.

A WAVELET BASED METHOD FOR AFFINE INVARIANT 2D OBJECT RECOGNITION

Abstract

Recognizing objects that have undergone certain viewing transformations is an important problem in the field of computer vision. Most current research has focused almost exclusively on single aspects of the problem, concentrating on a few geometric transformations and distortions. Probably, the most important one is the affine transformation which may be considered as an approximation to perspective transformation. Many algorithms were developed for this purpose. Most popular ones are Fourier descriptors and moment based methods. Another powerful tool to recognize affine transformed objects, is the invariants of implicit polynomials. These three methods are usually called as traditional methods. Wavelet-based affine invariant functions are recent contributions to the solution of the problem. This method is better at recognition and more robust to noise compared to other methods. These functions mostly rely on the object contour and undecimated wavelet transform. In this thesis, a technique is developed to recognize objects undergoing a general affine transformation. Affine invariant functions are used, based on on image projections and high-pass filtered images of objects at projection angles . Decimated Wavelet Transform is used instead of undecimated Wavelet Transform. We compared our method with the an another wavelet based affine invariant function, Khalil-Bayoumi and also with traditional methods.

Özet

Görüntü dönüşümüne uğramış objeleri tanımak, bilgisayarlı görüntüleme alanındaki önemli problemlerden biridir. Son zamanlardaki birçok araştırma, özellikle geometrik dönüşümler üzerine odaklanmıştır. Bu dönüşümlerin en önemlileri kamera hareketi ile meydana gelen perspektif dönüşümü ve onun yakınsaması olan ilgin dönüşümdür. Bunun için geliştirilmiş birçok yöntem mevcuttur. Bunları en önde gelenleri Fourier tanımlayıcıları; Momentler ve Örtük polinom eğrileridir. Bu yöntemler geleneksel yöntemler olarak da adlandırılırlar. Wavelet bazlı ilgin fonksiyonlar, son zamanlarda geliştirilen yöntemlerdir. Bu yöntem diğer yöntemlere göre daha efektif ve gürültüye karşı daha etkilidir. Bu yöntemlerde objelerin çevre eğrileri ve "undecimated wavelet" dönüşüm kullanılır. Bu tezde, ilgin dönüşüme uğramış nesneleri bilgisayarla tanımak için yeni bir yöntem önerilmektedir. Bu yöntemde ilgin fonksiyonlar, görüntü projeksiyonları ve high-pass filtrelenmiş resimlerin projeksiyonları kullanılmaktadır. Ayrıca, diğer "wavelet" bazlı metodların aksine "decimated wavelet" dönüşüm tercih edilmiştir. Yöntemimizi diğer "wavelet" bazlı yöntemler olan Khalil-Baoumi metodu ile ve geleneksel yöntemlerle karşılaştırdık.

Table of Contents

Acknowledgments	v
Abstract	vi
Ozet	vii
1 Introduction	1
2 Traditional Methods	5
2.1 Implicit Polynomials	5
2.1.1 Data set normalization	6
2.1.2 3L fitting	7
2.1.3 Affine invariants	9
2.2 Fourier Descriptors	10
2.2.1 Parametrization	11
2.2.2 Construction of Parameters from Fourier Coefficients	12
2.3 Moment Invariants	14
2.3.1 Moments	14
2.3.2 Algebraic Invariants	16
2.3.3 Affine Moment Invariants	18
3 Wavelet-based Affine Invariant Functions	20
3.1 Wavelet transform	21
3.1.1 Multiresolution Analysis and Discrete Wavelet Transform	22
3.2 Tieng-Boles Function	26
3.3 Khalil-Bayoumi Function	28
3.4 Wavelet Affine Function with Image Projection	32
3.5 Experimental Results	35
4 Conclusion	43
5 Appendix	44
Bibliography	52

List of Figures

3.1	The filterbank associated with multiresolution analysis. H_h, F_h are high-pass filters and H_d, F_d are low-pass filters. In the equations, high-pass filter is used as g and low-pass filter is used as h	25
3.2	Block diagram of dyadic wavelet transform(left) and its associated inverse transform(right). H_h, F_h are high-pass filters and H_d, F_d are low-pass filters.	26
3.3	Our algorithm	33
3.4	Projection(left) and projection of the high-pass filtered(right) of airplane model 12 at 40°	34
3.5	Projection(left) and projection of airplane model 12 after high-pass filtering (right) at $0^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ$, used as input signal to wavelet transform and then affine function	34
3.6	The airplane models	37
3.7	The test images	38
3.8	Low-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method.	39
3.9	Low-noise level correlation values for our method and Implicit polynomials. Thick line corresponds to our method and thin line with square marking corresponds to Implicit polynomials.	40
3.10	High-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method.	41

3.11 High-noise level correlation values for our method and Implicit polynomials. Thick line corresponds to our method and thin line with square marking corresponds to Implicit polynomials. Arrow head shows false detection.	41
3.12 Highest-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method. Arrow head shows false detection.	42

List of Tables

2.1	the values of $g+k$ and k for the invariants	19
3.1	Model Images Used to Produce the Test Images	36
5.1	the results at low-noise level for our method	45
5.2	the results at low-noise level for Khalil-Bayoumi method	45
5.3	the results at high-noise level for our method	46
5.4	the results at high-noise level for Khalil-Bayoumi method	46
5.5	the results at highest-noise level for our method and Khalil-Bayoumi method. In second column highest correlation values for our method is shown and in third column for Khalil-Bayoumi method	47
5.6	results for Tieng-Boles function through our method, with image pro- jections	47
5.7	Low-noise level experiment results for implicit polynomials	48
5.8	High-noise level experiment results for implicit polynomials	48
5.9	the results at low-noise level for Moment method	49
5.10	the results at high-noise level for Moment method	49
5.11	the results at low-noise level for Fourier descriptors	50
5.12	the results at high-noise level for Fourier descriptors	50

**A WAVELET BASED METHOD FOR AFFINE INVARIANT 2D
OBJECT RECOGNITION**

by
MEHMET YAĞMUR GÖK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

July 2003

A WAVELET BASED METHOD FOR AFFINE INVARIANT 2D OBJECT
RECOGNITION

APPROVED BY

Prof. Dr. Aytül Erçil
(Thesis Supervisor)

Prof. Dr. Ahmet Enis Çetin
(Thesis Co-Supervisor)

Assist. Prof. Dr. Mehmet Keskinöz

DATE OF APPROVAL:

©Mehmet Yağmur Gök 2003
All Rights Reserved

To My Family

Acknowledgments

I gratefully thank Prof. Dr Enis etin and Prof. Dr. Aytül Eril for their supervision, guidance and suggestions throughout the development of this Thesis. I also thank to Erdem Bala and İbrahim Hokelek for their helps.

A WAVELET BASED METHOD FOR AFFINE INVARIANT 2D OBJECT RECOGNITION

Abstract

Recognizing objects that have undergone certain viewing transformations is an important problem in the field of computer vision. Most current research has focused almost exclusively on single aspects of the problem, concentrating on a few geometric transformations and distortions. Probably, the most important one is the affine transformation which may be considered as an approximation to perspective transformation. Many algorithms were developed for this purpose. Most popular ones are Fourier descriptors and moment based methods. Another powerful tool to recognize affine transformed objects, is the invariants of implicit polynomials. These three methods are usually called as traditional methods. Wavelet-based affine invariant functions are recent contributions to the solution of the problem. This method is better at recognition and more robust to noise compared to other methods. These functions mostly rely on the object contour and undecimated wavelet transform. In this thesis, a technique is developed to recognize objects undergoing a general affine transformation. Affine invariant functions are used, based on on image projections and high-pass filtered images of objects at projection angles . Decimated Wavelet Transform is used instead of undecimated Wavelet Transform. We compared our method with the an another wavelet based affine invariant function, Khalil-Bayoumi and also with traditional methods.

Özet

Görüntü dönüşümüne uğramış objeleri tanımak, bilgisayarlı görüntüleme alanındaki önemli problemlerden biridir. Son zamanlardaki birçok araştırma, özellikle geometrik dönüşümler üzerine odaklanmıştır. Bu dönüşümlerin en önemlileri kamera hareketi ile meydana gelen perspektif dönüşümü ve onun yakınsaması olan ilgin dönüşümdür. Bunun için geliştirilmiş birçok yöntem mevcuttur. Bunları en önde gelenleri Fourier tanımlayıcıları; Momentler ve Örtük polinom eğrileridir. Bu yöntemler geleneksel yöntemler olarak da adlandırılırlar. Wavelet bazlı ilgin fonksiyonlar, son zamanlarda geliştirilen yöntemlerdir. Bu yöntem diğer yöntemlere göre daha efektif ve gürültüye karşı daha etkilidir. Bu yöntemlerde objelerin çevre eğrileri ve "undecimated wavelet" dönüşüm kullanılır. Bu tezde, ilgin dönüşüme uğramış nesneleri bilgisayarla tanımak için yeni bir yöntem önerilmektedir. Bu yöntemde ilgin fonksiyonlar, görüntü projeksiyonları ve high-pass filtrelenmiş resimlerin projeksiyonları kullanılmaktadır. Ayrıca, diğer "wavelet" bazlı metodların aksine "decimated wavelet" dönüşüm tercih edilmiştir. Yöntemimizi diğer "wavelet" bazlı yöntemler olan Khalil-Baoumi metodu ile ve geleneksel yöntemlerle karşılaştırdık.

Table of Contents

Acknowledgments	v
Abstract	vi
Ozet	vii
1 Introduction	1
2 Traditional Methods	5
2.1 Implicit Polynomials	5
2.1.1 Data set normalization	6
2.1.2 3L fitting	7
2.1.3 Affine invariants	9
2.2 Fourier Descriptors	10
2.2.1 Parametrization	11
2.2.2 Construction of Parameters from Fourier Coefficients	12
2.3 Moment Invariants	14
2.3.1 Moments	14
2.3.2 Algebraic Invariants	16
2.3.3 Affine Moment Invariants	18
3 Wavelet-based Affine Invariant Functions	20
3.1 Wavelet transform	21
3.1.1 Multiresolution Analysis and Discrete Wavelet Transform	22
3.2 Tieng-Boles Function	26
3.3 Khalil-Bayoumi Function	28
3.4 Wavelet Affine Function with Image Projection	32
3.5 Experimental Results	35
4 Conclusion	43
5 Appendix	44
Bibliography	52

List of Figures

3.1	The filterbank associated with multiresolution analysis. H_h, F_h are high-pass filters and H_d, F_d are low-pass filters. In the equations, high-pass filter is used as g and low-pass filter is used as h	25
3.2	Block diagram of dyadic wavelet transform(left) and its associated inverse transform(right). H_h, F_h are high-pass filters and H_d, F_d are low-pass filters.	26
3.3	Our algorithm	33
3.4	Projection(left) and projection of the high-pass filtered(right) of airplane model 12 at 40°	34
3.5	Projection(left) and projection of airplane model 12 after high-pass filtering (right) at $0^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ$, used as input signal to wavelet transform and then affine function	34
3.6	The airplane models	37
3.7	The test images	38
3.8	Low-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method.	39
3.9	Low-noise level correlation values for our method and Implicit polynomials. Thick line corresponds to our method and thin line with square marking corresponds to Implicit polynomials.	40
3.10	High-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method.	41

3.11 High-noise level correlation values for our method and Implicit polynomials. Thick line corresponds to our method and thin line with square marking corresponds to Implicit polynomials. Arrow head shows false detection.	41
3.12 Highest-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method. Arrow head shows false detection.	42

List of Tables

2.1	the values of $g+k$ and k for the invariants	19
3.1	Model Images Used to Produce the Test Images	36
5.1	the results at low-noise level for our method	45
5.2	the results at low-noise level for Khalil-Bayoumi method	45
5.3	the results at high-noise level for our method	46
5.4	the results at high-noise level for Khalil-Bayoumi method	46
5.5	the results at highest-noise level for our method and Khalil-Bayoumi method. In second column highest correlation values for our method is shown and in third column for Khalil-Bayoumi method	47
5.6	results for Tieng-Boles function through our method, with image pro- jections	47
5.7	Low-noise level experiment results for implicit polynomials	48
5.8	High-noise level experiment results for implicit polynomials	48
5.9	the results at low-noise level for Moment method	49
5.10	the results at high-noise level for Moment method	49
5.11	the results at low-noise level for Fourier descriptors	50
5.12	the results at high-noise level for Fourier descriptors	50

Chapter 1

Introduction

Object recognition is an important problem in computer vision and pattern analysis. Research in computer vision is aimed at enabling computers to recognize objects without human intervention. Applications are numerous, and include automatic inspection of parts in factories, detection of fires at high-risk sites and robot vision, especially for autonomous robots. Object recognition can be described as the task of finding and labelling parts of an image that corresponds to objects in the scene. The task is usually broken up into two stages, 'low-level' vision and 'high-level' vision. Low-level vision involves extracting significant features from the image, such as the outline of an object or regions with same texture, and often involves segmenting the image into separate 'objects'. The task of high-level vision is then to recognize objects.

High-level vision, in particular is concerned with finding the properties of an image which are invariant to transformations of the image caused by moving an object so as to change its perceived position and orientation. The idea of invariance arises from our own ability to recognize objects irrespective to such movement. If one looks at a car from different orientations, it is easy for a human being to recognize it as a car; it can be said that a car has properties which are invariant to size, position and orientation. Finding mathematical functions of an image that are invariant to the above transformations provides us with techniques for recognizing objects using computers.

The search for invariants is a classical problem in mathematics dating back to the 18th century. Invariant features form a compact, intrinsic description of an object and can be used to design recognition algorithms that are potentially more

efficient than, say, aspect-based approaches. Invariant features can be designed based on many different methods. They can be computed either globally, which requires the knowledge of the shape as a whole or locally, which are based on local properties such as curvature as arc length. Global invariants suffer when some parts of the image data are unavailable. On the other hand most local invariants have difficulties tolerating noise because its computation usually involves solving for high order derivatives.

Current research has focused almost exclusively on single aspects of the problem, concentrating on a few geometric transformations and distortions. Shape distortion, arising from observing an object by a camera under arbitrary orientations, can be most appropriately described as a perspective transformation [1]. However when the dimensions of the object are small compared to the distance from the camera to the object, a weak perspective can be assumed. In this case, the orthographic projection may be used as an approximation to the perspective projection, and the perspective distortion of the object can be modelled by shear in the image plane. Furthermore, the affine transformation, consisting rotation, scaling and shearing and translation transformations may be used as an approximation to the perspective transformation [1].

Image invariants can be designed to fit the needs of specific systems. Some require that it be nondiscriminating to an object's geometric pose or orientation. Others may be interested in it being insensitive to the change of illumination. More complex systems demand it to be insensitive to a combination of several environmental changes. Furthermore, invariant features can be designed based on many different methods. It can be computed either globally, which requires shape knowledge as a whole, or locally, which are based on local properties such as curvature and arc length. When some parts of image data is unavailable, global invariants are unable to produce good results. On the other hand, most local invariants have difficulties tolerating noise since then its computation usually involves solving for high order derivatives. Most of the current studies have focused almost exclusively on single aspects of the problem, concentrating on a few geometric invariants. Affine invariants are among most popular ones.

