INVARIANT OBJECT RECOGNITION

BY

AHMED ABDUL MUJEEB QHUSRO

A THESIS PRESENTED TO THE

DEANSHIP OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

KING FAHD UNIVERSITY
OF PETROLEUM & MINERALS

DECEMBER, 2004

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **AHMED ABDUL MUJEEB QHUSRO** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

Thesis Committee

_____
Dr. Muhammad Sarfraz (Advisor)

_____
Dr. Muhammed Al-Mulhem (Member)

_____
Dr. Mohammad Deriche (Member)

_____
Dr. Kanaan Abed Faisal
(Department Chairman)

_____
Dr. Mayez Al-Mouhamed (Member)

_____
Dr. Mohammad Abdul-Aziz Al-Ohali
(Dean of Graduate Studies)

_____
Dr. Hussein Al-Muallim (Member)

_____
Date

ACKNOWLEDGMENT

In the name of Allah, Most Gracious, Most Merciful

All thanks are due to Almighty Allah (Subhanahu Wa Taalaa) who gave me courage and patience to carry out this work. Peace and blessing of Allah be upon last Prophet Muhammad (Peace Be Upon Him).

Acknowledgment is due to King Fahd University of Petroleum & Minerals for supporting this research.

I wish to thank my advisor Dr. Muhammad Sarfraz for his continuous advise, support and encouragement throughout this work. I also thank my thesis committee members, Dr. Muhammed Al-Mulhem, Dr. Mohammad Deriche, Dr. Mayez Al-Mouhamed, and Dr. Hussein Al-Muallim for their help, support, and useful comments throughout.

I also acknowledge my colleagues and friends as I had a pleasant, enjoyable and fruitful company with them. Specially, I would like to thank my friends Abid-ur-Rahman, Abubakar, Adnan, Humayun, Siraj, for their continuous encouragement and support.

Thanks are due to my flat mates, Kashif, Sohel Khan, Riazuddin, Mujahed, Ismail, Junaid, Shafi, Moied for providing me a wonderful company.

Finally, I wish to express my gratitude to my family members for being patient with me.

TABLE OF CONTENTS

Page

List of Tables ..... **vi**

List of Figures ..... **vii**

Thesis Abstract ..... **x**

Thesis Abstract (Arabic) ..... **xi**

CHAPTER ONE **1**

INTRODUCTION **1**

1.1 Motivation 1

1.2 Introduction to Object Recognition 2

1.3 Main Contributions 10

1.4 Organization of thesis 11

CHAPTER TWO **12**

LITERATURE REVIEW **12**

2.1 Introduction 12

2.2 Related Work 12

CHAPTER THREE **35**

OBJECT RECOGNITION USING MOMENT **35**

INVARIANTS **35**

3.1 Introduction 35

3.2 Translation, rotation and scale invariant moments 36

3.3 Algorithm 39

3.4 Results and Analysis 40

3.5 Scope and Limitations 43

CHAPTER FOUR **44**

OBJECT RECOGNITION USING FOURIER **44**

DESCRIPTORS **44**

4.1 Introduction 44

4.2 Implementation Strategy 45
4.3 Algorithm 49
4.4 Results and Analysis 50
4.5 Scope and Limitations 53
CHAPTER FIVE 54
OBJECT RECOGNITION USING SHAPE SPACE APPROACH 54
5.1 Representation of 2d Shape 54
5.2 Landmarks Extraction 56
5.3 Classification of un-occluded objects 57
5.4 Classification of Occluded objects 58
5.5 Experimental Description and intermediate results 59
5.6 Time Complexity 63
5.7 Comparison with Other Experiments 63
CHAPTER SIX 67
OBJECT RECOGNITION USING INDEXING APPROACH 67
6.1 Introduction 67
6.2 Introduction to Indexing approach 67
6.3 Corner Detection 69
6.4 Feature Extraction 74
6.5 Building the model library 75
6.6 Hypothesizer 75
6.7 Verifier 77
6.8 Results and Analysis 79
6.9 Results comparison 85
6.10 Performance Comparison 87
6.11 Time Complexity 88
CONCLUSION 89
7.1 Introduction 89
7.2 Summary and Contributions 89
7.3 Limitations 90
7.4 Future work 91
Appendix A 92
References 100

LIST OF TABLES

Table Page

Table 1: Difference between parametric and non-parametric approaches 7

Table 2: Comparison of raw algorithm to enhanced algorithm..... 15

Table 3: Effect of noise on recognition (Moments Method)..... 42

Table 4: Similarity measures after the alternations 52

Table 5: Effect of noise on recognition (Fourier Descriptors Method)..... 53

Table 6: Comparison of results for 20 model objects..... 64

Table 7 : Effect of noise on corner points 82

Table 8: Effect of noise on recognition rate 83

Table 9: Comparison of recognition rates of various experiments..... 86

LIST OF FIGURES

Figure Page

Figure 1: Taxonomy of different neural network architectures..... 10

Figure 2: Moment invariants under different transformations 42

Figure 3: Pictorial Description of method..... 48

Figure 4: Fourier descriptors of an object under different transformations 52

Figure 5: An object and its centroidal distance representation..... 59

Figure 6: Polynomial approximation of centroidal distance representation..... 59

Figure 7: Approximation using Legendre's polynomial of degree seven..... 60

Figure 8: Representation of landmarks on object's boundary 60

Figure 9: Object with noise 61

Figure 10: Object after applying median filter..... 61

Figure 11: Noisy Boundary 62

Figure 12: Landmarks after approximation..... 62

Figure 13: Representation of landmarks on objects boundary..... 62

Figure 14: Representation of landmarks on occluded objects, with little change in center of gravity. 63

Figure 15: Effect of similarity transformations on accuracy..... 65

Figure 16: Effect of similarity transformations and noise on accuracy..... 65

Figure 17: Effect of similarity transformations and occlusion on accuracy 66

Figure 18: Effect of similarity transformations, noise and occlusion on accuracy 66

Figure 19: Flow Chart of Corner Detection Algorithm 71

Figure 20: Object one with corner points 72

Figure 21: Object two with corner points 73

Figure 22: Object three with corner points 73

Figure 23: Object four with corner points 73

Figure 24: Object five with corner points 74

Figure 25: Segment Descriptors 74

Figure 26: Generation of Hypothesis 76

Figure 27: Block Diagram of Hypothesizer 77

Figure 28: Marking of Segments 78

Figure 29: Block Diagram of Verifier 78

Figure 30: Block Diagram of Algorithm 78

Figure 31: Effect of database size on recognition time 80

Figure 32: Effect of database size on accuracy in case of similarity transformations ... 81

Figure 33: Effect of database size on accuracy in case of similarity transformations and occlusion 81

Figure 34: Effect of database size on accuracy in case of similarity transformations and noise 84

Figure 35: Effect of database size on accuracy in case of similarity transformations, noise and occlusion..... 85

Figure 36: Comparison of recognition rates of different experiments in case of similarity transformations..... 86

Figure 37: Comparison of accuracies in case of similarity transformations & noise..... 87

Figure 38: Performance comparison of different experiments..... 88

THESIS ABSTRACT

Name: Ahmed Abdul Mujeeb Qhusro

Title: Invariant Object Recognition

Major Field: MASTER OF SCIENCE

Date of Degree: December 2004

In many image analysis and computer vision applications, object recognition is the ultimate goal. Our experiments on object recognition are on recognizing the isolated objects under the circumstances of similarity transformations and in presence of noise and occlusion. In the past many techniques have been proposed for recognition incase of similarity transformations as well as in presence of noise and local distortions. In this research, we have used the moments, Fourier descriptors, landmarks on the boundary, local shape descriptors as the features in four different experiments. The minimum Euclidian distance was used as a classifier in the first three experiments, and an efficient indexing scheme was used in the fourth experiment in hypothesis generation and verification phases. The first two experiments used global features and hence they suffered in case of occlusion. An attempt to solve the problem of occlusion is made by extracting points called landmarks situated far away from each other by using Legendre's polynomials. In the fourth approach, the boundary is segmented by using corner points, and invariant local shape descriptors were used to define the segments. Index tables were built for model objects, and were used in the hypothesis and verification phases.

الرسالة ملخص

الاســـــــــــم : **أحمد عبدالمجيب قصرو**

العنـــــــــوان : **التعرف على الكائن ذو الشكل الثابت**

التخـــــــصص : **علوم الحاسب الآلي**

تاريخ التخـــــرج : **ديسمبر ٢٠٠٤**

في العديد من تطبيقات تحليل الصور وبصرية الحاسوب computer vision فإن التعرف على الكائن هو الهدف المنشود. تجاربنا في التعرف على الكائن بُنيت على التعرف على الكائنات المنعزلة تحت ظروف التحويلات المماثلة وغياب التشويش و الانسداد. في السابق ، اقترحت العديد من الطرق للتعرف على الكائن في حالة التحويلات المماثلة بالإضافة إلى غياب التشويش والتشويه. استخدمنا في هذا البحث Moments ومفسر Fourier و علامات الحدود landmark ومفسرات الشكل المحلي كملامح في أربع تجارب مختلفة. طريقة الأقل مسافة أوكليدية استخدمت كمصنفة في التجارب الثلاثة الأولى و خطة فهرسة فعالة استخدمت في التجربة الرابعة عند مراحل تكوين الفرضيات والاثباتات. استخدمت التجريتان الأوليتان الملامح العامة وبالتالي فإنهما تعانيان من الانسداد. وكمحاولة لحل مشكلة الانسداد صنعت باستخلاص النقاط المسماه علامات الحدود landmark والواقعة بعيدة عن بعضها البعض بواسطة Legendre's Polynomials. في الطريقة الرابعة ، جزءت الحدود باستخدام نقاط الزاوية واستخدمت مفسرات الشكل المحدد لمعرفة تلك الأجزاء. بنيت جداول فهرسية لنمذجة الكائنات واستخدمت هذه الجداول في مراحل الفرضيات والاثباتات.

CHAPTER ONE

INTRODUCTION

1.1 Motivation

Object recognition is becoming the major goal in industrial as well as military applications. Due to the rapid development of machine vision and advancement in image processing and computer vision, a great boost to object recognition applications has taken place.

Pattern recognition is a main problem in Computer vision. It is a crucial step in many areas such as medicine, biology, geology, robotics, astronomy, and others. Its applications go from the simple classification of needs to the medical diagnostic assisted by a computer.

Using invariants such as moments and Fourier descriptors are invariant in 2d transformations. Despite its success in some applications it has certain limitations.

Occlusion is the severe shape distortion. When the shape gets distorted, the global features don't work for recognition. So, we require the contour to be segmented in such as way that it is consistent in both occluded and un-occluded condition. Then we extract the local features of these segments and do matching.

The segmentation of contour and extraction of features which are invariant to similarity transformations is not an easy task. Further, partial shape matching using these local features efficiently is also a difficult task, which prompted me to take-up this work.

1.2 Introduction to Object Recognition

Pattern Recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of patterns. Object recognition is also an ultimate goal for most of the image analysis and computer vision applications. An object recognition system finds objects in the real world from an image of the world, using object models which are known a priori. This task is surprisingly difficult.

Objects can be recognized despite variations in actual or apparent size. The size of an object, such as the airplane, does not change the structural description of an object. When an object is moved to a new position in the environment, a different portion of the retina is stimulated. However, modest changes in position do not disrupt recognition accuracy in human subjects. That is, object recognition is translational invariant. Translational invariance indicates that people do not learn to recognize an object on the basis of the absolute position in the environment or its position relative to other objects. This explains why object recognition can occur even when objects are partially occluded. Finally, there is good evidence for rotational, size, and translational invariance in people.

In order to recognize occluded and noisy objects, we require the robust contour segmentation which should be consistent in case of un-occluded and occluded objects, and a strong local feature set invariant to similarity transformations is required.

Features represent the smallest set that can be used for discrimination purposes and for a unique identification of each object. Features of the object are extracted and matched with the existing ones to identify the object under investigation. Selection of a feature extraction method is one of the most important factor in achieving high recognition performance. Devijver and Kittler [23] define Feature Extraction as the problem of "Extracting from the raw data the information which is most relevant for classification purposes, in the sense of minimizing within-class pattern variability while enhancing between class pattern variability."

Extracting the meaningful features is the major part in the recognition process. In case of recognizing the object under similarity transformations, we require the features that are invariant under all the similarity transformations (translation, rotation and scale). Many cues were proposed such as color, texture and motion etc. Shape is perhaps the most common and dominant. Shape-based approaches extract a representation of the shape of the object and object similarity is measured by comparing these representations.

Recognition of objects independent of their position, size, orientation and other variations in geometry and colors has been the goal of much recent research. Finding efficient invariant object descriptors is the key to solving this problem. In the past, several groups of features have been used for this purpose, such as simple visual features (edges,

contours, textures, etc.), Fourier coefficients, differential invariants, and moment invariants, and others.

Generally, the most reliable type of object information that is available from an image is *geometric* information. So object recognition systems draw upon a library of geometric models, containing information about the shape of known objects. An invariant of a geometric configuration is a function of the configuration whose value is unchanged by a particular transformation. For example, the distance between two points is unchanged for a Euclidean transformation (translation or rotation).

We always obtain a silhouette of an object, and its extracted outline or contour is usually a closed curve. The object recognition step in computer vision makes use of the features that can be extracted from the contour. Image features, such as CORNERs, ENDs, ARCs and LINEs, *etc*., which are located on the outline of the object, always make great sense and are taken as the most important representatives to the shape of an object by human beings.

A number of techniques have been developed to extract features that are invariant to object translation, scale change and rotation. The moment invariant descriptors developed by Hu [1], Zernike moments have been used by several authors for character recognition of binary solid symbols [24, 25, 26]. However, Initial experiments suggest that they are well suited for gray-scale character sub images as well. Both Rotation –Variant and Rotation Invariant features can be extracted. Features invariant to illumination need to be developed for these features to be really useful for gray level character images. The Fourier descriptors of the boundary for recognizing closed contours is proposed in [4].

After the extraction of features, we classify or recognize the object based on features extracted. Many pattern recognition approaches fall in this category, and their potential has been demonstrated in many applications. Neural net-based approaches also fall in this class. In all cases, it is assumed that *N* features have been detected from an image and that these features have been normalized so that they can be represented in the same metric space. It is also assumed that the detected features for an object can be represented as a point in the *N*-dimensional feature space defined for that particular object recognition task. To decide the class of object, we measure its similarity with each class by computing its distance from the points representing each class in the feature space and assign it to the nearest class. The distance may be the well-known Euclidean or any other.

The process of recognizing a pattern is basically dealt with three approaches: Statistical, Syntactic and Neural [21]. Statistical approach is based on decision making (i.e., the probabilistic model), Syntactic deals with the structural description of pattern (i.e., formal grammar) and Neural approach is based on training the system with a large data set of input and storing the weights (Stable State), which is used later for recognition of trained patterns. For unsupervised training, such correct classifications are not available, and grouping must be made from the data.

Statistical Approach

There are two general ways to do supervised training: parametric and non-parametric approaches. The key difference is the form of the information “learned” by the training and passed on to the classifier. Parametric approaches assume that the patterns in the

training set fit a known statistical distribution. The form of the "learned" information is thus a set of parameters for that distribution, which is denoted by $\overline{\theta}$.

Non-parametric approaches are useful in cases where the underlying distribution cannot be easily parameterized. In such cases, we can approximate the distribution by the histogram of the training set.

In statistical approach each pattern is represented by *d* features represented in a *d*-dimensional space. Then the goal is to allow different categories of feature vectors to occupy the compact and disjoint regions in *d*-dimensional feature space. The effectiveness of the representation is determined by how well the patterns of different classes can be separated. Given a set of training patterns, the decision boundaries will be determined for different classes of patterns. In statistical decision theoretical approach the decision boundaries are determined by probability distributions of patterns belonging to different classes.

Table 1 summarizes the differences in between parametric and non-parametric approaches:

Table 1: Difference between parametric and non-parametric approaches

	Parametric	**Non-Parametric**
Classifies	Classifies Using distribution	Using histogram
Stores	Stores Estimated distribution parameters	Training set histogram
Advantages	More compact representation, Faster classifier	More storage required, Slower classification
Disadvantages	Assumes known distribution, Slower training	Works with any distribution, Faster training

Syntactic Approach

Syntactic pattern recognition is based on the structural information provided to the system. Syntactic pattern recognition has been applied to many practical pattern recognition problems, such as, character recognition, speech recognition, fingerprint recognition, remote sensing data analysis, biomedical data analysis etc., In the pattern recognition problems, besides the statistical approach, the structural information that describes the pattern is important. This information can be used to recognize the pattern. A pattern can be decomposed into simpler sub patterns, and each simpler sub pattern can be decomposed again into even simpler sub patterns, and so on. The simplest sub patterns are called primitives (symbols, terminals) [22]. A pattern can be described as a

representation, i.e., a string of primitives, a tree, a graph, an array, a matrix or an attributed string.

In recognition problems, involving complex patterns, where a pattern is composed of sub-patterns. The simplest sub-patterns to be recognized are called primitives, and the complex pattern is represented in terms of interrelationships between these primitives.

In syntactic pattern recognition, an analogy is drawn between structure of patterns and syntax of a language. Patterns are viewed as sentences belonging to a language and primitives are treated as alphabets of the language. So, a large collection of patterns can be described by a small set of primitives and grammatical rules.

Structural pattern recognition provides a description of how the patterns are constructed from the primitives. The implementation of syntactic approach leads to many difficulties such as segmentation of noisy patterns to detect primitives and to inference grammar from training data.

Neural Networks Approach

Artificial neural networks, which are also referred to as neural computation, connectionist models, and parallel distributed processing (PDP), are massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections between them. ANNs were designed with the goal of building ``Intelligent machines" to solve complex problems, such as pattern recognition and optimization, by mimicking the network of real neurons in the human brain [30].

An assembly of artificial neurons is called an artificial neural network. ANNs can be viewed as weighted directed graphs in which nodes are artificial neurons and directed edges (with weights) are connections from the outputs of neurons to the inputs of neurons. Based on the connection pattern (architecture), various ANNs can be grouped into two major categories as shown in Figure 1: Taxonomy of different neural network architectures: (a) *Feed Forward Neural Networks* in which no loop exists in the graph, and (b) *Feed Back or Recurrent Neural Networks* in which loops exist because of Feedback connections [31]. The most common family of Feed Forward networks is a layered network in which neurons are organized into layers with connections strictly in one direction from one layer to another. In fact, all the networks with no loops can be rearranged in the form of layered Feed Forward networks with possible skip-layer connections. Figure 1: Taxonomy of different neural network architectures also shows typical networks of each category. Generally speaking, Feed Forward networks are static networks, i.e., given an input, they produce only one set of output values, not a sequence of values. Feed Forward networks are memory less in the sense that the response of a Feed Forward network to an input is independent of the previous state of the network. An exception is the time delay Feed Forward network in which dynamics occurs because of different delay factors of the neurons in the network.

Recurrent networks are dynamic systems. Upon presenting a new input pattern, the Outputs of the neurons are computed. Because of the Feed Back paths, the inputs to each neuron are then modified, which leads the network to enter a new state. This process is

repeated until convergence. Obviously, different mathematical tools must be employed to treat these two different types of networks.