Consider a parametric curve $x(t), y(t)$ parameterized by t on a plane. Affine

transformation performs the following mappings:

$$\tilde{x}(t) = a_0 + a_1x(t) + a_2y(t). \quad (1.1)$$

$$\tilde{y}(t) = b_0 + b_1x(t) + b_2y(t). \quad (1.2)$$

Equations (3.1) and (3.2) can be written in the matrix form as:

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = A \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + B. \quad (1.3)$$

where A is a nonsingular square matrix representing the rotation, scaling and skewing transformations. The vector B represents the translation. When Affine transformation is applied to the whole image, the coordinate system changes and Jacobean J provides the information about this coordinate change.

$$J = \left| \frac{\partial(\tilde{x}, \tilde{y})}{\partial(x, y)} \right| = \begin{vmatrix} \frac{\partial(\tilde{x})}{\partial(x)} & \frac{\partial(\tilde{x})}{\partial(y)} \\ \frac{\partial(\tilde{y})}{\partial(x)} & \frac{\partial(\tilde{y})}{\partial(y)} \end{vmatrix} = a_1b_2 - a_2b_1 = \det(A). \quad (1.4)$$

let $I(t)$ be an invariant function and $\tilde{I}(t)$ be the same invariant function calculated using the points that are subjected to affine transformation. The relation between them can be formulated as:

$$\tilde{I} = IJ^w. \quad (1.5)$$

The exponent w which is the power of Jacobean J is called the weight of the invariant. In the case ; $w = 0$ the function is called absolute invariant. If $w \neq 0$ then it is called the relative invariant.

Many algorithms have been developed for the representation of objects undergoing affine transformation. They can be classified as local and global techniques. Global techniques are based on the use of global features of the object such as the Fourier Descriptors [2],[3],[4],[5],[6] which is effective against noise and the affine moment invariants derived by Flusser and Suk [7], which are the extension of the classical moment invariants developed by Hu [8]. High order moments are sensitive to noise so only a few low-order moment invariants are used and this limits the ability of object classification with a large size database. Local techniques use local features such as critical points [9]. Another algorithm to recognize affine transformed objects(Chapter 3) is the one based on implicit polynomials. Invariant features of implicit polynomials [11]-[14] are used for that purpose based on 3L fitting algorithm and data set normalization to remove "affineness" of the data

Tieng-Boles [15] and Khalil-Bayoumi [16] derived new techniques based on dyadic wavelet transform. This technique decomposes object contours into several components at different resolution levels and uses an affine invariant function derived by [15],[16]. These techniques combine the spatial and transform domain method's advantages. In our technique we do not use the object contour but instead, the one dimensional (1D) projection of objects from various angles and high-pass filtered images of objects at these angles.

Fourier descriptors, and affine moment invariants and implicit polynomial method, which are called as traditional methods are explained and experimental results are given in Chapter 2 . In Chapter 3, wavelet based affine invariant functions together with our technique is presented. Also, experimental results comparing our method with Khalil-Bayoumi and Tieng-Boles method are presented.

Chapter 2

Traditional Methods

2.1 Implicit Polynomials

Implicit polynomials are one of the leading shape representations in computer vision . Implicit polynomials have several strong features such as their interpolation property against missing data, smoothing property against noise and perturbations, Bayesian recognizers and the most important of all may be their algebraic invariants. Implicit polynomial related techniques require to have a robust and consistent implicit polynomial fits to data sets. This problem is solved through different minimization techniques. There are various polynomial fitting techniques; but we focus on 3L fitting technique [12],[17],[18] which seems to overcome many drawbacks of the other algorithms. For curve fitting, first sensed data points of an object to be recognized, the object contour, is fit by an implicit polynomial. Then a vector of polynomial coefficients is used to obtain the invariants which are used in object recognition. An implicit polynomial model in 2D, with an implicit curve of degree n , is defined by :

$$f(x, y) = \sum_{0 < i, j; i+j < n} a_{ij} x^i y^j = \underbrace{a_{00}}_{H_0} + \underbrace{a_{10}x + a_{01}y}_{H_1(x,y)} + \underbrace{a_{20}x^2 + a_{11}xy + a_{02}y^2}_{H_2(x,y)} + \dots$$
$$+ \underbrace{a_{n0}x^n + a_{n-1,1}x^{n-1}y + \dots + a_{0n}y^n}_{H_n(x,y)} = \sum_{r=0}^n H_r(x, y) = 0.$$
(2.1)

where $H_r(x, y)$ is a homogeneous binary (*i.e, two variables*) polynomial of degree r in x and y . Notice that in the above formula, the grading lexicographic monomials induced by $x_3 < x_2 < x_1$ is applied. $f(x, y)$ is written in the vector form, to facilitate the polynomial fitting, as:

$$f(x, y) = Y^T A. \quad (2.2)$$

where

$$A = [a_{00} \ a_{10} \ a_{01} \ a_{20} \ a_{11} \ a_{02} \ \dots \ a_{0n}]^T \quad (2.3)$$

and

$$Y = [1 \ x \ y \ x^2 \ xy \ y^2 \ x^3 \ \dots \ x^n \ \dots \ xy^{n-1} \ y^n]. \quad (2.4)$$

Such curves of degree 2 are the *circles, hyperbolas, ,straight line pairs, conics – ellipses*, etc. that are commonly used.

To use affine invariant property of implicit polynomials for determining the affine equivalence of two curves, we need an affine invariant fitting algorithm. The affine fitting algorithm performs fittings to the original data set and the affine transformed one, given a data set of points. 3L fitting algorithm, which is explained in section 2.1.2, is not affine invariant[17]. due to that the level set generation is based on an Euclidian invariant quantity. This problem can be solved by replacing the euclidian invariant quantity in the level set generation by an affine invariant quantity or removing the affineness of the data set by a scattering matrix normalization. We used data set normalization in our work.

2.1.1 Data set normalization

Data set normalization is used to remove the affineness of data. After this operation, also called as whitening, 3L fitting algorithm can be used without any modification [17],[18]. But this does not make the recognition process affine invariant. Extra work is required to make recognition process affine invariant. This is done via the use of invariants found by [17]. Data set normalization can be explained as follows:

The scatter matrix of a data set Σ , which is a positive symmetric matrix, can be written as :

$$\Sigma = Q \Lambda Q^T \quad (2.5)$$

where Q is an orthogonal matrix of normalized eigenvectors of Σ and Λ is the diagonal matrix of the corresponding eigenvalues. The scatter matrix of the data set becomes the identity matrix I , by applying the transformation, $A_w = \Lambda^{-1/2} Q^T$ to the data set. This transformation makes the spectrum of the eigenvectors uniform.

Assume that Γ_0 and $\widehat{\Gamma}_0$ are two data sets related by affine transformation. Mathematical transformation between them reduces to rotation after the A_w transformation is applied to both. So after this transformation we can use the 3L fitting algorithm to fit to data and then recognize affine transformed object.

2.1.2 3L fitting

To fit an implicit polynomial to the object boundary, the n_{th} degree implicit polynomial $f(x, y)$, that minimizes the average squared distance from the data points to the zero set $Z(f)$ of the polynomial, should be found. As no explicit expression is available, an iterative process is used to solve for *geometric distance*. A widely used distance approximation is:

$$d(p_i, Z(f)) \approx \frac{|f(p_i)|}{\|\nabla f(p_i)\|} \quad (2.6)$$

and the average squared distance becomes:

$$\bar{d}^2 \approx \frac{1}{N} \sum_{i=1}^N d^2(p_i, Z(f)) = \frac{1}{N} \sum_{i=1}^N \frac{|f(p_i)|^2}{\|\nabla f(p_i)\|^2} \quad (2.7)$$

This is a nonlinear optimization problem.

Usually only Γ_0 , data set of object boundary is taken into account by many fitting formulations. As it is explained in [17], " It is possible to fit the polynomial in a fast and stable way by fitting the explicit polynomial $f(x, y)$ to a portion of the distance transform $d(x, y)$ of Γ_0 . $d(x, y)$ is the function which, at (x, y) takes on the value of the *signed* distance from (x, y) to Γ_0 ; meaning that $d(x, y)$ is the shortest

distance between (x, y) to the closest point in Γ_0 and takes positive and negative values according to what side of the data set Γ_0 it is present. 3L fitting algorithm uses synthetically generated data sets Γ_{+c} and Γ_{-c} besides the data set Γ_0 . Data set Γ_{+c} contains the points at a distance c to one side of Γ_0 and Γ_{-c} to other side of Γ_0 . Γ_{+c} and Γ_{-c} are the *levelsets* of $d(x, y)$ at levels $+c$ and $-c$ ”.

We can use a distance transform computation algorithm to generate $d(x, y)$ from Γ_0 . For each data point in Γ_0 , The Euclidean distance transform determines a point in Γ_{+c} and one in Γ_{-c} . These are at a perpendicular distance c at each side of the original data set(curve) Γ_0 . Let $\Gamma_0 \cup \Gamma_{+c} \cup \Gamma_{-c} = (x_i y_i)^T \quad : 1 < i < 3K$ and

$$M = [Y_1 \ Y_2 \ \dots \ Y_{3K}]$$

where Y_i is Y at (4) evaluated at $p_i = (x_i, y_i)$. Also d is defined as a vector whose i_{th} component $d(x_i, y_i)$ which is the distance between point p_i and Γ_0 . The level sets used are only, $+c, 0$ and $-c$. The problem of estimating the vector of polynomial coefficients A becomes the minimization problem; minimize:

$$\sum_{i=1}^{3K} (d(x_i, y_i) - Y_i^T A)^2 \quad \text{or } \| MA - d \|^2$$

For this problem, the least squares solution is:

$$A = (M^T M)^{-1} M^T d. \tag{2.8}$$

Introduction of two level set constraints is because of two reasons. First reason is to have a more stable and consistent fitting with regard to the transformations of the data set Γ_0 and being more robust to noise. Fitting the polynomial to more than Γ_0 ; fitting $f(x, y)$ to a ribbon of data rather than to a curve of data, leads us to that accomplishment. Also, singularities are removed from the vicinity of data set and forced to occur at local extrema or saddle points and singularities are prevented to occur within synthetic ribbon by the use of synthetic data sets Γ_{+c} and Γ_{-c} . Second, as the fitted polynomial $f(x, y)$ is an approximation to the distance transform

$d(x, y)$, given a new data point (\hat{x}, \hat{y}) , $|f(\hat{x}, \hat{y})|$ is an approximation to the distance between (\hat{x}, \hat{y}) to Γ_0 .

2.1.3 Affine invariants

To use implicit polynomials to recognize affine transformed 2D objects we need to obtain invariants to affine transformation. We used the invariants obtained by Civi [17]. Here, first some relative affine invariants of fourth degree implicit polynomials are given as :

$$\begin{aligned} \Gamma_1 = & 45a_{13}^2a_{20}^2 - 30a_{12}a_{13}a_{20}a_{21} + 3a_{12}^2a_{21}^2 + 6a_{11}a_{13}a_{21}^2 + 48a_{04}a_{20}a_{21}^2 - 12a_{03}a_{21}^3 + \\ & 20a_{12}^2a_{20}a_{22} - 30a_{11}a_{31}a_{20}a_{22} - 120a_{04}a_{20}^2a_{22} - 16a_{11}a_{12}a_{21}a_{22} + 54a_{10}a_{13}a_{21}a_{22} + \\ & 12a_{03}a_{20}a_{21}a_{22} + 20a_{02}a_{21}^2a_{22} + 17a_{11}^2a_{22}^2 - 36a_{10}a_{12}a_{22}^2 - 8a_{02}a_{20}a_{22}^2 - 36a_{01}a_{21}a_{22}^2 + \\ & 72a_{00}a_{22}^3 - 12a_{12}^3a_{30} + 54a_{11}a_{12}a_{13}a_{30} - 162a_{10}a_{13}^2a_{30} - 72a_{04}a_{12}a_{20}a_{30} + 54a_{03}a_{13}a_{20}a_{13}a_{30} - \\ & 72a_{04}a_{11}a_{21}a_{30} + 54a_{03}a_{12}a_{21}a_{30} - 72a_{02}a_{13}a_{21}a_{30} + 432a_{04}a_{10}a_{22}a_{30} - 72a_{03}a_{11}a_{22}a_{30} + \\ & 12a_{02}a_{12}a_{22}a_{30} + 54a_{01}a_{13}a_{22}a_{30} - 81a_{03}^2a_{30}^2 + 216a_{02}a_{04}a_{30}^2 + 6a_{11}a_{12}^2a_{31} - 36a_{11}^2a_{13}a_{31} + \\ & 54a_{10}a_{12}a_{13}a_{31} + 180a_{04}a_{11}a_{20}a_{31} - 72a_{03}a_{12}a_{20}a_{31} + 54a_{02}a_{13}a_{20}a_{31} - 324a_{04}a_{10}a_{21}a_{31} + \\ & 54a_{03}a_{11}a_{21}a_{31} - 30a_{02}a_{12}a_{21}a_{31} + 54a_{01}a_{13}a_{21}a_{31} + 54a_{03}a_{10}a_{22}a_{31} - 30a_{02}a_{11}a_{22}a_{31} + \\ & 54a_{01}a_{12}a_{22}a_{31} - 324a_{00}a_{13}a_{22}a_{31} + 54a_{02}a_{03}a_{30}a_{31} - 324a_{01}a_{04}a_{30}a_{31} + 45a_{02}^2a_{31}^2 - \\ & 162a_{01}a_{03}a_{31}^2 + 972a_{00}a_{04}a_{31}^2 - 36a_{04}a_{11}^2a_{40} + 432a_{04}a_{10}a_{12}a_{40} - 72a_{03}a_{11}a_{12}a_{40} + \\ & 48a_{02}a_{12}^2a_{40} - 324a_{03}a_{10}a_{13}a_{40} + 180a_{02}a_{11}a_{13}a_{40} - 324a_{01}a_{12}a_{13}a_{40} + 972a_{00}a_{13}^2a_{40} + \\ & 216a_{03}a_{20}a_{40} - 576a_{02}a_{04}a_{20}a_{40} - 72a_{02}a_{03}a_{21}a_{40} - 120a_{02}^2a_{22}a_{40} + 432a_{01}a_{03}a_{22}a_{40} - \\ & 2592a_{00}a_{04}a_{22}a_{40} \end{aligned}$$

$$\begin{aligned} \Gamma_2 = & 144a_{40}a_{04}a_{00} - 36a_{40}a_{03}a_{01} + 12a_{40}a_{02}a_{02} - 36a_{31}a_{13}a_{00} + 9a_{31}a_{12}a_{01} - 6a_{31}a_{11}a_{02} + \\ & 9a_{31}a_{10}a_{03} + 9a_{30}a_{13}a_{01} - 6a_{30}a_{12}a_{02} + 9a_{30}a_{11}a_{03} - 36a_{30}a_{10}a_{04} + 12a_{22}a_{22}a_{00} - 6a_{22}a_{21}a_{01} + \\ & 4a_{22}a_{20}a_{02} - 6a_{22}a_{12}a_{10} + 2a_{22}a_{11}a_{11} + 2a_{21}a_{21}a_{02} - 6a_{21}a_{20}a_{03} + 9a_{21}a_{13}a_{10} - a_{21}a_{12}a_{11} + \\ & 12a_{20}a_{20}a_{04} - 6a_{20}a_{13}a_{11} + 2a_{20}a_{12}a_{12} \end{aligned}$$

$$\Gamma_3 = 6a_{22}a_{22}a_{22} - 27a_{13}a_{22}a_{31} + 81a_{04}a_{31}a_{31} + 81a_{13}a_{13}a_{40} - 216a_{04}a_{22}a_{40}$$

$$\Gamma_4 = 120a_{40}a_{04} - 30a_{31}a_{13} + 10a_{22}a_{22}$$

where $a_{00}, a_{01}, a_{10}, a_{20}, a_{11}, a_{02}, a_{30}, a_{21}, a_{12}, a_{03}, a_{40}, a_{31}, a_{22}, a_{13}, a_{04}$ are the implicit polynomial coefficients obtained by the affine invariant 3L fitting algorithm. In order to use an invariant in object recognition under affine transformation of the image plane, we should have an absolute weight invariant. Absolute invariants which we used are obtained through the relative invariants by [17].

$$I_1 = \frac{\Gamma_1 \Gamma_4}{\Gamma_2 \Gamma_3} \quad (2.9)$$

$$I_2 = \frac{\Gamma_1^2}{\Gamma_2^2 \Gamma_4} \quad (2.10)$$

2.2 Fourier Descriptors

Fourier descriptors provide a means for representing the boundary of a two dimensional shape. The basic idea is this: a closed curve may be represented by a periodic function of a continuous parameter, or alternatively, by a set of Fourier coefficients of this function. These coefficients are called Fourier Descriptors. In order to use Fourier descriptors for pattern classification applications, we must normalize the curve representation with respect to a desired transformation class. If the normalization is exact it will result in a set of Fourier descriptors which are invariant with respect to the desired transformation class.

The early similarity-invariant Fourier Descriptors were derived by normalization performed in the spatial domain, using the invariant properties of curvature and/or tangent angle. The calculation of these quantities implies the calculation of derivatives, which may be avoided by performing normalization completely in the Fourier domain. Class of Fourier transforms includes the similarity transforms, but in addition includes shearing.

Affine invariant Fourier descriptors were introduced by Arbter *et al.* [3]. Fourier descriptors were originally introduced to provide rotation invariance: if one has a closed contour described by $(x(s), y(s))$, $s \in S$. Then the curve can be approximated by a Fourier series with coefficients U_k, V_k defined as:

$$[U_k, V_k] = \frac{1}{S} \int_0^S [x(s), y(s)] e^{-j \frac{2\pi k s}{S}} ds. \quad (2.11)$$

the magnitudes of U_k and V_k are invariant to rotations; invariance to translations can be achieved by the coordinate origin at the image centroid, and invariance to changes in scale by forming the ratio of two coefficients. Invariance to affine transformation is not so straightforward because the curve length can change. We need a new parametrization.

2.2.1 Parametrization

The affine transform can be written as:

$$x = Ax^0 + b, \quad \det(A) \neq 0. \quad (2.12)$$

where $x, x^0 \in \mathfrak{R}^2, A$ is a 2x2 matrix, b is a 2-vector and x is the affine transformed version of x^0 or using the complex representation:

$$x = ax^0 + bx^{0*} + c, \quad aa^* - bb^* \neq 0. \quad (2.13)$$

where $x, x^0, a, b, c \in C$, complex plane; c is the constant representing translation. a, b are constants due to the linear part of the affine transformation.

The arc length is nonlinearly transformed under affine transformation so a new parametrization is needed which is linear under affine transformation and the parameterizing function must yield the same parametrization independent of the initial representation of the contour. The parametrization which satisfy these criteria is the *affine length* [31] :

$$t = \int_C \sqrt[3]{\det\left(\frac{dx}{d\xi} \frac{d^2x}{d\xi^2}\right)} d\xi = \int_C \sqrt[2]{x_\xi y_{\xi\xi} - y_\xi x_{\xi\xi}} d\xi. \quad (2.14)$$

where x_ξ, y_ξ are the first and $x_{\xi\xi}, y_{\xi\xi}$ are the second derivatives of the components $x(\xi)$ and $y(\xi)$ and C is the path along the curve. Affine length causes some difficulty since boundary encoding will eventually be with polygons and this parametrization involves a second order derivative. Use of the second order derivative will result in

a parametrization which is zero along the sides of the polygon and infinite at the vertices. Instead a first order form is used:

$$t = \frac{1}{2} \int_c |det(x(\xi), x_\xi)| d\xi = \frac{1}{2} \int_c |x(\xi)y_\xi - x_\xi y(\xi)| d\xi. \quad (2.15)$$

this parametrization will not be invariant for the case $b \neq 0$ (18) ; that is translation. To avoid this problem the coordinate system is initially moved to the area center define by:

$$x_s = \frac{2 \oint_C x(\xi) det(x(\xi), x_\xi) d\xi}{3 \oint_C det(x(\xi), x_\xi) d\xi}. \quad (2.16)$$

The area center of an affine contour is the affine transform of the area center due to the fact that the affine transformation transforms areas with a constant scale $det(A)$.

2.2.2 Construction of Parameters from Fourier Coefficients

The boundary is encoded as a function of parameter and the Fourier Transform of the resulting function is taken. A point on the boundary is described by a vector function:

$$x = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}. \quad (2.17)$$

Fourier transform is then applied to the functions $u(t)$ and $v(t)$, resulting in a matrix of coefficients:

$$\begin{bmatrix} U_0 & V_1 \\ \dots & \dots \\ V_0 & U_1 \end{bmatrix} \quad (2.18)$$

Although these coefficients are complex, the functions $u(t)$ and $v(t)$ are real and so $U_{-k} = U_k^*$ $V_{-k} = V_k^*$. and all coefficients $[U_k, V_k]^T$ can be discarded for $k < 0$.

A description of the boundary is to be constructed from the Fourier coefficients. The pair $[U_0, V_0]^T$ is discarded due to that it contains no shape information and it depends on translation. Remaining coefficients are shift invariant. We define the relative invariants that is a set of numbers $I_k, I_k \in \mathbb{C}$ (complex plane), which satisfy the following relations. Let I_k^0 represent the k_{th} invariant measured on the reference image, and let I_k represent the same invariant measured on the observed image. If I_k is indeed a relative invariant, it will satisfy:

$$I_k = \mu I_k^0. \quad (2.19)$$

Furthermore, μ will be the same constant for all k . A larger set of invariants can be found as follows: let X_k represent the k_{th} Fourier coefficient vector resulting from the transform of the observation and let X_k^0 represent the same coefficient from the transform of the reference. If the observation did infact result from the affine transform \mathbf{A} applied to the reference, we have to satisfy:

$$X_k = \mathbf{A}X_k^0. \quad (2.20)$$

since the Fourier transform is a linear operator. Choose any two coefficients, say k , and p , and construct the 2x2 matrix:

$$[X_k, X_p]. \quad (2.21)$$

using such a matrix, it may be written:

$$[X_k, X_p] = \mathbf{A}[X_k^0, X_p^0]. \quad (2.22)$$

taking the determinant of both sides we have:

$$\det[X_k, X_p] = \det(\mathbf{A})\det[X_k^0, X_p^0]. \quad (2.23)$$

and we have invariant scalars which obey the definition of (2.16), where $\mu = \det(\mathbf{A})$. To reduce the cardinality and also redundancy of this set we fix p to some constant value such that $p \neq 0$ and $X_p \neq 0$ and define the set of relative invariants Δ_k :

$$\Delta_k = \det[X_k, X_p^*]. \quad (2.24)$$

that set of invariants is complete, that is two planar curves will have the same set of descriptors if and only if they are affine. The absolute invariants are derived from relative invariants of equation (2.21), eliminating the effects of μ , by simply dividing all the invariants by Δ_p :

$$Q_k = \frac{\Delta_k}{\Delta_p} = \frac{|X_k, X_p^*|}{|X_p, X_p^*|} = \frac{U_k V_p^* - V_k U_p^*}{U_p V_p^* - V_p U_p^*}. \quad (2.25)$$

In the absence of noise equation (2.20) may be chosen, but when noise is available, signal to noise ratio should be as high as possible and equation (2.22) should be considered with p for which $|X_k, X_p^*|$ is as large as possible.

2.3 Moment Invariants

Moment invariants are useful features of a two dimensional image. They are invariant to shifts, to changes of scale and to rotations. In other words, they are invariant and to general linear transformations of the image. Affine transformation is a linear transformation, when translation part is removed. So moment invariants can be used to recognize affine transformed objects. These moment invariants are called affine moment invariants.

2.3.1 Moments

Let image $f(x, y)$ to be the intensity function of the image, which is assumed to be piecewise continuous and has compact support, is given. The regular moment m_{pg} is defined as:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy \quad , \quad p, q = 0, 1, 2, \dots \quad . \quad (2.26)$$

Given that intensity function is piecewise continuous and has compact support, it can be proved that moments of all orders exist and that $f(x, y)$ is uniquely determined by infinite set of moments and conversely moments are uniquely determined by $f(x, y)$. The moment generating function of $f(x, y)$ is defined as:

$$M(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{ux+vy} f(x, y) dx dy. \quad (2.27)$$

Note that u and v are real. If moments of all orders exists as assumed, then $M(u, v)$ can be expanded into power series in the moments m_{pq} as follows:

$$M(u, v) = \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} m_{pq} \frac{u^p}{p!} \frac{v^q}{q!}. \quad (2.28)$$

Central moments are defined as:

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy. \quad (2.29)$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$.

The central moments are equivalent to the regular moments of the image that has been shifted such that the image centroid (\bar{x}, \bar{y}) coincides with the origin.

It is assumed that the origin is chosen to coincide with the centroid of the image; therefore, μ_{pq} can also be expressed as:

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy, \quad p, q = 0, 1, 2, \dots \quad . \quad (2.30)$$

2.3.2 Algebraic Invariants

A binary algebraic form of f of order p is defined as:

$$f = a_{p,0}u^p + \binom{p}{1} a_{p-1,1}u^{p-1}v + \dots + \binom{p}{p-1} a_{1,p-1}uv^{p-1} + a_{0,p}v^p \quad (2.31)$$

where u and v are the variables, and $a_{p,0} \dots a_{0,p}$ are the coefficients. Each binary form of order $p = 1, 2, \dots$ has one or more invariants, which are defined as follows: a homogeneous k_{th} order polynomial $\Gamma(a_{p,0}, \dots, a_{0,p})$ of the coefficients is an algebraic invariant of weight g and order k if :

$$I(a'_{p,0}, \dots, a'_{0,p}) = \Delta^g I(a_{p,0}, \dots, a_{0,p}). \quad (2.32)$$

where $a'_{p,0}, \dots, a'_{0,p}$ are the new coefficients obtained by the following general linear transformation into binary form (13):

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha & \gamma \\ \beta & \delta \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix}, \quad \Delta = \begin{vmatrix} \alpha & \gamma \\ \beta & \delta \end{vmatrix} \neq 0. \quad (2.33)$$

if $g = 0$, the invariant is an absolute invariant; otherwise it is called a relative invariant. Given two relative invariants, an absolute invariant can be formed by dividing the suitable powers of relative invariants to remove the Δ^g terms. A simple example of an absolute invariant is that of the binary quartic:

$$f_4(x, y) = ax^4 + 4bx^3y + 6cx^2y^2 + 4dxy^3 + ey^4 \quad (2.34)$$

which has two relative invariants:

$$S = ae - 4bd + 3c^2, \quad g = 4$$

$$T = ace + 2bcd - ad^2 - eb^2 - c^3, \quad g = 6$$

System of linear, quadratic and cubic forms

In the following A,B,C will represent the coefficients of the binary form $Ax^2+2Bxy+Cy^2$, $\alpha, \beta, \gamma, \delta$ those of the cubic form $\alpha x^3 + 3\beta x^2y + \dots + \delta y^3$ and a, b, \dots, e those of the quartic form $ax^4 + 4bx^3y + \dots + ey^4$.

The quadratic form

Invariant: $Q = AC - B^2$ with weight $g = 2$.

The cubic form

Invariant: $P = (\alpha\delta - \beta\gamma)^2 - 4(\alpha\gamma - \beta^2)(\beta\delta - \gamma^2)$, with $g = 6$.

The system of cubic and quadratic forms

Invariants:

$$I = A(\beta\delta - \gamma^2) - B(\alpha\delta - \beta\gamma) + C(\alpha\gamma - \beta^2)$$

$$R = \alpha^2 C^3 - 6\alpha\beta BC^2 + 6\alpha\gamma C(2B^2 - AC) + \alpha\delta(6ABC - 8B^3) + 9\beta^2 AC^2 - 18\beta\gamma ABC + 6\beta\delta A(2B^2 - AC) + 9\gamma^2 A^2 C - 6\gamma\delta A^2 B + \delta^2 A^3.$$

$$M = A^3(3\beta\gamma\delta^2 - \alpha\delta^3 - 2\gamma^3\delta) + 6A^2B(\alpha\gamma\delta^2 - \beta^2\delta^2 - \beta\gamma^2\delta + \gamma^4) + 3A^2C(2\beta^2\gamma\delta - \alpha\gamma^2\delta - \beta\gamma^3) + 12AB^2(2\beta^2\gamma\delta - \alpha\gamma^2\delta - \beta\gamma^3) + 3C(AC + 4B^2)(\alpha\beta^2\delta + \beta^3\gamma - 2\alpha\beta\gamma^2) + 4AB(2B^2 + 3AC)(\alpha\gamma^3 - \beta^3\delta) + 6BC^2(\alpha^2\gamma^2 + \alpha\beta^2\gamma - \alpha^2\beta\delta - \beta^4) + C^3(\alpha^3\delta + 2\alpha\beta^3 - 3\alpha^2\beta\gamma)$$

quartic forms

Invariants:

$$S = ae - 4bd + 3c^2$$

$$T = ace + 2bcd - ad^2 - eb^2 - c^3$$

system quartic and quadratic forms

Invariants:

$$L = eA^2 + 4cB^2 + aC^2 - 4bBC + 2cAC - 4dAD \quad g = 4$$

$$N = A^2(ce - d^2) + B^2(ae - c^2) + C^2(ac - b^2)$$

$$+ 2BC(bc - ad) + 2AC(bd - c^2) + 2AB(cd - be), \quad g = 6$$

system quartic and cubic forms

Invariant:

$$K = a(\beta\delta - \gamma^2)^2 - 2b(\alpha\delta - \gamma\beta)(\beta\delta - \gamma^2) - 2d(\alpha\gamma - \beta^2)(\alpha\delta - \gamma\beta) + c[2(\alpha\gamma - \beta^2)(\beta\delta - \gamma^2) + (\alpha\delta - \gamma\beta)^2] + e(\alpha\gamma - \beta^2)^2$$

Invariants to affine image transformations can be easily constructed from algebraic invariants by using the *Revised Fundamental Theorem of Moment Invariants* via the method explained in [10],[30].