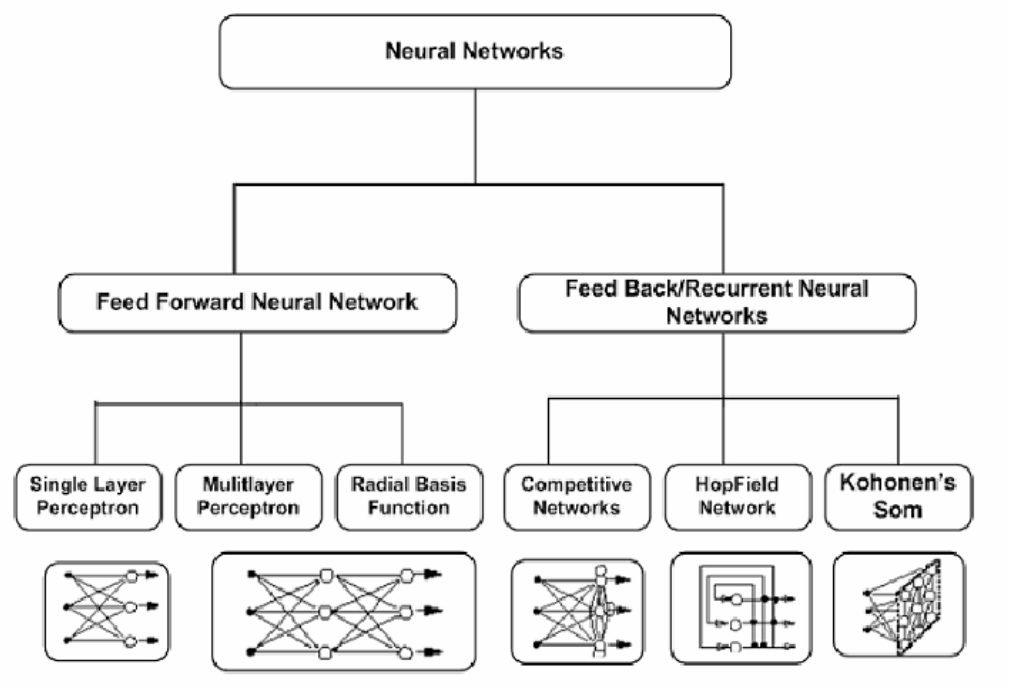


Figure 1: Taxonomy of different neural network architectures

1.3 Main Contributions

The main contributions in this thesis work are as follows:

1. Literature survey on 2d invariant object recognition.
2. Literature survey on 2-D Occluded object recognition.
3. Proposing a feature set that leads to efficient recognition under similarity transformations and occlusions.

4. Proposing an effective and efficient method for recognition of objects whose recognition time is non-linear to the database size.

1.4 Organization of thesis

The rest of the thesis is organized as follows. Chapter 2 gives an extensive literature survey of the related work. Chapter 3 presents the invariant object recognition using moment invariants. Chapter 4 presents the invariant object recognition using Fourier descriptors. Chapter 5 presents the invariant object recognition using shape space approach. Chapter 6 presents the invariant object recognition through an indexing approach. Chapter 7 gives the conclusion and future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In this chapter, we will present the work related to 2d invariant object recognition as well as papers related to occluded object recognition we have found in the literature.

2.2 Related Work

John W. Gorman, O Robert Mitchell [12] proposed a partial shape recognition technique utilizing the local features described by Fourier descriptors and a dynamic programming formulation. A polygonal approximation to the contour is found, and the vertices of this polygon used as break points of the contour. The contour is next divided into two sections at the two most distant points on the curve; the two end points are taken to be the two vertices of polygonal approximation. A line is drawn between the end points, and the point on the contour farthest from the line is located. The new point becomes another vertex and the half contour is split into two quarter contours about this new point. This procedure continues recursively until the distance between the line and the farthest point on the contour falls below a given threshold.

After contour splitting procedure is completed; the contour segments are formed using three of these vertices at a time. Each segment consisted of the chain code which describes the contour traced from the first vertex for that segment, through the second

vertex to the third vertex, and then back through second to the first. The contour segments were described by Fourier coefficients. Normalized Fourier coefficient was used to assure that the comparison will be independent of size, orientation, and contour starting point. By experiment he found that six Fourier coefficients are reasonable to describe the contour segment. The distance between the segments is defined as

$$d^2 = \sum_{i=-N}^{N} \left| x(i) - y(i) \right|^2$$

For comparing the two entire contours, an intersegment distance table is used. The columns represent the segments of one contour and the rows represent the segments of the other. Let the unknown contour have m segments represented by columns and let the known contour have n segments represented by rows in the intersegment distance table. The criterion for the path completeness is that the path must make use of all M segments of the unknown. Since partial shapes are being examined, the number of segments of an unknown contour that match with the corresponding known contour will be less than or equal to the number of segments in the unknown contour.

To find the minimum distance path or desired path, the directions of legal motion through the intersegment distance table, such as the path can only proceed down and to the right is formulated, the path is allowed to deviate from a diagonal path by no more than half of the number of segments in the known or unknown contour. Since it is not known that which segment of the known contour matches with the first segment of the unknown contour, the rows of the table are repeated resulting in 2N rows and M columns.

A dynamic programming formulation technique was used to find the minimum distance path in the intersegment distance table. The values of D(i,0), i=1,…,N represent the lengths of the minimum distance paths through the intersegment distance table from each starting point I in the known contour , running from the first segment to the last segment of the known contour. The path with the lowest D(i,0) is the minimum distance path through the distance table. Since the raw path length includes distances from non-matching pairings and gives no indication of the amount of the contour that actually matched . So, the following distance formulae was used to compare the partial contours

$$d = d'\left(\frac{1}{f^2} - \frac{1}{f} + 1\right)$$

d' is the sum of intersegment distances of the segment pairs that match. *f is* the fraction of the contour segments which got matched. The minimum distance path found by the dynamic programming technique is the minimum distance in the global sense. So, the dynamic programming method will produce the desired paths when the locally optimal path is also globally optimal path. An enhanced dynamic algorithm is also formulated to find the path that is both locally and globally optimum.

A library of 143 views of each of the six classes of aircraft were used. For each class of aircraft fifty unknown views were generated. To create partial shapes , the unknown contours were chopped , with chopped portions replaced by a straight line , a 90 degree turn and another straight line.

Table 2: Comparison of raw algorithm to enhanced algorithm

Percent Contour Chopped	Classification Accuracy	
	Raw Algorithm	Enhanced Algorithm
0	89.33	93.00
10	78.33	82.33
20	73.67	75.00
30	64.33	71.67
40	54.33	64.67
50	42.33	51.67

John W. Gorman, Richard M. Ulmer [13] used simulated annealing for partial shape recognition. Features are acquired from small segments of an object's contour. These small segments are obtained by splitting the contour of the object. The size of the segments is controlled by a threshold. Once the contour has been split into segments, the local features are described by Fourier descriptors. Fourier descriptors are calculated by finding the discrete Fourier transform for a segment. This transforms the segment representation from the space-domain into the frequency domain. Translating into frequency-domain representation eliminates the dependency on position, size, and orientation.

Two segments represented by Fourier descriptors are compared by finding the distance between the segments and the distance between them id represented in the intersegment distance table. The minimum distance path through the intersegment distance table is not directly used to determine the contour matching. This raw path includes the path through matching and non-matching pairings. It does not give any indication of how much and how well the contours match. The comparison is based on the distance over the portion of the path that matches along with the fraction of the contour covered by the matching segments.

$$d = d'\left(\frac{1}{f^2} - \frac{1}{f} + 1\right)$$

d is the new distance which is used to determine if the two contours match. d' is the sum of the intersegment distances along the path which match. f is the fraction of the contour covered by the matching segments. For finding the minimum distance path through the intersegment distance table simulated annealing was used. The four basic parts of simulated annealing algorithm used. A concise description of the configuration of the system, a method for randomly generating new configurations, a cost function to measure the amount of optimization, and an annealing schedule of temperatures.

Thirteen aircraft contours are chopped by 10, 20, 30, 40, and 50 percent starting at the same location of the contour. This produces 65 unknown contours to be matched with 1 of the 13 original known contours. Chopping is done by replacing a percentage of the contour with a straight line, a 90 degree turn, and another straight. The program failed to correctly identify an aircraft one time out of 65 tests. This is an accuracy of 98.5 percent.

G.J.Ettinger [14] proposed an object representation that is both hierarchical in structure (whole object to component sub-parts) and scale (gross to fine features).He introduced a system that exploits hierarchy of object structure and object scale (Coarse to fine). He focuses his attention to use these hierarchies to achieve robust recognition based on the optimization that allow him to use an indexing scheme for his model library. He develops an object shape representation that incorporates a component sub-part hierarchy. These sub-parts allow mutual relative parameterization and they are the building material for the model library. His implementations uses a representation based on significant contour curvature changes.

This structure hierarchy can contain an arbitrary number of levels, but is limited by the size of the smallest component. The smallest sub-part can contain just one feature, which cannot be broken down into smaller parts since features are generally the lowest level primitives used. Sub-parts consist of subsets of these features that partition the object into components. Once the sub-parts have been generated, the representation must specify the relationship between them. One characteristic of this relationship is the type of connection between the subparts. The connection consists of the description of the features separating the components and the point along each sub-part. Sub-parts consist of subsets of these features that partition the object into components. These hierarchies allow the system to stress the important features of the object and to reduce the large problem of recognizing the whole object into several much smaller problems of recognizing the object's sub-parts. The system thus demonstrates a recognition

behavior of focusing on the correct interpretation while reducing the combinatorics of the process.

In scale hierarchy a coarse to fine recognition scheme for reducing the fraction of the search space that is explored is used. The idea is to take advantage of the coarse shape cues first to derive rough interpretations and then use the finer features to refine them. An object should be described by its coarsest primitives, at the top level, and by more detailed primitives at lower levels.

Hypotheses and predictions are made at the level of sub-parts, while the actual recognition as well as the verification steps are accomplished using the scale hierarchy of the sub-part features. Hypotheses are generated by counting votes for each model sub-part in the library based on compatible features found in the scene. Each scene feature is checked against each feature of each sub-part. If the model and scene features are found to be compatible, then the model sub-part receives a vote. Only the coarsest features are used for generating the hypothesis.