Revised Fundamental theorem of invariants states that:

Let $|\Delta|$ be the absolute value of the determinant Δ of the affine image transformation. If the binary form of order p has an algebraic invariant $I(a_{p,0}, a_{p-1,1}, \dots, a_{0,p})$ of weight w and order k , i.e

$$I(a'_{p,0}, a'_{p-1,1}, \dots, a'_{0,p}) = \Delta^w I(a_{p,0}, a_{p-1,1}, \dots, a_{0,p}). \quad (2.35)$$

then the moments of order p have the same invariant but with the additional factor $|\Delta|^k$:

$$I(a'_{p,0}, a'_{p-1,1}, \dots, a'_{0,p}) = \Delta^g |\Delta|^k I(a'_{p,0}, a_{p-1,1}, \dots, a'_{0,p}). \quad (2.36)$$

2.3.3 Affine Moment Invariants

Affine moment invariants are the moment based descriptors of the planar shapes, which are invariant under general affine transformation.

The affine transformation can be decomposed into six one parameter transformations:

$$\begin{array}{ll} u = x + \alpha, & u = \delta.x \\ v = y & v = y \\ u = x, & u = x + t.y \\ v = y + \beta & v = y \\ u = \omega.x, & u = x \\ v = \omega.y & v = t'.x + y \end{array}$$

<i>Invariant</i> :	μ	Q	P	I	R	S	T	L	N	K	G_i	J	E	F
$g + k$	1	4	10	7	11	6	9	7	10	13	10	14	8	16
K	1	2	4	3	5	2	3	3	4	5	4	4	2	4

Table 2.1: the values of g+k and k for the invariants

Any function F of moments which is invariant under these six transformations will be invariant under the general affine transformation. As talked above, affine moment invariants can be obtained from algebraic invariants using the method in [30] based on the theorem of moment invariants, where the coefficients (a, b, c, \dots) in the expressions for algebraic invariants are replaced by corresponding central moments *i.e* the coefficients $(A, B, C)(x, y)^2$ are replaced by $\mu_{20}, \mu_{11}, \mu_{02}$ respectively; similarly, the coefficients $\alpha, \beta, \gamma, \delta$ of the cubic form $(\alpha, \beta, \gamma, \delta)(x, y)^3$ are replaced by $\mu_{30}, \mu_{21}, \mu_{03}$ respectively and so on for higher forms. As an example, the simplest invariant Q becomes $\mu_{20}m\mu_{02} - m\mu_{11}^2$. If only central moments up to fourth order are used, this means 13 non-zero moments, that leads to 9 independent absolute invariants. One set of nine is presented below.

$$\Gamma_1 = \frac{Q}{\mu_{00}^4} \quad \Gamma_2 = \frac{P}{\mu_{00}^{10}} \quad \Gamma_3 = \frac{I}{\mu_{00}^7}$$

$$\Gamma_4 = \frac{R}{\mu_{00}^{11}} \quad \Gamma_5 = \frac{S}{\mu_{00}^6} \quad \Gamma_6 = \frac{T}{\mu_{00}^9}$$

$$\Gamma_7 = \frac{L}{\mu_{00}^7} \quad \Gamma_8 = \frac{N}{\mu_{00}^{10}} \quad \Gamma_9 = \frac{G_1}{\mu_{00}^9}$$

Most of the invariants are of high order in the coefficients, hence has a large number of terms in their expressions. This is undesirable because they are more noise sensitive compared to invariants with fewer terms.

Chapter 3

Wavelet-based Affine Invariant Functions

A new technique for affine invariant representation is the dyadic wavelet transform based representation. Object contours are decomposed into several components at different resolution levels. Since the wavelet transform is essentially a recurrent filtering process with a kernel which is a bandpass filter [28], the components at each resolution level have a limited bandwidth in the frequency domain. As a result this can limit the effect of noise by selecting a suitable number of resolution levels in the representation. Also, due to preserving spatial information at each resolution level, establishing the point correspondence between elements can be easily achieved. So, A, advantages of spatial (i.e Moment method) and transform domain (i.e Fourier descriptors) representations are combined.

Wavelet coefficients for certain scale values can be efficiently calculated via the discrete dyadic wavelet transform (DWT) Discrete dyadic wavelet transform of a signal is implemented using the filters proposed by Mallat [28],[29] . A filterbank composed of lowpass and highpass filters together with downsamplers are used. This filterbank produces two sets of coefficients: orthogonal detail coefficients which are the even outputs of the highpass filter; and also called as the wavelet coefficients and the approximation coefficients which are the even outputs of the lowpass filter. Downsamplers drop the odd indexed samples . By downsampling, computational cost of implementing DWT drops to $O(N \log N)$.

3.1 Wavelet transform

The Fourier transform is a tool widely used for many scientific purposes, but it is well suited only to the study of stationary signals where all frequencies have an infinite coherence time. The Fourier analysis brings only global information which is not sufficient to detect compact patterns. Gabor introduced a local Fourier analysis, taking into account a sliding window, leading to a time frequency-analysis. This method is only applicable to situations where the coherence time is independent of the frequency. This is the case for instance for singing signals which have their coherence time determined by the geometry of the oral cavity. Morlet introduced the Wavelet Transform in order to have a coherence time proportional to the period [32]. It replaces the Fourier transform's sinusoidal waves by a family generated by translations and dilations of a window called a wavelet. It has two arguments: time and scale. Wavelet transform localizes a function both in space and scaling and has some desirable properties compared to the Fourier Transform . The Morlet-Grossmann definition of the continuous wavelet transform [33] for a 1D signal $f(x) \in L^2(R)$:

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \psi^* \left(\frac{x - a}{b} \right) dx. \quad (3.1)$$

where z^* denotes the complex conjugate of z , ψ^* is the analyzing wavelet, $a(> 0)$ is the scale parameter and b is the position parameter

the transform is characterized by following properties:

- (i). It is a linear transformation
- (ii). It is covariant under translations

$$f(x) \longrightarrow f(x - \mu),$$

$$W(a, b) \longrightarrow W(a, b - \mu), \text{ and}$$

- (iii).It is covariant under dilations

$$f(x) \longrightarrow f(sx), W(a, b) \longrightarrow S^{\frac{-1}{2}} W(sa, sb).$$

The last property makes the wavelet transform very suitable for analyzing hierar-

chical structures. It is like a mathematical microscope with properties that do not depend on the magnification. In Fourier space, we have:

$$\widehat{W}(a, \nu) = \sqrt{a} \widehat{f}(\nu) \psi^*(a, \nu). \quad (3.2)$$

When the scale a varies, the filter $\psi^*(a, \nu)$ is only reduced or dilated while keeping the same pattern

Now consider a function $W(a, b)$ which is the wavelet transform of a given function $f(x)$. $f(x)$ can be restored using the formula:

$$f(x) = \frac{1}{C_x} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} W(a, b) \chi\left(\frac{x-b}{a}\right) \frac{da \cdot db}{a^2}. \quad (3.3)$$

where

$$C_x = \int \frac{\widehat{\psi^*} \widehat{\chi}(\nu)}{\nu} d\nu. \quad (3.4)$$

here $\chi(x)$ is the wavelet function for synthesis. Generally, $\chi(x)$ is selected such that: $\chi(x) = \psi(x)$ for reconstruction.

3.1.1 Multiresolution Analysis and Discrete Wavelet Transform

Multiresolution analysis [35] results from the embedded subsets generated by the interpolations at different scales. A function $f(x)$ is projected at each step j onto the subset V_j . This projection is defined by the scalar product $c_j(k)$ of $f(x)$ with the scaling function $\phi(x)$ which is dilated and translated:

$$c_j(k) = \langle f(x), 2^{-j} \phi(2^{-j}x - k) \rangle. \quad (3.5)$$

As $\phi(x)$ is a scaling function with the property:

$$\frac{1}{2}\phi\left(\frac{x}{2}\right) = \sum_n h(n)\phi(x - n). \quad (3.6)$$

or

$$\widehat{\phi}(2\nu) = \widehat{h}(\nu)\widehat{\phi}(\nu). \quad (3.7)$$

where $\widehat{h}(\nu)$ is the Fourier transform of the function. then

$$\widehat{h}(\nu) = \sum_n h(n)e^{-2\pi n\nu}. \quad (3.8)$$

Equation 6 permits us to compute directly the set $c_{j+1}(k)$ from $c_j(k)$. If we start from the set $c_0(k)$ and compute all the sets $c_j(k)$, with $j > 0$, without directly computing any other scalar product:

$$c_{j+1}(k) = \sum_n h(n - 2k)c_j(n). \quad (3.9)$$

At each step, the number of scalar products is divided by 2. Step by step the signal is smoothed and information is lost. The remaining information can be restored using the complementary subspace W_{j+1} of V_{j+1} in V_j . This subspace can be generated by a suitable wavelet function $\psi(x)$ with translation and dilation.

$$\frac{1}{2}\phi\left(\frac{x}{2}\right) = \sum_n g(n)\phi(x-n). \quad (3.10)$$

or

$$\widehat{\phi}(2\nu) = \widehat{g}(\nu)\widehat{\phi}(\nu). \quad (3.11)$$

We compute the scalar product $\langle f(x), 2^{-(j+1)}\psi(2^{-(j+1)}x - k) \rangle$ with:

$$w_{j+1}(k) = \sum_n g(n-2k)c_j(n). \quad (3.12)$$

with this analysis we have built the first part of a filter bank [35]. In order to restore original data, *Mallat* uses the properties of orthogonal wavelets [28].

In the decomposition, the function is successively convolved with the two filters H_d (low frequencies) and H_h (high frequencies). Each resulting function is decimated by suppression of one sample out of two; and we iterate as in Figure 3.1.

Dyadic Wavelet Transform

Dyadic wavelet transforms are scale samples of wavelet transforms following a geometric sequence of ratio 2. Time is not sampled. The dyadic wavelet transform of f is defined by

$$Wf(u, 2^j) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{2^j}} \psi\left(\frac{t-u}{2^j}\right) dt = f * \overline{\psi}_{2^j}(u) \quad ; \quad \overline{\psi}_{2^j}(t) = \overline{\psi}_{2^j}(-t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{-t}{2^j}\right) \quad (3.13)$$

It defines a stable complete representation if its Heisenberg boxes cover all of the frequency axis, that is, if there exist A and B such that:

$$\forall \omega \in \mathbb{R}, \quad A \leq \sum_{-\infty}^{+\infty} |\widehat{\psi}(2^j \omega)|^2 \leq B. \quad (3.14)$$

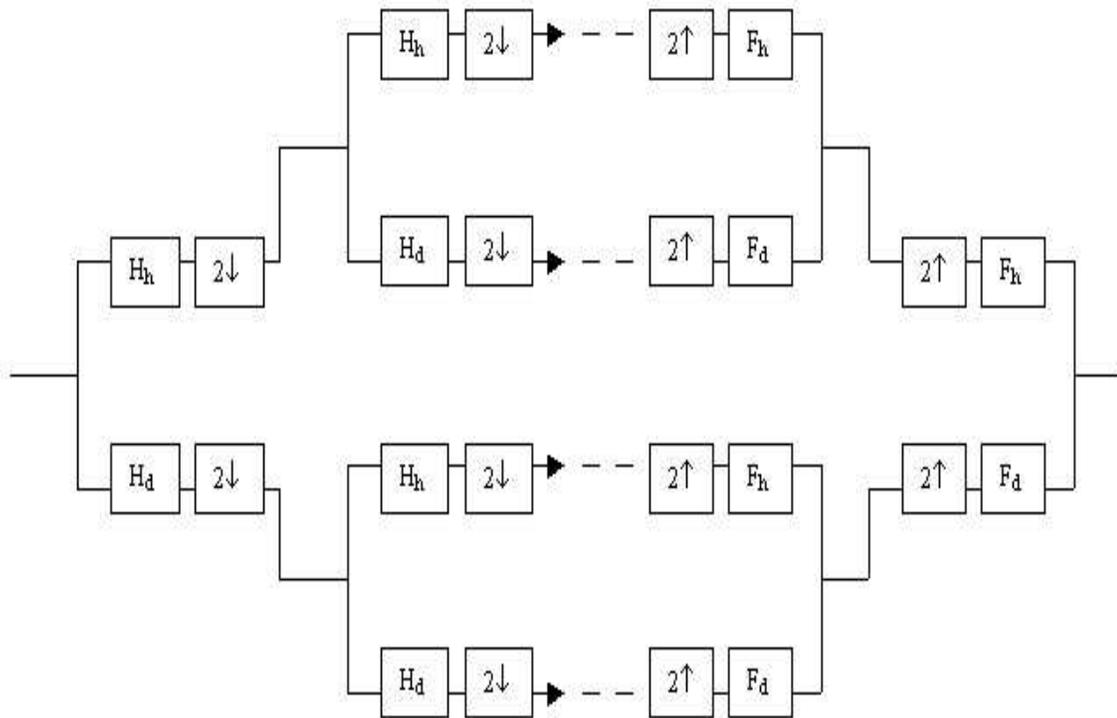


Figure 3.1: The filterbank associated with multiresolution analysis. H_h, F_h are high-pass filters and H_d, F_d are low-pass filters. In the equations, high-pass filter is used as g and low-pass filter is used as h .

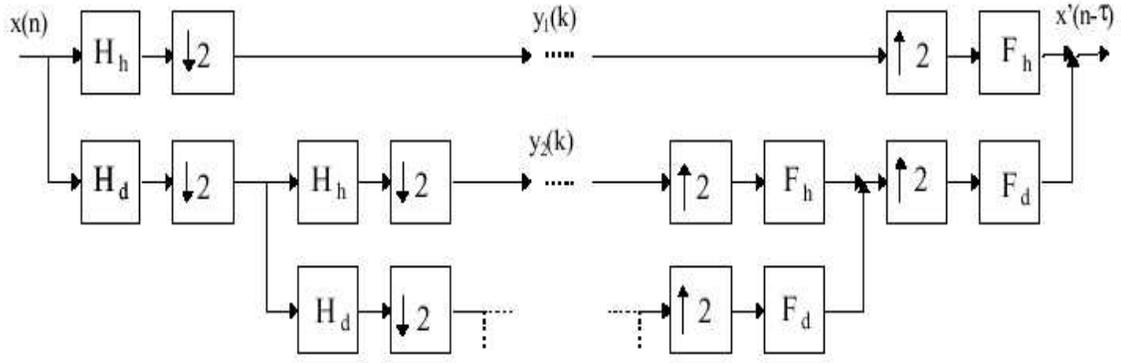


Figure 3.2: Block diagram of dyadic wavelet transform(left) and its associated inverse transform(right). H_h, F_h are high-pass filters and H_d, F_d are low-pass filters.

The family of dyadic wavelets is a frame of $L^2(R)$. To build dyadic wavelets, it is sufficient to satisfy the previous condition. To do so, it is possible to proceed as for the construction of orthogonal and biorthogonal wavelet bases, using conjugate mirror or perfect reconstruction filter banks. The wavelets satisfy then scaling equations and the fast dyadic wavelet transform is implemented using filter banks. Filterbank structure of dyadic wavelet transform is shown in Figure 3.2.

3.2 Tieng-Boles Function

Tieng and Boles proposed an affine invariant function using one dyadic level [15],[25]. Their approach starts with constructing the parametric equation of the object contours based on one of the affine invariant parameters, the enclosed area [34]. Let $x(t), y(t)$ denote the contour of the object. Then the enclosed area parameter can be calculated by contour integration along the object contour C as follows:

$$\sigma = \frac{1}{2} \int_C |xy - yx| dt. \quad (3.15)$$

In the discrete case, the first derivatives \dot{x}, \dot{y} can be calculated via finite difference

equation as follows:

$$\dot{x}(t) = x(t+1) - x(t), \quad \dot{y}(t) = y(t+1) - y(t) \quad (3.16)$$

by replacing the integration by summation, contour integral can be implemented as:

$$\sigma(k) = \begin{cases} 0, & \text{if } k=1 ; \\ \frac{1}{2} \sum_{t=1}^{k-1} |x(t)y(t+1) - x(t+1)y(t)|, & \text{otherwise.} \end{cases} \quad (3.17)$$

where $k = 1, \dots, T$ and T is the equally spaced segments along the contour.

Since the objects have undergone a general affine transformation, their enclosed areas will generally have different values. A normalization process is required to make these value same for matching objects. Furthermore, although the object contour is uniformly sampled, the parameter σ is digitized with nonequal intervals when arc length and enclosed area are not linearly dependent. As dyadic wavelet transform, which requires input data to be sampled with equal intervals, is used, normalized parameter σ must also be resampled prior to processing.

Let $[x(\sigma), y(\sigma)]$ and $[x_a(\sigma_a), y_a(\sigma_a)]$ be the parametric equations of two contours that differ only by general affine transformation. After normalization and resampling, there is a one-to-one relation between σ and σ_a . Thus these parameters are dropped in the following equations for simplicity. The relation between the two contours can be expressed as :

$$\begin{aligned} x_a &= a_{11}x + a_{12}y + b_1 \\ y_a &= a_{21}x + a_{22}y + b_2 \end{aligned} \quad (3.18)$$

The next step in the representation is to calculate the dyadic wavelet transform of both sides via the undecimated FWT (Fast Wavelet Transform). After taking dyadic wavelet transform of both sides, the relation between the approximation, A_j and detail W_j , signals of x_a, y_a, x and y at a particular resolution level j is given by:

$$\begin{aligned} A_j x_a &= a_{11}A_j x + a_{12}A_j y + 2^{j/2}b_1 \\ A_j y_a &= a_{21}A_j x + a_{22}A_j y + 2^{j/2}b_2 \end{aligned} \quad (3.19)$$

and

$$\begin{aligned} W_j x_a &= a_{11} W_j x + a_{12} W_j y \\ W_j y_a &= a_{21} W_j x + a_{22} W_j y \end{aligned} \quad (3.20)$$

As seen in the above equations, approximation signals depend on the position of object contours in images but the detail signals do not. If the object centroid is used as the origin, then $b_1 = b_2 = 0$. Equations (3.24) and (3.25) can be written in matrix form as:

$$\begin{bmatrix} A_j x_a & W_j x_a \\ A_j y_a & W_j y_a \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} A_j x & W_j x \\ A_j y & W_j y \end{bmatrix}. \quad (3.21)$$

Based on the determinant properties, *relatively* affine invariant representation is obtained in the form:

$$A_j x W_j y - A_j y W_j x. \quad (3.22)$$

The representation can be made *absolutely* invariant by representing it in ratio form as:

$$M_j(k) = \frac{A_j x(k) W_j y(k) - A_j y(k) W_j x(k)}{A_i x(n) W_i y(n) - A_i y(n) W_i x(n)} \quad (3.23)$$

with $A_i x(n) W_i y(n) - A_i y(n) W_i x(n) \neq 0$.

3.3 Khalil-Bayoumi Function

Khalil-Bayoumi proposed an affine-invariant representation[16] similar to Tieng-Boles representation, but their representation uses only the detail signals from two dyadic levels. Their wavelet-based affine invariant function also use undecimated wavelet transform. We here explain the Khalil-Bayoumi affine invariant function based on decimated wavelet transform and propose a different method based on

image projections with Khalil-Bayoumi and Tieng-Boles representations instead of using the contour points.

Let us denote the wavelet transform of a signal $x(t)$ at the resolution scale i as $W_i x(t)$, then the wavelet transform of (3.1) and (3.2) is given as :

$$W_i \tilde{x}(t) = a_1 W_i x(t) + a_2 W_i y(t). \quad (3.24)$$

$$W_i \tilde{y}(t) = b_1 W_i x(t) + b_2 W_i y(t). \quad (3.25)$$

$W_i a_0 = W_i b_0 = 0$ due to the high-pass filter. Let the object be defined by $x(t)$ and $y(t)$. An affine invariant function for an object using the wavelet coefficients of signals $x(t)$ and $y(t)$ for two wavelet scales i, j can be defined as:

$$f_{ij}(t) = W_i x(t) W_j y(t) - W_i y(t) W_j x(t). \quad (3.26)$$

It can be shown that:

$$\tilde{f}_{ij}(t) = W_i \tilde{x}(t) W_j \tilde{y}(t) - W_i \tilde{y}(t) W_j \tilde{x}(t) = \det(A) f_{ij}(t). \quad (3.27)$$

The invariant function $f_{ij}(t)$ uses only the detail coefficients of the two different levels. The Khalil-Boyoumi's affine function is computed using undecimated wavelet transform meaning that no downsampling operation is performed. This results in a lot of computational cost of the wavelet transform. If the length of the signal is N , then in the undecimated transform case, length- N signals are filtered at each scale. In decimated case, downsampling halves the signal length at each level.

The wavelet signal $W_i x(t)$, at resolution scale $i=1$ can be expressed as ,

$$W_i x(t) = \sum d_k w(t - k), i = 1. \quad (3.28)$$

d_k are the wavelet coefficients produced by a decimated Filterbank at resolution

scale $i = 1$ and we call $w(t)$ as the mother wavelet. If the length of the data is N , then the limits of the summation in (3.33) is from $k=0$ to $k=N$ where a circular computation of DWT is assumed. Similarly for $j = 2$, $W_j y(t)$ can be expressed as:

$$W_j y(t) = \sum e_i w(t/2 - l). \quad (3.29)$$

where e_i are the wavelet coefficients at resolution scale, $j = 2$. The limits of the summation is from $l = 0$ to $l = N/2$ because of the downsampling. Let us assume that $w(t)$ is the Haar wavelet, i.e:

$$w(t) = 1 \quad \text{for } 0 < t < 0.5, w(t) = -1 \quad \text{for } 0.5 < t < 1, w(t) = 0 \quad \text{otherwise.} \quad (3.30)$$

we can express the first term of the affine function (3.31) as :

$$W_i x(t) W_j y(t) = \sum \sum d_k e_l w(t - k) w(t/2 - l) \quad \text{for } i = 1, j = 2; \quad (3.31)$$

Direct computation of (3.36) and the affine invariant function (3.31) requires $NxN/2$ and NxN multiplications, respectively. However as, $w(t)w(t/2) = w(t)$, $w(t)w(t/2 - k) = 0$ for $k > 1$, since the Haar wavelet has a compact support with length 2. Similarly, $w(t - 2)w(t/2 - 1) = w(t - 2)$, $w(t - 3)w(t/2 - 1) = -w(t - 3)$, etc. As a result of these relations, equation (3.36) can be expressed as:

$$W_i x(t) W_j y(t) = \sum_{k=0, \text{even}}^N d_k e_{k/2} w(t - k) - \sum_{k=1, \text{odd}}^N d_k e_{(k-1)/2} w(t - k) \quad \text{for } i = 1, j = 2; \quad (3.32)$$

we require only N multiplications for the right hand side of equation (3.37). The affine invariant function $f_{ij}(t)$, for $i=j+1$, can be expressed as in [36]:

$$f_{ij}(t) = \sum_{k, \text{even}} d_k^i e_{k/2}^{i+1} w_i(t - k) - \sum_{k, \text{odd}} d_k^i e_{(k-1)/2}^{i+1} w_i(t - k)$$

$$+ \sum_{k, \text{even}} e_k^i d_{k/2}^{i+1} w_i(t-k) - \sum_{k, \text{odd}} e_k^i d_{(k-1)/2}^{i+1} w_i(t-k) \quad (3.33)$$

where $w_i(t) = w(t/2^i)$ is the wavelet of the resolution scale i , d_k^i , and e_k^i are the wavelet coefficients of the signals at resolution level i , respectively. This equation has an important feature that one can compute it by using the computationally efficient orthogonal wavelet transform as the filterbank having downsamplers can compute wavelet coefficients d_k^i , and e_k^i . Equations (3.37) and (3.38) are developed for the specific case of $i = 1, j = i + 1$. However similar equations with $O(N)$ can be easily developed for any i, j values due to the fact that $w(t)w(t/2^j) = w(t), \dots, w(t-j)w(t/2^j) = -w(t-j)$, and 0, otherwise etc because $w(t)$ has a compact support. In the undecimated WT implementation as in the case for Khalil-Bayoumi [16] paper, length- N signals are filtered at each level; but in the decimated implementation $-N/2^i$ signals are filtered at resolution level i so the final stage of constructing $f_{ij}(t)$ requires only $O(N)$ arithmetic.

Equation (3.38) is obtained by taking the advantage of the fact that Haar wavelet has compact support. Some computationally efficient signal reconstruction algorithms from WT also take advantage of this fact [27]. In fact, all wavelets constructed from FIR filters have compact support. So the double summation in (3.36) can be reduced to as set of single summations as (3.37) for all compactly supported wavelets and one can obtain equations like (3.38) as well. As an example, widely used Daubechies-4 wavelet[35] has a compact support of length 6, ie $w(t) = 0$ for $t > 6$, and $t < 0$. In the case of Daubechies-4 wavelet, $w(t)w(t/2 - k) = 0$ for $k > 3$. This results in a slightly higher computational cost compared to the Haar wavelet case but more robustness against noise is achieved. In general the length of data N is much higher than the support length of most wavelets and this leads to significant computational savings.

3.4 Wavelet Affine Function with Image Projection

As stated above our method relies on image projections and high filtered images from various angles. We do not use the contour points as signals to wavelet analysis and then to invariant function, instead we take projection of an image from various angles and high-pass filter image and use combination of the image projection and projection of the high-pass filtered image from various projection angles as input signals to Khalil-Bayoumi affine function[16] after taking their dyadic wavelet transform.

Define $g(\phi, s)$ as a 1D projection of an image at angle ϕ ; $q(\phi, s)$ is the line integral of the image intensity $f(x, y)$, along a line l that is distance s from the origin and at an angle ϕ from the x axis,

$$g(\phi, s) = \int_l f(x, y) dl \tag{3.34}$$

All points on the line satisfy the equation: $x \cdot \sin(\phi) - y \cdot \cos(\phi) = s$. Therefore the projection function $g(\phi, s)$ can be written as:

$$g(\phi, s) = \int \int f(x, y) \delta(x \sin(\phi) - y \cos(\phi) - s) dx dy \tag{3.35}$$

Projection signal is the first signal we used in the affine invariant function [16]. After taking the projection of the binary image at various angles, we add the signals one after another and obtain a final signal. For the second signal as talked above, we use the projection of the high-pass filtered image. At each projection angle, we high-pass filter the image with the basic high-pass filter [-1,1]. After filtering the binary synthetic image, we calculate absolute value of the image. Then, we combine these signals by adding one after another to have a final signal and use this final signal as the second input signal to the affine function (3.41),[16]. The algorithm may be shown as below in figure 3.3 flowgraph.

A model image, model airplane 12 in Figure 3.7, has projection at 40° and projection of high-pass filtered image of it, at projection angle 40° is shown in Figure

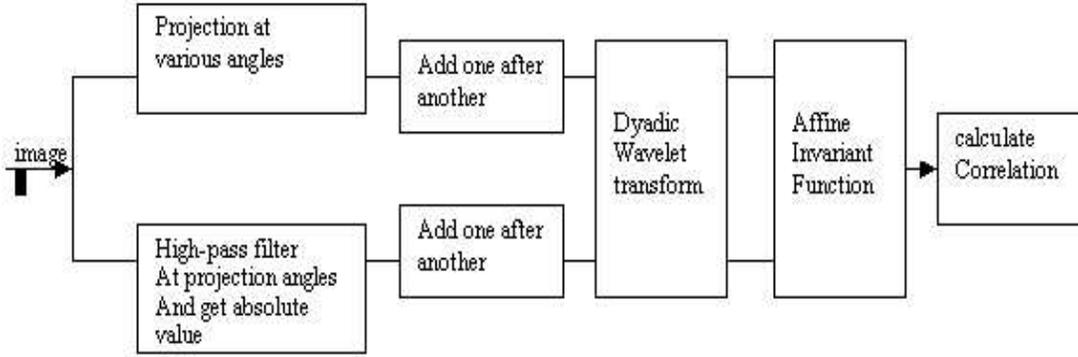


Figure 3.3: Our algorithm

3.4. In Figure 3.5 projection and high-pass filtered image for model image 12 is shown at angles, $0^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ$. These two signals are subjected to dyadic wavelet transform. Then signals at different scales are used to construct the affine function:

$$I = W_{ix}W_{jy} - W_{iy}W_{jx} \quad (3.36)$$

where x stands for the projection of the image at various angles and y stands for the projection of the high-pass filtered image at same angles. W_i, W_j are the detail signals, in other words, outputs of the high-pass filters at Filterbank at resolution scale i and j respectively.

To recognize the object, correlation between the invariant function calculated for object and invariant functions of objects at database is compared. Let R_{ij} be the correlation function [16] between the test object's affine function and the affine function of an object in database, calculated using dyadic scales i and j . A final correlation function using various correlation functions at different dyadic scales is given as [36]:

$$R_{final} = v_1 R_{i_1 j_1} + v_2 R_{i_2 j_2} + \dots + v_k R_{i_k j_k}. \quad (3.37)$$

Final correlation function is used for final decision.