The sub-parts with high vote scores are further processed by some heuristics in order to better identify the candidates to be recognized. These heuristics include sorting by sub-part size and degree of sharing in the library. This task consists of matching scene to model features in such a manner that the object an be identified in case of rotation, translation, scaling, and object reversal (mirror image).

The library-based recognition system was extensively tested using a library of model objects found on traffic signs. Thirteen model objects were presented to

the system for the purpose of constructing the library. These models consisted of the following sign objects: bike, car, disallow, right-turn, no-left turn, straight, U-turn, bend, junction, and the word objects: reduce, speed, now, parking. The average percentage of explored configurations for all ten sample scenes was 0.58%. When only the models similar to the ones actually in the scene are placed in the library, the average percentage of explored configurations increased to 2.4%. In order to check the relationship of recognition time to number of objects in the library, several tests were made with different size libraries. These libraries were composed of different sets of the model objects. The result indicated the sub-linear growth in the recognition time. Based on his paper, it's not clear how well the system works with occlusion.

Hong-Chih Liu and Mandyam D. Srinath [15] proposed an algorithm for partial shape recognition, which first calculates the curvature function from the digitized image of an object. The points of local maxima and minima extracted from the smooth curvature function which are used as control points to segment the boundary and to guide the boundary matching procedure. The boundary matching procedure considers two shapes at a time, one shape from the template data bank, and the other being the object under classification, The procedure tries to match the control points in the unknown shape to those of a shape from the template data bank, and estimates the translation, rotation, and scaling factors to be used to normalize the boundary of the unknown shape. The chamfer *3/4* distance transformation and a partial distance measurement scheme are used as the final step to measure the similarity between

these two shapes. The unknown shape is assigned to the class corresponding to the minimum distance.

Two sets of data were used in the classification experiments to test the effectiveness. The first set consists of four kinds of aircraft. The second data set consists of four classes of shapes which are the aerial views of lakes Erie, Huron, Michigan, and Superior. Partial shapes were generated by chopping 0, 10, 20, and 30% of the contour, and replacing the chopped out portion either by a straight line or by a straight line-90° turn-straight line to close the contour. For each class, ten partial shapes, each with different orientation location and size are collected for each percentage of distorted pixels on the boundary. The algorithm recognized partial shapes of lakes and aircrafts without any misclassification.

Nanning Zheng, Yaoyong Li [16] proposed a method for recognizing partial objects using local features and neural networks. In the proposed method the corners of the contour of an object are taken as local features. The corner is represented by a 8-elements vector, $a = (a_1, a_2, .., a_8)$, which we call the corner feature of the object.

a_1 -- the corner tag(type of corner). $a1 = 1, 0.5, -0.5$ or -1 means a line-line corner, a line- arc corner, an arc-line corner, or an arc-arc corner, respectively.

a_2 -- the value of the corner angle in degree, and divided by 180.0, so it is in the interval [-1,+1].

a_3 -- the tag of the corner's left segment.

$a_3 = 1, -0.5$ or -1 means a line-segment, a counterclockwise arc-segment, or a clockwise arc-segment, respectively.

a_4-- the value of the left segment's length, divided by *dm. Dm=maximum{*a_4,a_5,a_6,a_7,a_8*}*.

a5-- the value of the radius of the left segment, divided by *dm,* if the left segment is an arc; else $a_5 = 0$.

a_6, ,a_7 and a_8 represents the same as a_3, ,a_4 and a_s respectively, but for the corner's right segment.

The ANNFR(Artificial Neural Network For Recognition) consisting of 8 neurons in first layer and *k* (number of model objects) neurons in the output layer are used in the experiment to train and test the objects. The ANNFR is trained for five objects, namely, handle, key gear, cap, and screw, denoted as *ml,m2,m3,m4* and *m5,* respectively, Then the trained ANNFR is used to recognize the unknown object(s) in the image. The test image consisted of different objects overlapping each other. The author did not mention the recognition performance, but he concluded that the method is good for image having many corner features.

Califano and mohan [17] proposed an indexing algorithm which has two stages, firstly short range autocorrelation operators are used to map from the image pixels into a small set of simple localized shape descriptors. A global autocorrelation operator maps combinations of these local descriptors into invariant indices. The indices are used to address the cells of a global lookup table which contains the shape model representations.

They argue for the use of higher-dimensional spaces in indexing. Their analysis indicates a dramatic reduction in recognition time by increasing the size of the feature vectors. However, they again use hash tables for the lookup and do not search across bin boundaries. Thus, in their method a pose clustering stage is required to accumulate results,

presumably because of the high likelihood that a query will land in a different bin than that of the best neighbour. Because a number of separate groupings is required to initiate a hypothe-sis, and each grouping is already non-robust due to being high-dimensional, their method will lead to a large number of false negatives when real images are used.

They show extensive tests with database of up to 300arbitrary shapes. They demonstrated the fault tolerance of their system by systematically destroying random points of the shape memory and then evaluating recognition performance. When the size of the table is increased linear growth constant is so small that recognition time on a shape table with 100 objects is only 10% more than the one with a single object. By comparison, with a table 10 times smaller, the increase is of 123%. The memory required in the two cases, however, is the same since the number of entries stored in the table is not a function of its total size and sparse data can be represented in a compact form by means of a hashing paradigm.

Joong-Hwan Baek, Keith A. Teague [18] proposed local features such as corners, arcs, parallel-lines, and corner-arcs which were extracted from the preprocessed image and the hashing method was used in order to match the hypothesized objects. The extended local features from the image of each model object are extracted. The local features such as corners, arcs, parallel-lines, and corner-arcs from the preprocessed images are termed as extended local features. Once the extended local features are extracted from the image with a model object, the model object is modeled by the extracted features. A database of model objects contains the name of the object, number of each feature type, and data of the feature type. In order to recognize the objects in an input image, the feature extraction

is performed for the input image. Then one or more objects are hypothesized by searching the knowledge-base. Since the searching time increases linearly with the size of the knowledge-base, the *hashing* scheme is used to reduce the searching time. A simple hashing function using the modulo operator is applied. For hypothesizing the objects in an input image, the feature data from the input image are converted into keys, and then each key is searched from the knowledge-base in the same manner as the insertion. If a key is found, the objects listed under the key are hypothesized. The geometric transform using clustering, which brings a model point to the corresponding Image point is used to verify the hypothesized objects. The geometric transform maps the coordinates of a model features into the coordinates of an image features.

Ten different industrial tools were used for model objects and the occlusion ratio, the occluded area of the object to the area of the object is used. The objects were placed in such a way that one object overlaps the other. The matching is performed for the 20 synthetic images. The occlusion ratios for the 20 synthetic images range from 0.3 % to 100% and the average occlusion ratio is 35.8%. The number of the matched objects is 57 out of 74 objects. Thus overall matching rate is 77%. Also, 61 out of 74 objects (82.4%) are correctly hypothesized.

M. Al-Mouhamed [19] introduced an efficient indexing scheme for object storage and recognition. The approach is based on partitioning contours into a set of constantly curved segments, extracting descriptors from these coarse segments, and using the descriptors in pruning the models. The scene object's contour is segmented by polygonal approximation of the contour. The algorithm used for polygonal approximation consisted of detecting

breakpoints by comparing the average direction of a set of pixels to the gradient of the current reference segment. In other words, the method is based on successive conditional merging operations that are performed along the chain of directions until the least-squares error line fit of the merged directions thus exceeds a preset threshold. In this case, a breakpoint is detected, recorded, and a new segment is started. The process is continued until all chains have been visited.

A set of descriptors that are invariant with respect to scale, rotation, and translation are extracted from the geometric features of the segments. A set of eight local and global descriptors which are classified as straight segment descriptors and curved segment descriptors are used in DS-A. In DS-B, a set of four descriptors is defined on the basis of referring the current coarse segment S_i with respect to its previous segment S_{i-1}. To increase discriminability, four categories of segment connections are used to distinguish between different possible combination of straight and curved segments. For example, in type straight-curved (s-c), the current segment S_i is straight and the previous segment S_{i-1} is curved. The four possible types of the two-segment descriptors are: 1) straight-straight (s-s); 2) straight-curved (s-c); 3) curved-straight (c-s); and curved-curved (c-c).

The descriptors were used to carry out an efficient indexed search over the models so as to reduce the search space. Fragments of contours extracted from partially occluded scenes are individually matched by using the local shape descriptors. Pruning of large portions of the models is carried out by keeping only some matched classes which received the highest vote. This significantly

reduced the search and enabled the use of finer matching operators, such as comparing the positioning of segments in the scene to positioning of matched segments in the model. More sophisticated matching is applied in later stages over a much restricted number of hypotheses.

All of the studied objects are laboratory mechanical tools with different sizes and different shapes, having between 20-140 fine segments or between 10-50 coarse segments. Entirely visible objects are recognized with a reasonably high efficiency (80%), even with a change in viewpoint of up to 25°. The efficiency smoothly decreased, but remained above 60% when the percentage of visible segments drops to 50% and the change in viewpoint is same as above. The effects of increasing the library size on the recognition and classification of scenes of three objects under partial occluding is also studied. Each of the studied objects has between 10-30 coarse segments. The experiment is conducted by setting up *LB1O* and randomly selecting three scene objects. Four settings of the model are used: 1) 10 objects *(LB1O);* 2) 30 objects *(LB30);* 3) 60 objects *(LB60);* and 4) 100 objects *(LB1OO).* The experiment started by setting up *LB1O* and randomly selecting three scene objects. To build *LB30, 20 more* objects are randomly selected and added LB10, and so on. The recognition time slightly increased with the library size. However, the descriptors of DS-B have better selectivity and discrimination power than those of DS-A. The recognition time is sub-linear to the size of library.

M. Al-Mouhamed [20] proposed a gross to fine pattern recognition system, so that so that the recognition time would mainly depend on the scene complexity without explicit

dependence on the database size. Features are obtained by using a constant-curvature criterion and used to carry out efficient coarse-to-fine recognition. A robust shape matching is proposed for comparing contour fragments from scenes with partial occluding. In order to carry out an early pruning of a large portion of the models, hypotheses are only generated for a subset of contours with enough discriminative information. Poor scene contours are used later in validating or invalidating a relatively small set of hypotheses. Hypotheses are selectively verified and blocking is avoided by extending current matching through pairing of hypotheses, predictive matching, and retrieving the next weighted hypotheses. This avoided the processing of a large number of initial hypotheses.

The recognition time is studied as a function of model size. A model database of *sizes* 10, 40, 70, 100, and 130 are used. The recognition algorithm (seven-feature) was run for each of the three-object scenes. The recognition time recorded was independent to the number of hypotheses originally generated. The time increases are relatively small as the algorithm spent only 20% extra time when the database size becomes 13-fold that of *DBlO*. Partially occluded scenes having three objects are recognized with a success rate of 84%. The results are reproducible against changes in scale, rotation, and translation.

Mokhtarian [32] developed a complete object recognition system based on closed object silhouettes. The system is designed to recognize free-form objects that have only a few stable views in an environment where only one object will be present. To do this, a light box is used to illuminate the object and make the boundary between background and

object easier to detect, and objects are isolated by simple threshold. Boundary curves (extracted by contour following) are then represented by calculating its curvature scale space (CSS) representation. The matching of CSS curves is done based on the location of the maxima of the curvature zero-crossings. For convex boundary curves the CSS representation is smoothed until only four maxima points are remaining, while concave curves utilize all maxima obtained at a given scale. During recognition the aspect ratio of the object's silhouette is used to pre-filter possible scene/model matches before the silhouettes are matched. This technique provides a fast way of matching the coarse features of a scene silhouette with an object's silhouette. The best silhouette matches are then verified by registering the two curves and measuring the error.

The system is tested using a total of 22 model curves and 19 images. The following model contours were used to test the system: bottle, calculator, spray can lid, paper clip, fork, glue stick, key, monkey wrench (two sides), panda, two connector cases, screw driver, scissors, spoon, tape dispenser, vase, wire cutter and two wrenches (two sides each). All model contours were concave except the calculator. The system correctly identified all nineteen objects in its database.

Che-Bin Liu, Narendra Ahuja [10] proposed a method of using Fourier descriptors and temporal changes in Fourier coefficients to recognize the objects, synthesis of dynamic shape after learning from a given image sequence, and predict the contours of moving regions. The paper aimed at modeling gradual changes in the 2-D shape of an object. The 2-D region shape is represented in terms of the spatial frequency content of the region contour using Fourier coefficients. The temporal changes in these coefficients are used as

the temporal signatures of the shape changes also called autoregressive model of the coefficient series. A dynamic shape model presented describes shape at any given time using Fourier transform coefficients and an autoregressive (AR) model to capture the temporal changes in these coefficients. The Fourier description possesses boundary, numeric, and information preserving properties. The autoregressive model is a simple probabilistic model that has shown remarkable effectiveness in the mapping and prediction of signals. The efficacy of the model is demonstrated on several applications. First, the model parameters are used as discriminating features for object recognition and classification. Second, they showed the use of the model for synthesis of dynamic shape using the model learned from a given image sequence. Third, they showed the model can be used to predict contours of moving regions which can be used as initial estimates for the contour based tracking methods.

L. A. Torres-Mendez, *et all*. [35] proposed a method addressing the problem of similarity transformations. The method proposed has two steps: preprocessing and recognition. The first takes into account the moment of inertia of the object. The second step is done by using a holographic nearest-neighbor algorithm (HNN), where vectors obtained in the preprocessing stage are used as inputs to it. The first step of the method (preprocessing) is defined as the extraction of appropriate invariant features that are then used for recognition by a classification system. The moment of inertia depends on the position of the axis of rotation and on the shape and mass of the rotating object. It is invariant under translation and rotation. For binary images, the moment of inertia of the object with

respect to its centroid (central moment of inertia) is calculated by using the below formulae

$$I = \sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} ((x_i - C_x)^2 + (y_i - C_y)^2)$$

Where $C_x and\ C_y$ are the centroid coordinates, x_i, y_i the image pixel coordinates of the object, and N the total number of pixels in the object. Translation and rotation invariance is achieved by calculating the central moment of inertia. On the other hand, by dividing I by N^2 (we will name it IN), scaling invariance is achieved. The invariant features obtained are real numbers that are fed as vectors to the classification system. The HNN algorithm used is similar to the well-known nearest-neighbor algorithm (NN). HNN algorithm is based on the idea of the minimum Euclidean distance between the input and each training vector can be used to classify the input vector.

They considered two dimensional (2-D) binary images and tested the algorithm for invariant-character recognition. They advocated that the method could be extended easily for multilevel images. The results were presented for recognition of characters in the real images (grey scale) of car plates. The algorithm was tested in character recognition using the 26 upper-case letters of the alphabet. Four different orientations and only one size were used for training. Recognition was tested with 17 different sizes and 14 rotations for each size. Invariant object recognition is obtained with almost 100% accuracy on images with sizes between 100 × 100 and 45 × 45 pixels. An accuracy of 98% is obtained, with images having up to 60% of random noise. The performance of the model decreases slightly for smaller letters

S. Berretti, A .Del Bimbo, P. Pala [36] addressed the problem of retrieval by shape similarity, using local features and metric indexing. The system presented supports image retrieval by shape by combining an effective shape description with an efficient index structure. Shape is partitioned into tokens in correspondence of its main protrusions, according to curvature analysis. Each token is modeled by a set of perceptually salient attributes and two distinct distance functions are used to model token similarity and shape similarity. Shape indexing is obtained by arranging tokens into an **M** -tree index structure. Each token is represented as a point in a multidimensional feature space and tokens are organized according to an M -tree index structure. M-tree has been explicitly designed to act as a dynamic access method and combines the advantages of metric trees and database access methods by optimizing both distance computations and I/O costs. Thus, the structure supported efficient access to database items.

The experiments were carried out for a set of 20 shapes, representing the contours of different objects, such as bottles, horses, busts and vases. Each shape was subject to three degrees of occlusions: 0% (the original shape), 30% and 60% of the shape contour length. Robustness to occlusion has been measured by evaluating the extent to which the presence of a contour occlusion affects correct retrieval. The retrieval rate shown is in the form of a graph, which is approximately 59%, 65%, and 70% in case of 60%, 30%, and 0% occlusions.

Thomas Bernier, Jacques-Andre landry [6] proposed a method for the representation and comparison of irregular two-dimensional shapes. This method used a polar transformation of the contour points about the geometric centre of the object. The distinctive vertices of

the shape are extracted and used as comparative parameters to minimize the difference of contour distance from the centre. In order to permit representational invariance to position, rotation and scale, the geometric centre of the shape is selected as a reference point. The distinctive vertices of the contour are described by polar coordinates. This coordinate substitution is done only to simplify the handling of scale and rotational variations. In order to provide for scale invariance, the maximum distance is computed and all distances are normalized to it. Thus all values fall between 0 and 1 regardless of the scale of the image or object. Similarly, in order to provide for rotational invariance, the phase of the polar representation is shifted by the angle associated with the maximum distance, such that the maximum is associated with an angle of zero. Shape similarity measure between two respective polar representations is calculated by the below formulae. The similarity of two shapes varying in scale and orientation is inversely proportional to the area lying between their respective polar representations. Shape similarity measure between two respective polar representations is calculated by the below formulae.

$$\sim \left(\frac{1}{2\pi} \int_{\theta=0}^{2\pi} \sqrt{(R_\theta - R'_\theta)^2}\, \partial\theta \right)^{-1}$$

Where θ is the angle of deflection from maximum radius (0 - *2n)* and R_θ the fraction of max radius at angle θ. Optimization of similarity is achieved by simply finding the relative orientation in which the area lying between the respective polar representations of the shapes is minimized.

This method was applied on natural images of leaves to demonstrate the accuracy and ability to distinguish between closely similar shapes. The categorization was 100% in all cases except one misclassification of a sumac leaf.

Ansari and Delp [7] proposed a landmark based approach for recognizing partially occluded objects. The landmarks of an object are points of interest relative to the object that have important shape attributes. Given a scene consisting of partially occluded objects, a model object in the scene is hypothesized by matching the landmarks of the model with those in the scene. A measure of similarity between two landmarks, one from the model and the other from the scene, is needed to perform this matching. In this correspondence they introduced a new local shape measure, sphericity. It is shown that any invariant function under a similarity transformation is a function of the sphericity. To match landmarks between the model and the scene, a table of compatibility, where each entry in the table is the sphericity value derived from the mapping of a set of three model landmarks to a set of three scene landmarks, is constructed. A technique, known as hopping dynamic programming, is described to guide the landmark matching through the compatibility table. The location of the model in the scene is estimated with a least squares fit among the matched landmarks. A heuristic measure is then computed to decide whether the model is in the scene.

The objects used in the library are wrench, needle-nose plier, wire cutter, speciality plier, wire stripper, Borneo, Hamahera, Luzon, Mindanao, New Guinea, Sulawesi, spacecraft. The test scenes are generated by overlapping few of the objects one over the other. The experimental results indicate that it can handle occlusion reasonably.

Lena Gorilick, Meirav Galun *et all* [11] proposed a novel approach that reliably computes many useful properties of a silhouette such as part structure, rough skeleton, local orientation, Aspect ratio of different parts, and convex & concave sections of the boundaries. They considered a silhouette surrounded by a simple, closed contour. Based on the notion of random walks, they computed a function that assigns, for every internal point in the silhouette, a value reflecting the mean time required for a random walk beginning at the point to hit the boundaries. This function is formalized as a partial differential equation, called the *Poisson equation,* with the silhouette contours providing boundary conditions. Then, they showed how we can use the solution to the Poisson equation to reliably extract various properties of a shape including its part structure and rough skeleton, local orientation and aspect ratio of different parts, and convex and concave sections of the boundaries. In addition to this they also discussed properties of the solution and show how to efficiently compute this solution using multigrid algorithms. They also demonstrated the utility of the extracted properties by using them for shape classification.