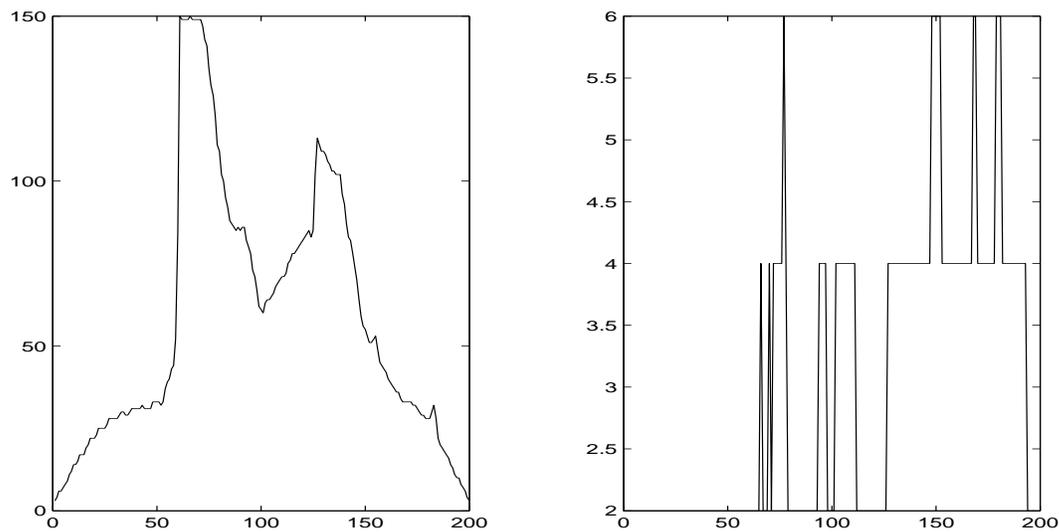


Figure 3.4: Projection(left) and projection of the high-pass filtered(right) of airplane model 12 at 40°

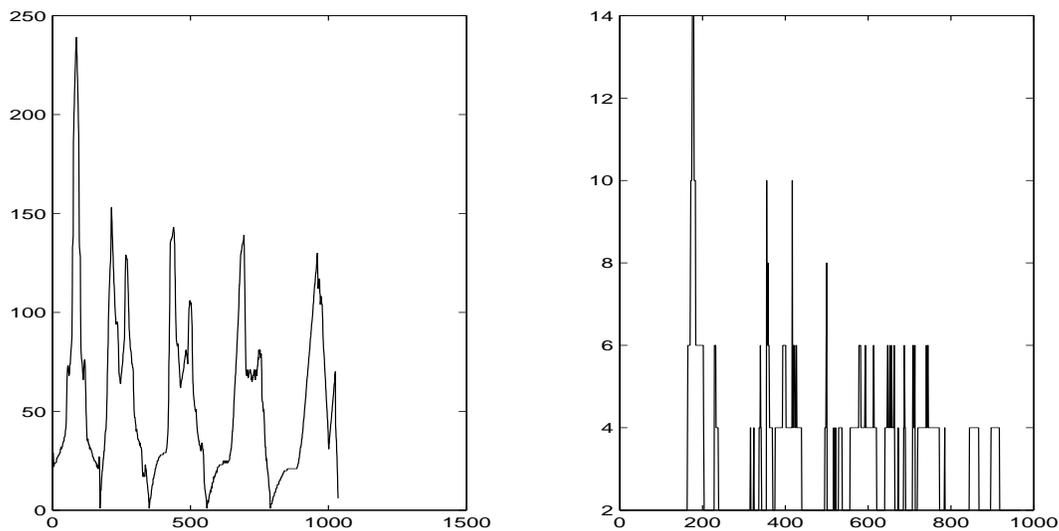


Figure 3.5: Projection(left) and projection of airplane model 12 after high-pass filtering (right) at $0^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ$, used as input signal to wavelet transform and then affine function

3.5 Experimental Results

A computationally efficient algorithm using projections is offered in the previous section based on Khalil-Bayoumi affine invariant function[13]. We performed an experiment to investigate the recognition power of the method.

In this experiment, k invariant functions $f_{ij}(t)$ for a given test object are calculated by using consecutive pairs of resolution levels $(i_1, i_{1+1}), (i_2, i_{2+1}), \dots, (i_k, i_{k+1})$ corresponding k invariant functions for each model object are kept in a database. The correlations between the k invariant functions of the test object and each model object are calculated to get correlation values R_1, R_2, \dots, R_k , which are defined as:

$$R(I_1(t), I_2(t)) = \frac{\int I_1(t)I_2(t)dt}{\|I_1\|\|I_2\|} \quad (3.38)$$

where $I_1(t)$ and $I_2(t)$ represent the invariant functions. The final decision for the test airplane and model airplane is found by linearly combining the k correlation values as follows:

$$R_{final} = \nu_1 R_1 + \nu_2 R_2 + \dots + \nu_k R_k \quad (3.39)$$

where $\nu_1 + \nu_2 + \dots + \nu_k = 1$. As a rule of thumb more weight should be given to resolution levels containing more signal energy to be more robust against noise.

<i>Test image</i>	1	2	3	4	5	6	7	8	9	10
<i>Model image</i>	6	2	19	15	5	1	8	16	20	10

Table 3.1: Model Images Used to Produce the Test Images

This experiment is carried out with airplane images that were used in Figure 3.6 and Figure 3.7. Table 3.1 shows which test image is produced from which model image. The same type of wavelet with [16] is used. 20 model images and 10 test images are constructed by random affine transformation to randomly selected 10 model images. We used all the 10 affine transformed test images in our experiment. Parameters for test images are in appendix part. For our method, projection and high-pass filtered signals are normalized to length 512. The correlations between the test images and model images are calculated and the result is determined according to the model producing the highest correlation value. The signal to noise ratio is defined as in [16]. Three noise levels are tested . In the first experiment SNR is about 50 dB, in the second experiment SNR is 20 dB and finally third experiment is performed with an SNR of 15 dB. Signal to noise ratios are determined as in [13]. By using the same data set, we compared our method with the Khalil-Bayoumi method, Implicit polynomials, Fourier descriptors and Moment invariants.

The results are given in table formats in appendix. In a table, the first column gives the number of the test image. In the second column, highest correlation and the number of the model image(in parenthesis) which highest correlation value is calculated, is shown. In the third column, second highest correlation value and number of model image which it is calculated is shown. At the fourth and fifth columns, only the third and fourth highest correlation values are shown. The number of the model plane which they are calculated from, is not given.

As seen from the tables, our method performs well at all three noise levels. Test images are recognized with a hundred percentage efficiency at all noise levels. Khalil-Bayoumi method also performed well but at the 15dB noise level it failed to recognize one of the test images. This means that our method is more robust to noise

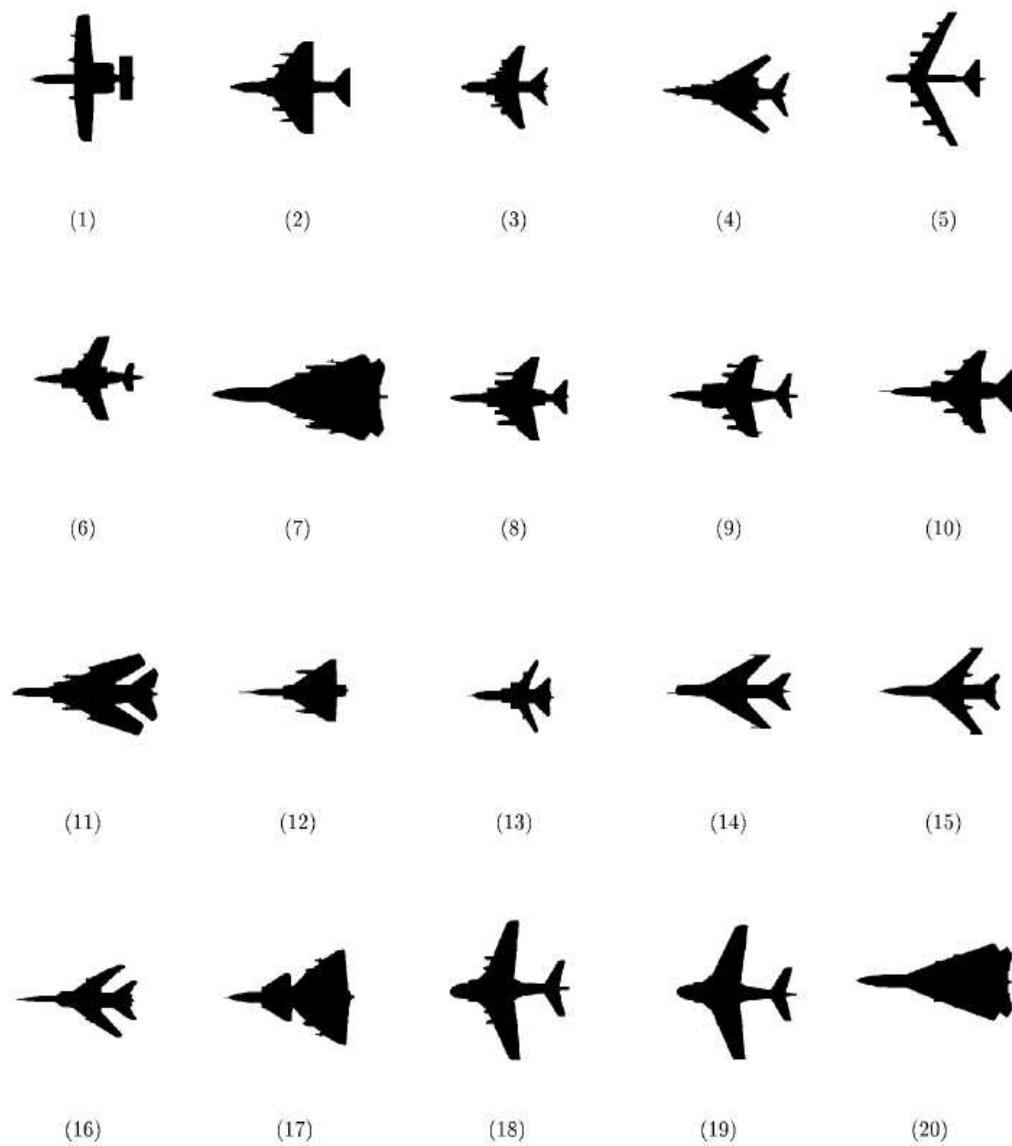


Figure 3.6: The airplane models

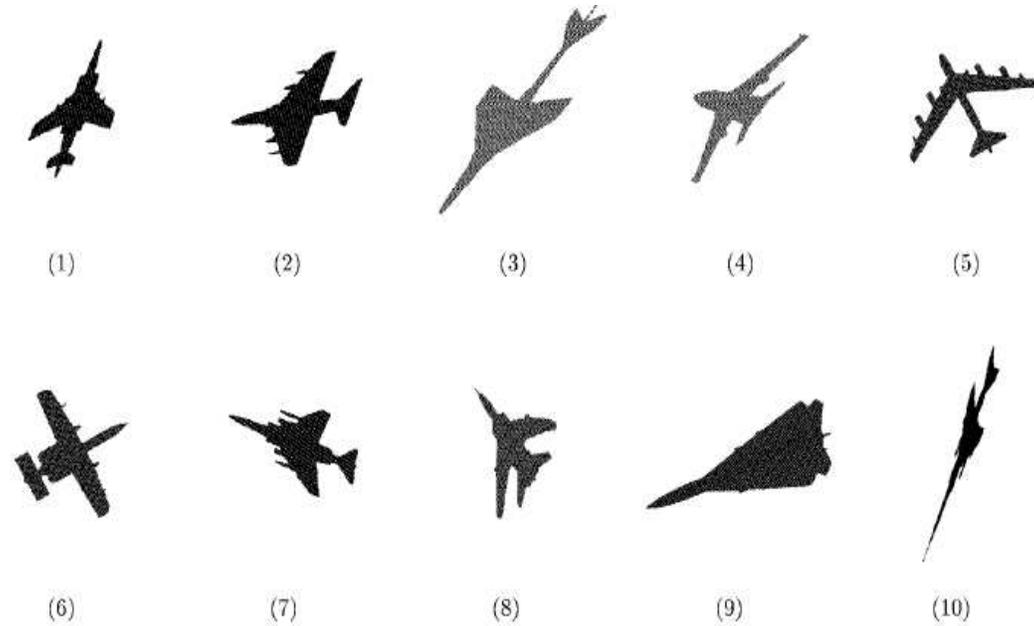


Figure 3.7: The test images

compared to wavelet-based Khalil- Bayoumi method. Our method performed better than the traditional methods at all noise levels. Fluctuations at data in graphics is due to that, wavelet representation used, i.e. wavelet function and filters , are not very suitable for that shape. We used the representation which gives the best result at overall(Deubechies-4 filters).

We made an experiment with Fourier descriptors and Moment invariants with the same data set we used for testing our method. We computed Euclidean distance between resulting vectors to match the test image to model image which test image is produced from. For each test image, 10 experiments are done resulting 100 trials. Best results for Fourier descriptors and Moment invariants are given in appendix. For the low-noise case (50 dB), Moment invariants achieved a recognition performance of 82 percentage and Fourier descriptors achieved a performance of 76 percentage. For high-noise level case (20dB), Moment invariants performed 63 percentage and Fourier descriptors had 78 percentage. This is an expected result since moment invariants are sensitive to noise. Our method which has a recognition performance of 100 percentage at low-noise level and 96 percentage at high-noise level, is obviously more successful compared to Fourier descriptors and Moment in-

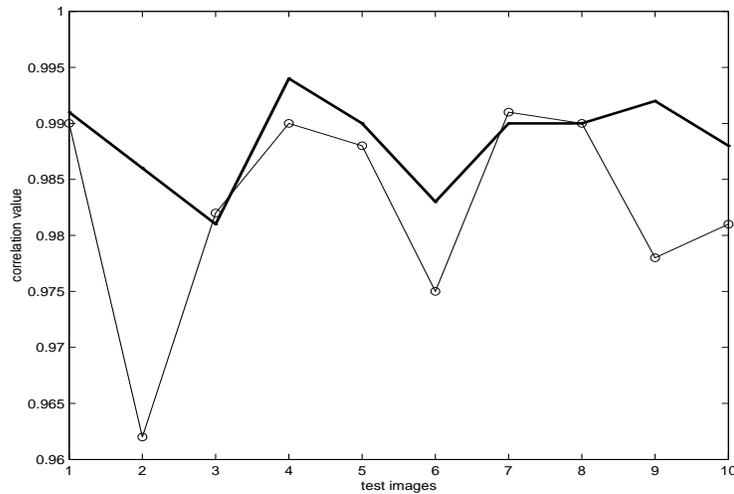


Figure 3.8: Low-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method.

variants and more robust against noise. For the implicit polynomials, we used the same data set and same affine transformed test images. We obtain the implicit polynomial coefficients via the affine invariant 3L fitting algorithm [17],[18]. But this process is not completely affine invariant. We replaced the coefficients into the invariants obtained in [17] and used these invariants for the recognition of our 10 affine transformed model planes.