M.K. Hu [1] introduced moment invariants that are independent of position, size and orientation. Using the fundamental theorem of moment invariants, If the algebraic form of order p has an algebraic invariant, then the moments of order p have the same invariant but with the additional factor $|\ j\ |$. The method presented achieves orientation independence without ambiguity by using either absolute or relative orthogonal invariants to characterize each pattern for recognition.

A.Khotanzad and Y.H. Hong [26] introduced a new set of rotation invariant features. They are the magnitudes of a set of orthogonal complex moments of the image known as Zernike moments. Scale and translation invariance are obtained by first normalizing the image with respect to these parameters using its regular geometrical moments, A systematic reconstruction-based method for deciding the highest order of Zernike moments required in a classification problem is developed, The "quality" of the reconstructed image is examined through its comparison to the original one. More moments are included until the reconstructed image from them is close enough to the original picture. Tthe orthogonality property of the Zernike moments which simplifies the process of image reconstruction made the suggested feature selection approach practical. The method is tested using clean and noisy images from a 26-class character data set and a 4-class lake data set. The superiority of Zernike moment features over regular moments and moment invariants is experimentally verified.

CHAPTER THREE

OBJECT RECOGNITION USING MOMENT INVARIANTS

3.1 Introduction

Moment invariants have been frequently used as features for image processing, remote sensing, shape recognition and classification. Moments can provide characteristics of an object that uniquely represent its shape. Several techniques have been developed that derive invariant features from moments for object recognition and representation. These techniques are distinguished by their moment definition, such as the type of data exploited and the method for deriving invariant values from the image moments. It was Hu [1] that first set out the mathematical foundation for two-dimensional moment invariants and demonstrated their applications to shape recognition. They were first applied to aircraft shapes and were shown to be quick and reliable by [29]. These moment invariant values are invariant with respect to translation, scale and rotation of the shape. The moments which have the property of invariant image recognition as well as image reconstruction given the moment descriptors was introduced by [26]

Hu [1] defined seven of these shape descriptor values computed from central moments through order three that are independent to object translation, scale and orientation.

Translation invariance is achieved by computing moments that are normalized with respect to the centre of gravity so that the centre of mass of the distribution is at the origin (central moments). Size invariant moments are derived from algebraic invariants but these can be shown to be the result of simple size normalization. From the second and third order values of the normalized central moments a set of seven invariant moments can be computed which are independent of rotation.

3.2 Translation, rotation and scale invariant moments

Traditionally, moment invariants are computed based on the information provided by both the shape boundary and its interior region [1]. The moments used to construct the moment invariants are defined in the continuous but for practical implementation they are computed in the discrete form. Given a function *f(x,y)*, these regular moments are defined by:

$$M_{pq} = \int\int x^p y^q f(x,y)dxdy \qquad (1)$$

M_{pq} is the two-dimensional moment of the function *f(x,y)*. The order of the moment is (p + q) where p and q are both natural numbers. For implementation in digital from this becomes:

$$M_{pq} = \sum_X \sum_Y x^p y^q f(x,y) \qquad (2)$$

To normalize for translation in the image plane, the image centroids are used to define the central moments. The co-ordinates of the centre of gravity of the image are calculated using equation (3) and are given by:

$$\bar{x} = \frac{M_{10}}{M_{00}} \qquad \bar{y} = \frac{M_{01}}{M_{00}} \qquad (3)$$

The central moments can then be defined in their discrete representation as:

$$\mu_{pq} = \sum_{X}\sum_{Y}(x-\bar{x})^{p}(y-\bar{y})^{q} \qquad (4)$$

The moments are further normalized for the effects of change of scale using the following formula:

$$\eta_{pq} = \mu_{pq} \Big/ \mu_{00}^{\gamma} \qquad (5)$$

Where the normalization factor: $\gamma = (p + q / 2) + 1$.

Hu[1] recognized that rotation invariance is the most difficult to achieve and proposed two different methods for computing rotationally invariant moments. The first method

known as "*principal axis method*" uses the second order moments to compute the major and the minor axes of an ellipse that completely encloses the object. The second technique is based on a normalization procedure that achieves rotation invariance of *moments.* The algebraic moment invariants up to third order have been used for the recognition of different types of shapes and images.

The method used for rotation invariance through *principal axis* is based on the observation that moments may be computed relative to a unique set of *principal axes* of distribution and will therefore be invariant to the orientation of the distribution. The principal axis moments are obtained by rotating the axis of the central moments until μ_{11} is zero [3], [11-12]. The angle θ, measured from the original axis. The principal axis defined by:

$$\tan 2\theta = \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}$$

$$\text{So} \qquad \theta = \frac{1}{2}\tan^{-1}\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \qquad (6)$$

The angle obtained with Equation (6) is usually measured with respect to the major *principal axis*. In special case *Hu* defined seven values computed from central moments through order three, that are invariant to object scale, position, and orientation. In terms of the central moments, the seven moment invariants are given by

$\phi_1 = \eta_{20} + \eta_{02}$

$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta^2_{11}$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2$$

$$\phi_5 = (3\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{30})^2]$$

3.3 Algorithm

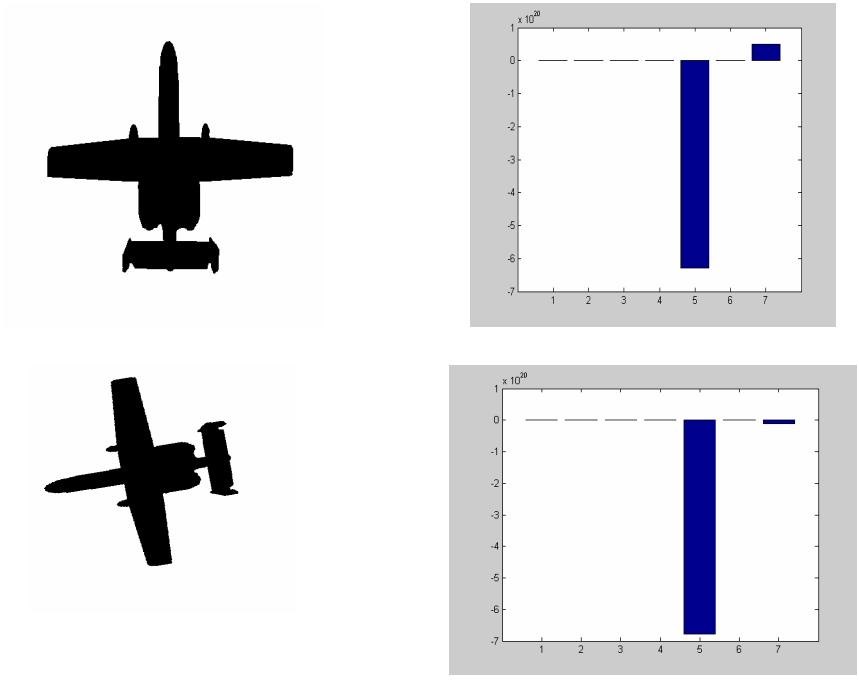
1. Clean up the image of noise by using a median filter and then removing all but the largest of the objects in the scene.
2. Find the edge of the image using the edge detector.
3. Find the moments of order three using the Equation 2
4. Find the centre of gravity of the image

5. Find the central moments, which eliminates the translation effect.
6. Normalize the moments by dividing them with the m_{00} (size of the image) for removing the scaling effect.
7. Find the rotational invariant moments ϕ_1 - ϕ_7
8. For a given test image, its moment invariants are compared with all the model objects moment invariants using the Euclidean distance between them.
9. The model object with least Euclidean distance is a recognized object.

3.4 Results and Analysis

The recognition system is tested by generating the test objects by translating, rotating, and scaling and adding noise to the model objects contained in a database of size 60. The test objects were randomly rotated and translated, but scaled to factor of around $\frac{3}{4}th$, $1\frac{1}{4}th$ and some without scale of their model sizes. The database objects used can be seen in Appendix A. About 100 test objects were used for each of the experiments for testing similarity transformation and noisy objects with similarity transformations. The salt & pepper noise of density 10% is added to the objects for generating the noisy test objects. Median filter was used in the experiment to filter the noise, so that the noise remains on the boundary of the object. Median filtering is a type of neighborhood processing that is particularly useful for removing 'salt and pepper' noise from an image [33]. The median filter considers each pixel in the image and it looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the

pixel value with the *mean* of neighboring pixel values, it replaces it with the *median* of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value [34]. The percentage of recognition recorded in case of just similarity transformations are 65%. In case of similarity transformations with noise, it is about 61%. Effect of noise on recognition rate is depicted in Table 3. The moment invariants for an object in case of similarity transformations, noise and occlusion are shown Figure 2. We can see that the moment invariants did not change much in case of similarity transformations and noise. In the last figure the occlusion caused the change in the values of invariants.



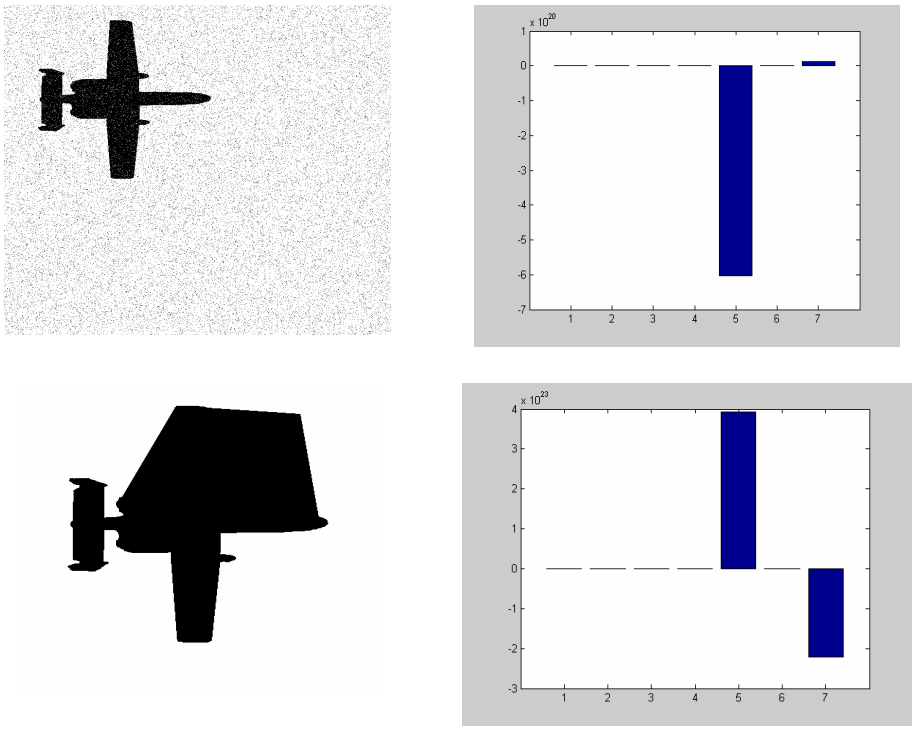


Figure 2: Moment invariants under different transformations

Table 3: Effect of noise on recognition (Moments Method)

Percent Noise	Recognition rate
10%	59%
20%	48%
30%	41%

3.5 Scope and Limitations

- The moment invariants of an object are robust to similarity transformations
- The moment invariants are little sensitive to noise
- The moment invariants are sensitive to occlusion

CHAPTER FOUR

OBJECT RECOGNITION USING FOURIER DESCRIPTORS

4.1 Introduction

To characterize objects we use features that remain invariant to translation, rotation and small modification of the object's aspect. The invariant Fourier descriptors of the boundary of the object can be used to identify an input shape, independent on the position or size of the shape in the image.

Fourier transform theory [37] has played a major role in image processing for many years. It is a commonly used tool in all types of signal processing and is defined both for one and two-dimensional functions. In the scope of this research, the Fourier transform technique is used for shape description in the form of Fourier descriptors. The Fourier descriptor is a widely used all-purpose shape description and recognition technique [38]. The shape descriptors generated from the Fourier coefficients numerically describe shapes and are normalized to make them independent of translation, scale and rotation. These Fourier descriptor values produced by the Fourier transformation of a given image represent the shape of the object in the frequency domain. The lower frequency descriptors store the general information of the shape and the higher frequency the smaller details. Therefore,

the lower frequency components of the Fourier descriptors define a rough shape of the original object

The Fourier transform theory can be applied in different ways for shape description. One method works on the change in orientation angle as the shape outline is traversed [4]. But in our research the following procedure was implemented, the boundary of the image is treated as lying in the complex plane. So the row and column co-ordinates of each point on the boundary can be expressed as a complex number,

The advantage of using the Fourier transform is in its invariant properties. Rotating the object merely causes a phase change to occur, and the same phase change is caused to all the components. In the Fourier spectrum the magnitude given by: $\sqrt{F_s^2(u) + F_c^2(u)}$ and the phase by $\tan^{-1}(\frac{F_s^2(u)}{F_c^2(u)})$. The magnitude is independent of the phase, and so unaffected by rotation. (This is an example of the very important property of shift invariance).

The simple geometric transformations of the Fourier transform

- ➔ Translation: $u(n)+t \rightarrow a(k)+t\delta(k)$
- ➔ Rotation : $u(n)ej\theta \rightarrow a(k)ej\theta$
- ➔ Scaling: $su(n) \rightarrow sa(k)$
- ➔ Starting point: $u(n-t) \rightarrow a(k)\ ej2\prod tk/N$

4.2 Implementation Strategy

The boundary of the image is treated as lying in the complex plane. So the row and column co-ordinates of each point on the boundary can be expressed as a complex

number, x + jy where j is sqrt (-1). Tracing once around the boundary in the counter-clockwise direction at a constant speed yields a sequence of complex numbers, that is, a one-dimensional function over time. In order to represent traversal at a constant speed it is necessary to interpolate equi-distant points around the boundary. Traversing the boundary more than once results in a periodic function. The Fourier transform of a continuous function of a variable *x* is given by the equation:

$$F(u) = \int_{-\infty}^{\infty} f(u) e^{-j2\pi ux} dx \qquad (1)$$

When dealing with discrete images the Discrete Fourier Transform (DFT) is used. So equation (1) transforms to:

$$F(u) = \left(\frac{1}{N}\right) \sum_{x=0}^{N-1} f(u) e^{\frac{-j2\pi x}{N}} \qquad (2)$$

The variable x is complex, so by using the expansion *e[-j A]* = cos *(A)* – *j.* sin *(A)* where *N* is the number of equally spaced samples, equation (2) becomes:

$$F(u) = \left(\frac{1}{N}\right) \sum_{x=0}^{N-1} f(x + jy)).(\cos(Ax) - j.\sin(Ax)) \qquad (3)$$

Where $A = 2\pi u/x$.

The DFT of the sequence of complex numbers, obtained by the traversal of the object contour, gives the Fourier descriptor values of that shape.

The Fourier descriptor values can be normalized to make them independent of translation, scale and rotation of the original shape. Simply, translation of the shape by a complex quantity having *x* and *y* components, corresponds to adding a constant $x + jy$ to each point representing the boundary. Scaling a shape is achieved by multiplying all co-ordinate values by a constant factor. The DFT results in all members of the corresponding Fourier series being multiplied by the same factor. So by dividing each coefficient by the same member, normalization for size is achieved. Rotation normalization is achieved by taking only the magnitude. Since rotating the object merely causes a phase change to occur, and the same phase change is caused to all the components. In the Fourier spectrum the magnitude given by: $\sqrt{F_s^2(u) + F_c^2(u)}$ and the phase by $\tan^{-1}(\frac{F_s^2(u)}{F_c^2(u)})$. The magnitude is independent of the phase, and so unaffected by rotation.

To apply the Fourier descriptor technique to the data set extracted as the boundary of the object, the points are stored as a series of complex numbers and then processed using the Fourier transform resulting in another complex series also of length N. If the formula for the discrete Fourier transform were directly applied each term would require N iterations to sum. As there are N terms to be calculated, the computation time would be proportional to N^2. So the algorithm chosen to compute the Fourier descriptors was the Fast Fourier Transform (FFT) for which the computation time is proportional to NlogN.

Given two sets of Fourier descriptors, how do we measure their degree of similarity? An appropriate classification is necessary if unknown shapes are to be compared to a library of known shapes. If two shapes, A and B, produce a set of values represented by a(i) and b(i) then the distance between them can be given as c(i) = a(i) – b(i). If a(i) and b(i) are

identical then c(i) will be zero. If they are different then the magnitudes the coefficients in c(i) will give a reasonable measure of the difference. It proves more convenient to have one value to represent this rather than the set of values that make up c(i). The easiest way is to treat c(i) as a vector in a multi-dimensional space, in which case its length, which represents the distance between the planes, is given by the square root of the sum of the squares of the elements of c(i). We implemented a simple classifier that calculates the Euclidean distances of the corresponding Fourier descriptors of the input shape and each of the shapes contained in the database as shown in Figure 3.

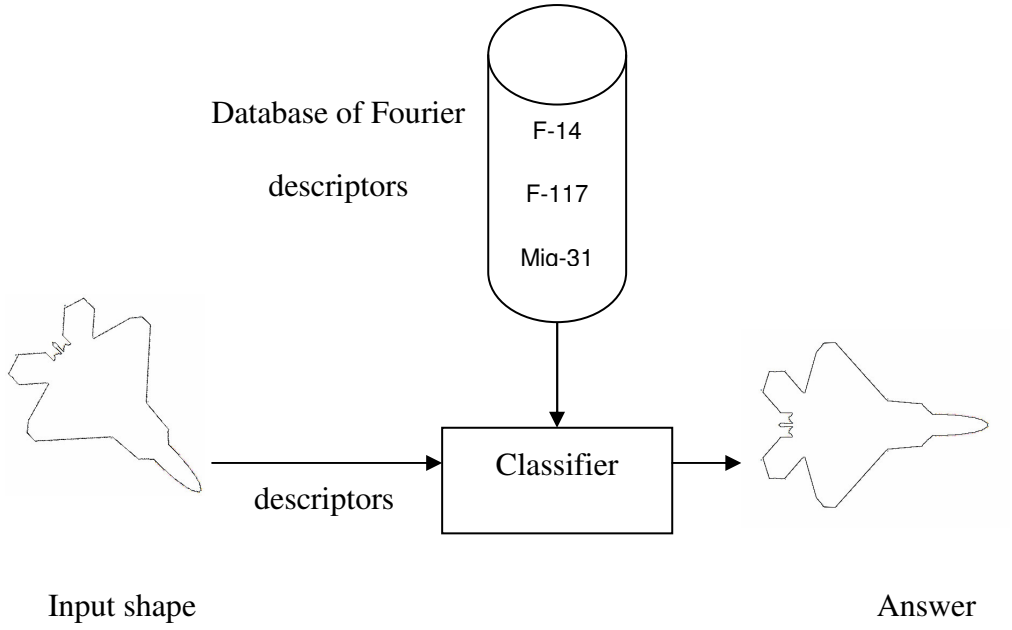


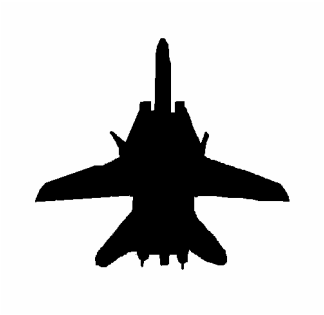
Figure 3: Pictorial Description of method