Figure 3.7, 3.8 shows the highest correlation value between test image and the model images for Our method, Khalil-Bayoumi method and Implicit Polynomials. All methods managed to recognize 10 test images. In Figure 3.9, Our method and Khalil-Bayoumi comparison is given. Both methods recognized 10 test images. Our correlation values are higher than Khalil-Bayoumi method's. Figure 3.10 shows our method vs Implicit polynomials; where Implicit polynomials failed to recognize one of the test images correctly (shown by arrow head in figure). When we increased the noise to 15dB, Khalil-Bayoumi had a false detection; shown by arrow head in Figure 3.11.

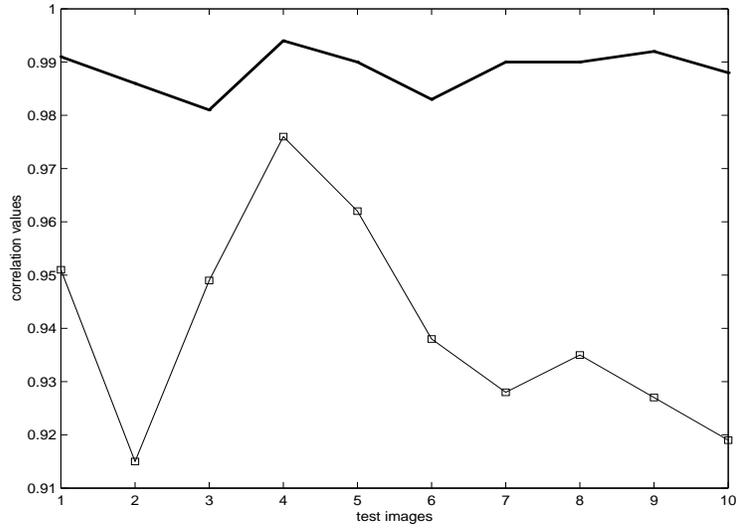


Figure 3.9: Low-noise level correlation values for our method and Implicit polynomials. Thick line corresponds to our method and thin line with square marking corresponds to Implicit polynomials.

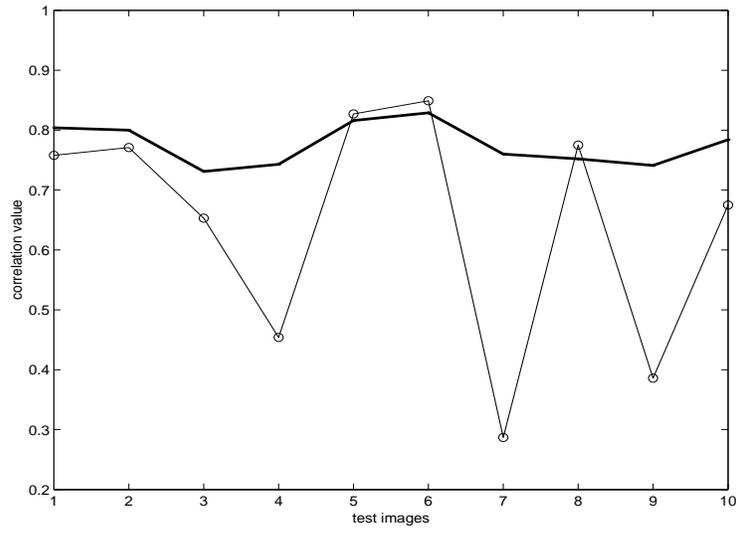


Figure 3.10: High-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method.

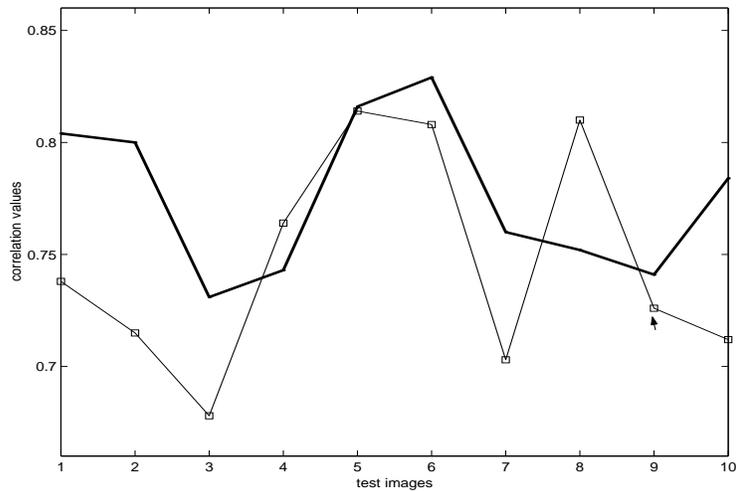


Figure 3.11: High-noise level correlation values for our method and Implicit polynomials. Thick line corresponds to our method and thin line with square marking corresponds to Implicit polynomials. Arrow head shows false detection.

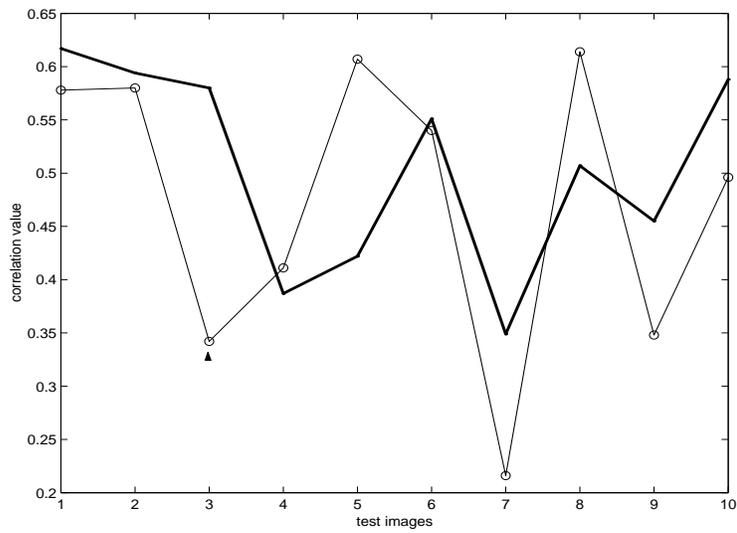


Figure 3.12: Highest-noise level correlation values for our method and Khalil-Bayoumi method. Thick line corresponds to our method and thin line with circle marking corresponds to Khalil-Bayoumi method. Arrow head shows false detection.

Chapter 4

Conclusion

In this thesis, the problem of 2D object recognition using wavelet based affine invariant functions is considered. In previous works, affine invariant representations use the affine invariant parameters affine arc length and enclosed area. Adopting these invariant parameters, a method is derived based on objects contours and undecimated dyadic wavelet transform. We derived an algorithm using the 1D projections of the 2D objects and high-pass filtered images of objects with decimated wavelet transform. Wavelet functions that have compact support and symmetric shape are used. We compared the image recognition power of our method with previous affine invariant function based methods with experiments using synthetic images. Besides a comparison of the method with Fourier descriptors, Affine invariant moments and Implicit polynomials is carried out. Experimental results show that affine invariant function based methods are more successful than traditional methods. Our method detected the affine transformed objects accurately under noise even better than other affine invariant function based methods.

Chapter 5

Appendix

I

The results are given in table formats as explained in section 3.3. In a table, first column gives the number of the test image. In second column, highest correlation and the number of the model image(in parenthesis) which highest correlation value is calculated, is shown. In third column, second highest correlation value and number of model image which it is calculated is shown. In fourth and fifth columns, only the third and fourth highest correlation values are shown. The number of the model plane which they are calculated from, is not given. In the table showing the results for Tieng-Boles function, only highest correlation values are given for 50dB and 20dB noise levels. For Implicit polynomials, Invariant values determines the match. For Fourier descriptors and Moment invariants, vector coefficients are by calculating Euclidean distance similar to correlation and smallest value of the Euclidean distance is used to match the test image and model images.

<i>test.im \ b.match</i>	1	2	3	4
(01)	0.991(06)	0.918(10)	0.900	0.810
(02)	0.986(02)	0.845(03)	0.795	0.725
(03)	0.981(19)	0.867(18)	0.816	0.763
(04)	0.994(15)	0.905(14)	0.828	0.796
(05)	0.990(05)	0.726(01)	0.711	0.632
(06)	0.983(01)	0.869(05)	0.819	0.781
(07)	0.990(08)	0.970(09)	0.910	0.733
(08)	0.990(16)	0.887(13)	0.842	0.728
(09)	0.992(20)	0.847(07)	0.751	0.618
(10)	0.988(10)	0.909(06)	0.876	0.799

Table 5.1: the results at low-noise level for our method

<i>test.im \ b.match</i>	1	2	3	4
(01)	0.990(06)	0.724(09)	0.690	0.606
(02)	0.962(02)	0.765(10)	0.759	0.756
(03)	0.982(19)	0.774(18)	0.536	0.244
(04)	0.990(15)	0.606(06)	0.594	0.339
(05)	0.988(05)	0.444(02)	0.382	0.367
(06)	0.975(01)	0.598(18)	0.557	0.541
(07)	0.991(08)	0.749(10)	0.561	0.514
(08)	0.990(16)	0.769(14)	0.739	0.632
(09)	0.978(20)	0.702(07)	0.377	0.167
(10)	0.981(10)	0.765(06)	0.749	0.683

Table 5.2: the results at low-noise level for Khalil-Bayoumi method

<i>test.im \ b.match</i>	1	2	3	4
(01)	0.804(06)	0.773(10)	0.714	0.685
(02)	0.800(02)	0.786(03)	0.741	0.678
(03)	0.731(19)	0.706(18)	0.722	0.594
(04)	0.743(15)	0.720(14)	0.697	0.659
(05)	0.816(05)	0.785(01)	0.707	0.603
(06)	0.829(01)	0.794(05)	0.656	0.655
(07)	0.760(08)	0.753(09)	0.700	0.568
(08)	0.752(16)	0.710(13)	0.688	0.615
(09)	0.741(20)	0.730(07)	0.693	0.529
(10)	0.784(10)	0.766(06)	0.726	0.577

Table 5.3: the results at high-noise level for our method

<i>test.im \ b.match</i>	1	2	3	4
(01)	0.758(06)	0.583(09)	0.557	0.533
(02)	0.771(02)	0.716(09)	0.631	0.606
(03)	0.653(19)	0.409(18)	0.382	0.189
(04)	0.454(15)	0.333(06)	0.243	0.217
(05)	0.827(05)	0.424(02)	0.405	0.403
(06)	0.849(01)	0.606(18)	0.605	0.486
(07)	0.287(08)	0.202(10)	0.167	0.156
(08)	0.775(16)	0.580(14)	0.549	0.546
(09)	0.386(20)	0.368(01)	0.438	0.278
(10)	0.675(10)	0.510(08)	0.508	0.474

Table 5.4: the results at high-noise level for Khalil-Bayoumi method

<i>test.im \ b.match</i>	<i>our method</i>	<i>Khalil – Bayoumi</i>
(01)	0.617(06)	0.443(06)
(02)	0.594(02)	0.580(02)
(03)	0.580(19)	0.342(18)
(04)	0.387(15)	0.411(15)
(05)	0.422(05)	0.607(05)
(06)	0.551(01)	0.540(01)
(07)	0.349(08)	0.216(08)
(08)	0.507(16)	0.614(16)
(09)	0.455(20)	0.348(20)
(10)	0.588(10)	0.496(10)

Table 5.5: the results at highest-noise level for our method and Khalil-Bayoumi method. In second column highest correlation values for our method is shown and in third column for Khalil-Bayoumi method

<i>test.im \ b.match</i>	<i>Low noise</i>	<i>High noise</i>
(01)	0.914(06)	0.817(06)
(02)	0.816(02)	0.748(02)
(03)	0.842(19)	0.734(18)
(04)	0.960(15)	0.705(15)
(05)	0.952(05)	0.717(05)
(06)	0.946(01)	0.708(01)
(07)	0.954(08)	0.617(08)
(08)	0.968(16)	0.396(16)
(09)	0.935(20)	0.678(20)
(10)	0.901(10)	0.763(10)

Table 5.6: results for Tieng-Boles function through our method, with image projections

<i>test.im \ b.match</i>	1	2	3	4
(01)	0.951(06)	0.923(19)	0.805	0.734
(02)	0.915(02)	0.848(10)	0.762	0.755
(03)	0.949(19)	0.886(18)	0.817	0.703
(04)	0.976(15)	0.890(14)	0.822	0.771
(05)	0.962(05)	0.894(01)	0.811	0.678
(06)	0.938(01)	0.883(18)	0.807	0.665
(07)	0.928(08)	0.894(09)	0.841	0.765
(08)	0.935(16)	0.847(15)	0.814	0.705
(09)	0.927(07)	0.881(20)	0.803	0.788
(10)	0.919(10)	0.893(09)	0.812	0.679

Table 5.7: Low-noise level experiment results for implicit polynomials

<i>test.im \ b.match</i>	1	2	3	4
(01)	0.738(06)	0.706(19)	0.685	0.596
(02)	0.715(02)	0.682(10)	0.609	0.588
(03)	0.678(18)	0.663(19)	0.584	0.535
(04)	0.764(15)	0.698(14)	0.633	0.570
(05)	0.814(05)	0.742(14)	0.684	0.613
(06)	0.808(01)	0.700(19)	0.643	0.528
(07)	0.703(09)	0.588(08)	0.449	0.401
(08)	0.810(16)	0.731(15)	0.675	0.566
(09)	0.726(07)	0.718(20)	0.662	0.541
(10)	0.712(10)	0.655(09)	0.606	0.558

Table 5.8: High-noise level experiment results for implicit polynomials

<i>test.im \ b.match</i>	1	2
(01)	0.189(06)	0.224(13)
(02)	0.174(02)	0.191(08)
(03)	0.267(18)	0.273(19)
(04)	0.206(15)	0.214(14)
(05)	0.143(05)	0.372(15)
(06)	0.120(01)	0.226(18)
(07)	0.195(08)	0.235(09)
(08)	0.179(16)	0.253(11)
(09)	0.284(06)	0.290(20)
(10)	0.342(10)	0.394(08)

Table 5.9: the results at low-noise level for Moment method

<i>test.im \ b.match</i>	1	2
(01)	0.356(06)	0.398(13)
(02)	0.387(03)	0.421(02)
(03)	0.467(18)	0.511(19)
(04)	0.336(15)	0.379(16)
(05)	0.345(05)	0.413(14)
(06)	0.277(01)	0.356(19)
(07)	0.295(08)	0.335(09)
(08)	0.348(16)	0.382(14)
(09)	0.355(07)	0.393(20)
(10)	0.410(10)	0.461(08)

Table 5.10: the results at high-noise level for Moment method

<i>test.im \ b.match</i>	1	2
(01)	0.215(06)	0.242(09)
(02)	0.230(02)	0.265(09)
(03)	0.284(18)	0.302(19)
(04)	0.304(15)	0.329(14)
(05)	0.214(05)	0.282(13)
(06)	0.258(01)	0.391(08)
(07)	0.273(08)	0.308(10)
(08)	0.205(16)	0.332(14)
(09)	0.345(06)	0.360(20)
(10)	0.318(10)	0.406(03)

Table 5.11: the results at low-noise level for Fourier descriptors

<i>test.im \ b.match</i>	1	2
(01)	0.262(06)	0.294(09)
(02)	0.318(02)	0.365(09)
(03)	0.351(18)	0.410(19)
(04)	0.369(15)	0.422(14)
(05)	0.307(05)	0.390(13)
(06)	0.299(01)	0.435(08)
(07)	0.343(08)	0.388(10)
(08)	0.325(16)	0.418(14)
(09)	0.395(06)	0.438(20)
(10)	0.367(10)	0.442(03)

Table 5.12: the results at high-noise level for Fourier descriptors

II

For the experiments, we obtained the test images by applying affine transformation to the randomly selected 10 model images in Figure 3.5. The model images , 1,2,5,6,8,10,15,16,19,20 are subjected to the affine transformation:

$$T = 2^{-k/2} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a & b \\ 0 & 1/a \end{bmatrix} \quad (5.1)$$

Parameters for the test images are given below. First column shows the test image. Second column is the model image from which test image is produced via the affine transformation. Other columns give the parameters of the affine transformation given by above formula.

<i>test.im</i>	<i>model.im</i>	θ	a	b
1	6	100	1	0.2
2	2	-10	1	-0.1
3	19	-50	1	1
4	15	30	1	-0.2
5	5	70	1	0
6	1	150	1	0
7	8	30	1	0.1
8	16	60	0.8	0.2
9	20	-30	1	-0.1
10	10	-70	1.4	0.2

Bibliography

- [1] R.C.Gonzales and R.E.Woods,Digital Image Processing. Addison-Wesley,1993.
- [2] J.L.Mundy and A.Zisserman, Geometric Invariance in Computer Vision. Cambridge,Mass.:MIT Press,199.
- [3] K.Arbter,W.E Synder, H. Burkhardt, and G.Hirzinger," Application of Affine Invariant Fourier Descriptors to Recognition of 3-D Objects ", IEEE Trans. Pattern Analysis and Machine Intelligence,vol.12,no. 7,pp.640-647,july 1990.
- [4] T.Crimmins," A Complete Set of Fourier Descriptors for Two-dimensional Shapes",IEEE Trans.Syst.,Man,Cybern.,vol.SMC-12,no. 6,1982.
- [5] Fenske A. and Burkhardt H. " Affine Invariant Recognition of Gray Scale Objects by Fourier Descriptors", Proceedings of the SPIE,1567:53-64,1991.
- [6] C.C. Lin and R. Chellepa, " Classification of Partial 2-D Shapes Using Fourier Descriptors",in Proc. IEEE Conf. Comput. Vision and Pattern Recognition, Miami Beach,FL,June 1986.
- [7] J. Flusser and T.Suk,"Pattern Recognition by Affine Moment Invariants ", Pattern Recognition, vol.26,no.1,pp. 167-174,1993.
- [8] M.K. Hu, "Visual Pattern Recognition by Moment Invariants ", IRE Trans. Inform. Theory, vol.IT-8,pp. 179-187,Feb. 1962.
- [9] H.Freeman, " Shape Description via the Use of Critical points",Pattern Recognition,vol.10,np.3,pp.159-166,1978.

- [10] T.H. Reiss, "The Revised Fundamental Theorem of Moment Invariants", IEEE Trans. Pattern analysis and Machine Intelligence, vol.13, no.8, pp. 830-834, Aug. 1991.
- [11] D.Keren, D.B. Cooper and J. Subrahmonia, "Describing Complicated Objects by Implicit Polynomials", IEEE Trans. Pattern analysis and Machine Intelligence, pp. 38-53, Jan. 1994.
- [12] Z. Lei, M.M. Blane and D.B. Cooper, "3L Fitting of Higher Degree Implicit Polynomials", In Proceedings of the Third IEEE Workshop on Applications of Computer Vision, Sarasota, FL, December 1996.
- [13] T.W. Sederberg and D.C. Anderson, "Implicit Representation of Parametric Curves and Surfaces", Computer Vision, Graphics and Image Processing, pp. 72-84, 1984.
- [14] Z. Lei and D.B. Cooper, "Linear Programming Fitting of Implicit Polynomials", Under review for IEEE Trans. Pattern analysis and Machine Intelligence, May 1996.
- [15] Q.M. Tieng and W.W. Boles, "Wavelet-Based Affine Invariant Representation: A Tool for Recognizing Planar Objects in 3D Space", IEEE Trans. Pattern analysis and Machine Intelligence, vol.19, no.8, pp. 846-857, Aug. 1997.
- [16] M.I. Khalil and M.M. Bayoumi, "A Dyadic Wavelet Affine Invariant Function for 2D Shape Recognition", IEEE Trans. Pattern analysis and Machine Intelligence, vol.23, no.10, pp. 1152-1154, Oct. 2001.
- [17] H.Civi, "Implicit Algebraic Curves and Spaces for Shape Modelling and Recognition": PhD thesis, Bogazici University, 1992.
- [18] H.Civi, A.Ercil, "Affine Invariant 3L Fitting of Implicit Polynomials", Pattern Recognition and Image Analysis, vol.13, No.3, 2003, pp.489-494.
- [19] Z.H. Huan and F.S. Cohen, "Affine-invariant B-spline Moments for Curve matching", Image Processing, vol.5, no.10, pp. 1473-1480, Oct. 1996.

- [20] S.A. Dudani, K.J. Breeding and R.B McGhee, "Aircraft Identification by Moment Invariants", IEEE trans. Comput., vol. C-26, pp. 39-46, Jan. 1977.
- [21] A.P. Reeves, R.J. Prokop, S.E Andrews and P.K Kuhl, "Three Dimensional Shape Analysis Using Moments and Fourier Descriptors", IEEE Trans. Pattern analysis and Machine Intelligence, 12:640-647, July 1990.
- [22] P.S. Modenov and A.S. Parkhomenko, Geometric Transformations, vol.1. New York: Academic Press, 1965.
- [23] I. Weiss, "Geometric Invariants and Object Recognition", Int'l J. Computer Vision, vol.10, no.3, pp.207-231, 1993.
- [24] H.W. Guggenheimer, Differential Geometry. New York: McGraw-Hill: 1963.
- [25] Q.M Tieng and W.W. Boles, "Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation", IEEE Trans. Pattern analysis and Machine Intelligence, vol.19, no.8, pp. 910-916, Aug. 1997.
- [26] M.I. Khalil and M.M. Bayoumi, "Invariant 2D Object Recognition Using the Wavelet Modulus Maxima", Pattern Recognition Letters, vol.21, no.9, pp.863-872, 2000.
- [27] A.E. Cetin, R. Ansari, "Signal Recovery from Wavelet Transform Maxima", IEEE Trans. Signal Processing, vol.42, no.1, pp. 194-196, Jan. 1994.
- [28] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. Pattern analysis and Machine Intelligence, vol.11, no.7, pp. 674-693, July 1989.
- [29] S. Mallat, "Zero Crossings of a Wavelet Transform" IEEE Trans. Inform. Theory, vol.37, no. 4, pp.1019-1033, 1991.
- [30] T.H. Reiss, Recognizing Planar Objects Using Invariant Image Features. Berlin: Springer-Verlag, 1993.
- [31] J. Naas and H. Schmid, Mathematisches Wörterbuch. Berlin: Akademie-Verlag, 1961.

- [32] J.M.Combes ,A.Grossmanand P.Tchamitchian,Eds.,Wavelets:Time-Frequency Methods and Phase Space, Berlin:Springer ,IPTI,1989.
- [33] A.Grossman,R.Krnonland-martinet adn J.Morlet,"Reading and Understanding Continious Wavelet Transforms " in wavelets: Time-Frequency Methods and Phase Space. Berlin:Springer,1989.
- [34] T. Pavlidis,"Algorithms for Shape Analysis of Contours and Waveforms" ,IEEE Trans.Pattern Analysis and Machine Intelligence,vol. 2,no.4,pp. 301-312,Apr. 1980.
- [35] G.Strang and T.Nguyen, Wavelets and Filterbanks , Wellesley-Cambridge press: 1996.
- [36] E.Cetin, E.Bala,"Computationally Efficient Wavelet Affine Invariant Functions for Shape Recognition". IEEE conf. ICIP 2003,Barcelona,2003.

Bibliography

- [1] R.C.Gonzales and R.E.Woods,Digital Image Processing. Addison-Wesley,1993.
- [2] J.L.Mundy and A.Zisserman, Geometric Invariance in Computer Vision. Cambridge,Mass.:MIT Press,199.
- [3] K.Arbter,W.E Synder, H. Burkhardt, and G.Hirzinger," Application of Affine Invariant Fourier Descriptors to Recognition of 3-D Objects ", IEEE Trans. Pattern Analysis and Machine Intelligence,vol.12,no. 7,pp.640-647,july 1990.
- [4] T.Crimmins," A Complete Set of Fourier Descriptors for Two-dimensional Shapes",IEEE Trans.Syst.,Man,Cybern.,vol.SMC-12,no. 6,1982.
- [5] Fenske A. and Burkhardt H. " Affine Invariant Recognition of Gray Scale Objects by Fourier Descriptors", Proceedings of the SPIE,1567:53-64,1991.
- [6] C.C. Lin and R. Chellepa, " Classification of Partial 2-D Shapes Using Fourier Descriptors",in Proc. IEEE Conf. Comput. Vision and Pattern Recognition, Miami Beach,FL,June 1986.
- [7] J. Flusser and T.Suk,"Pattern Recognition by Affine Moment Invariants ", Pattern Recognition, vol.26,no.1,pp. 167-174,1993.
- [8] M.K. Hu, "Visual Pattern Recognition by Moment Invariants ", IRE Trans. Inform. Theory, vol.IT-8,pp. 179-187,Feb. 1962.
- [9] H.Freeman, " Shape Description via the Use of Critical points",Pattern Recognition,vol.10,np.3,pp.159-166,1978.

- [10] T.H. Reiss, "The Revised Fundamental Theorem of Moment Invariants", IEEE Trans. Pattern analysis and Machine Intelligence, vol.13, no.8, pp. 830-834, Aug. 1991.
- [11] D.Keren, D.B. Cooper and J. Subrahmonia, "Describing Complicated Objects by Implicit Polynomials", IEEE Trans. Pattern analysis and Machine Intelligence, pp. 38-53, Jan. 1994.
- [12] Z. Lei, M.M. Blane and D.B. Cooper, "3L Fitting of Higher Degree Implicit Polynomials", In Proceedings of the Third IEEE Workshop on Applications of Computer Vision, Sarasota, FL, December 1996.
- [13] T.W. Sederberg and D.C. Anderson, "Implicit Representation of Parametric Curves and Surfaces", Computer Vision, Graphics and Image Processing, pp. 72-84, 1984.
- [14] Z. Lei and D.B. Cooper, "Linear Programming Fitting of Implicit Polynomials", Under review for IEEE Trans. Pattern analysis and Machine Intelligence, May 1996.
- [15] Q.M. Tieng and W.W. Boles, "Wavelet-Based Affine Invariant Representation: A Tool for Recognizing Planar Objects in 3D Space", IEEE Trans. Pattern analysis and Machine Intelligence, vol.19, no.8, pp. 846-857, Aug. 1997.
- [16] M.I. Khalil and M.M. Bayoumi, "A Dyadic Wavelet Affine Invariant Function for 2D Shape Recognition", IEEE Trans. Pattern analysis and Machine Intelligence, vol.23, no.10, pp. 1152-1154, Oct. 2001.
- [17] H.Civi, "Implicit Algebraic Curves and Spaces for Shape Modelling and Recognition": PhD thesis, Bogazici University, 1992.
- [18] H.Civi, A.Ercil, "Affine Invariant 3L Fitting of Implicit Polynomials", Pattern Recognition and Image Analysis, vol.13, No.3, 2003, pp.489-494.
- [19] Z.H. Huan and F.S. Cohen, "Affine-invariant B-spline Moments for Curve matching", Image Processing, vol.5, no.10, pp. 1473-1480, Oct. 1996.

- [20] S.A. Dudani, K.J. Breeding and R.B McGhee, "Aircraft Identification by Moment Invariants", IEEE trans. Comput., vol. C-26, pp. 39-46, Jan. 1977.
- [21] A.P. Reeves, R.J. Prokop, S.E Andrews and P.K Kuhl, "Three Dimensional Shape Analysis Using Moments and Fourier Descriptors", IEEE Trans. Pattern analysis and Machine Intelligence, 12:640-647, July 1990.
- [22] P.S. Modenov and A.S. Parkhomenko, Geometric Transformations, vol.1. New York: Academic Press, 1965.
- [23] I. Weiss, "Geometric Invariants and Object Recognition", Int'l J. Computer Vision, vol.10, no.3, pp.207-231, 1993.
- [24] H.W. Guggenheimer, Differential Geometry. New York: McGraw-Hill: 1963.
- [25] Q.M Tieng and W.W. Boles, "Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation", IEEE Trans. Pattern analysis and Machine Intelligence, vol.19, no.8, pp. 910-916, Aug. 1997.
- [26] M.I. Khalil and M.M. Bayoumi, "Invariant 2D Object Recognition Using the Wavelet Modulus Maxima", Pattern Recognition Letters, vol.21, no.9, pp.863-872, 2000.
- [27] A.E. Cetin, R. Ansari, "Signal Recovery from Wavelet Transform Maxima", IEEE Trans. Signal Processing, vol.42, no.1, pp. 194-196, Jan. 1994.
- [28] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. Pattern analysis and Machine Intelligence, vol.11, no.7, pp. 674-693, July 1989.
- [29] S. Mallat, "Zero Crossings of a Wavelet Transform" IEEE Trans. Inform. Theory, vol.37, no. 4, pp.1019-1033, 1991.
- [30] T.H. Reiss, Recognizing Planar Objects Using Invariant Image Features. Berlin: Springer-Verlag, 1993.
- [31] J. Naas and H. Schmid, Mathematisches Wörterbuch. Berlin: Akademie-Verlag, 1961.

- [32] J.M.Combes ,A.Grossmanand P.Tchamitchian,Eds.,Wavelets:Time-Frequency Methods and Phase Space, Berlin:Springer ,IPTI,1989.
- [33] A.Grossman,R.Krnonland-martinet adn J.Morlet,"Reading and Understanding Continious Wavelet Transforms " in wavelets: Time-Frequency Methods and Phase Space. Berlin:Springer,1989.
- [34] T. Pavlidis,"Algorithms for Shape Analysis of Contours and Waveforms" ,IEEE Trans.Pattern Analysis and Machine Intelligence,vol. 2,no.4,pp. 301-312,Apr. 1980.
- [35] G.Strang and T.Nguyen, Wavelets and Filterbanks , Wellesley-Cambridge press: 1996.
- [36] E.Cetin, E.Bala,"Computationally Efficient Wavelet Affine Invariant Functions for Shape Recognition". IEEE conf. ICIP 2003,Barcelona,2003.