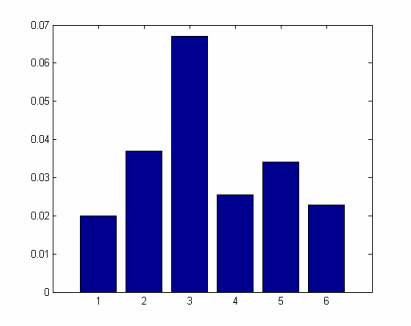
4.3 Algorithm

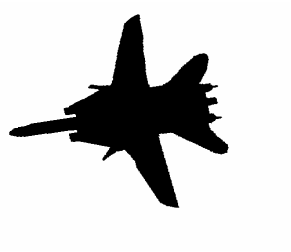
1. Clean up the image of noise by using a median filter and then removing all but the largest of the objects in the scene.
2. Find the boundary of the image.
3. Convert the *x*, *y* coordinates in the contour to a one-dimensional vector by treating them as a complex pair. That is: $U(n) = X(n) + i * Y(n)$.
4. Perform the Fast Fourier Transform on *U* and take the absolute value to create a new vector *A* which is the magnitude of the coefficients.
5. Throw away $A(0)$ since it is the DC component; that is, it represents only the *translation* of the contour.
6. Truncate $A(>6)$ since higher frequency components don't add much to the shape and are wildly affected by noise.
7. Normalize the remaining magnitudes by dividing each element of *A* by $A(0)$. Reason: when a shape is scaled by a constant factor (*alpha*), the magnitude of each of the coefficients in the resulting FFT is also multiplied by *alpha*. To remove *alpha* from the equation we simply divide by a number, $A(0)$, which is known to be a product of *alpha*.
8. The result in *A* is the Fourier Descriptor.
9. For a given test silhouette, Its FD's are compared with all the model objects FD's using the Euclidean distance between them.
10. The model object with least Euclidean distance is a recognized object.

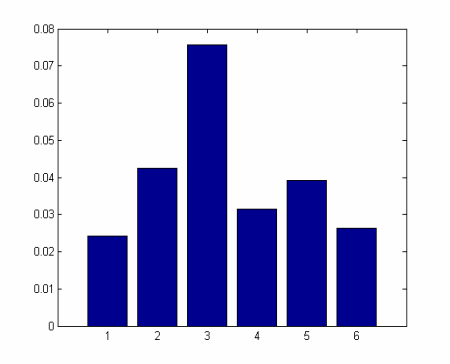
4.4 Results and Analysis

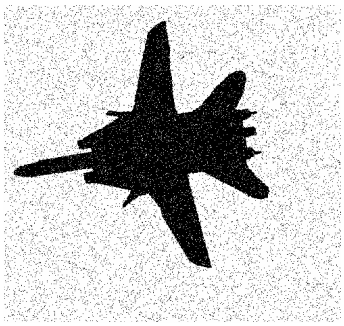
The recognition system is tested by generating the test objects by translating, rotating, and scaling and adding noise to the model objects contained in a database of size 60. About 100 test objects were randomly generated for each of the experiments in testing similarity transformation and noisy objects with similarity transformations. The test objects were randomly rotated and translated, but scaled to factor of around $\frac{3}{4}th$, $1\frac{1}{4}th$ and some without scale of their model sizes. For generating the noisy images, the same procedure used in section 3.4 of the previous chapter was used. The percentage of recognition recorded in case of just similarity transformations are 85%. In case of similarity transformations with noise, it is about 83%. The less effect of noise on recognition is due to no change in low frequency Fourier descriptors, while affecting the high frequency descriptors. In Figure 4, we can see that the Fourier descriptors did not change much in case of similarity transformations and noise. The effect of noise on recognition rate is shown in Table 5. In the last figure the occlusion caused the change in the values of descriptors. Table 4 depicts the effect on the values of the Fourier descriptors in case of different transformations.

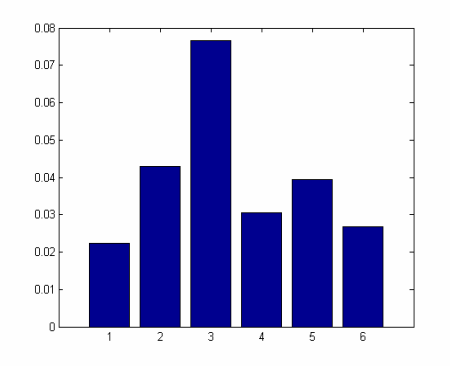



0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
1
2
3
4
5
6




0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
1
2
3
4
5
6




0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
1
2
3
4
5
6

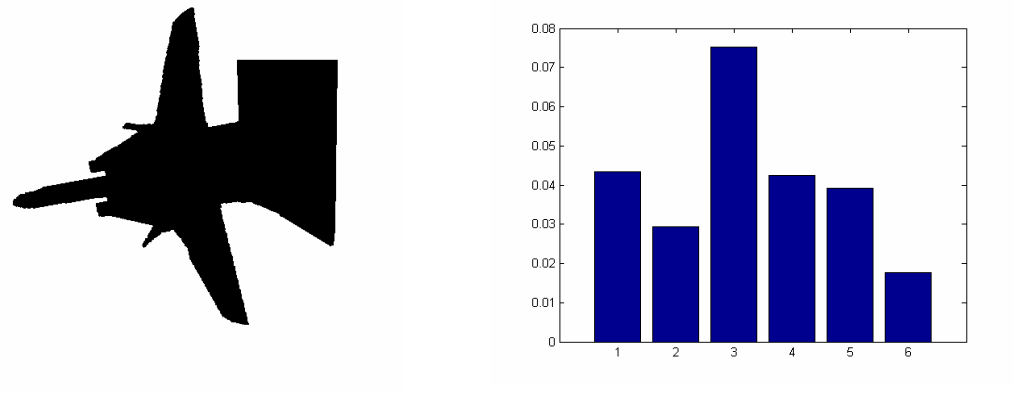


Figure 4: Fourier descriptors of an object under different transformations

Table 4: Similarity measures after the alternations

Changes from the original	**Similarity Measure**
20 degree rotation	98
0.75 scaling	99
Random translation	100.0
Combination of all three	96
Noise of 10% density	98

Table 5: Effect of noise on recognition (Fourier Descriptors Method)

Percent Noise	Recognition rate
10%	83%
20%	81%
30%	80%
40%	78%
50%	74%

4.5 Scope and Limitations

- The Fourier descriptors of the boundary are robust to similarity transformations
- The Fourier descriptors are little sensitive to noise, since addition of noise on boundary effects only the high frequency Fourier descriptors keeping low frequency Fourier descriptors unchanged.
- The Fourier descriptors are sensitive to occlusion as it changes the complete wave form

CHAPTER FIVE

OBJECT RECOGNITION USING SHAPE SPACE APPROACH

The recognition problem of partially occluded objects is mainly addressed as a difficult problem. A recognition system is designed and used for the recognition of two-dimensional occluded objects using contour feature.

5.1 Representation of 2d Shape

The representation of the object is first thing in the feature extraction phase. Ansari and Delp [7] represented 2D objects using the landmarks that are grouped into consecutive triples called local triangles. Hoffman and Richards [40] suggest that high curvature points on the objects boundary contour are perceptually salient and play a significant role in the recognition of the object

Suppose a 2D object is represented as a set of points on the plane, called landmarks. For example, they may be perceptual salient points on the object boundary, such as high-curvature and extreme points. Let the landmarks be represented by the vector,

$$x = [\, x_1 , x_2 , \ldots , x_n \,] \qquad (1)$$

Where n is the total number of landmarks and x_i is the position of i^{th} landmark, represented as a complex number. Then x is a point in C^n , the n-dimensional complex space.

According to Kendall [27], the shape of x is "What is left when the effects associated with translation, scaling and rotation are filtered away". To remove the effect of translation, we let

$$x' = [x_1', x_2' \ldots\ldots x_n'] = [x_1 - x', x_2 - x', \ldots\ldots, x_n - x'] \qquad (2)$$

where

$$\bar{x} = 1/n \sum_{i=1}^{n} x_i \qquad (3)$$

is the centroid of $x_1, x_2, \ldots x_n$.

Now x' satisfies

$$\sum_{i=1}^{n} x_i' = 0 \qquad (4)$$

Hence, x' is a point on a *n-1* dimensional complex hyperplane passing through the origin of C^n. Similarly to remove the effect the rotation and scaling, we associate x' with an equivalence class

$$x' = \{ \lambda x', \lambda \in C \} \qquad (5)$$

Where C is the set of complex numbers. As λ varies over C, x' covers all possible scaling and rotations of x'. Now x' is the shape of x.

x' represents the complex line passing through origin and on the *n-1* dimensional complex hyperplane defined by Eq.(4)

5.2 Landmarks Extraction

High curvature points on an object's boundary contour play a significant role in the recognition of the object. But the problem with high curvature points is that noise on the boundary often generates very high curvature. So, we require the extreme points which are uniformly distributed over the boundary contour. To Extract the extreme points as landmarks, we represented the objects boundary contour using a "centroidal distance representation $C(s)''$. $C'(S) = 0$ will give the maxima and minima of *C(s)* which we do not require. What we need is to find a set of extrema of *C(s)* that are relatively far away from each other. To achieve such points Legendre polynomials provide good solution to this problem.

Legendre polynomials [29] form an orthonormal basis of L^2[-1,+1] and they can be recursively generated as follows:

$P_{n+1}= (2n+1/n+1)*x\ P_n(x)-(n/n+1)*P_{n-1}(x)$

where $P_0(x)=1$ and $P_1(x)=x$

Legendre polynomial have the desirable property that any function *f(x)* in L^2[-1,+1], its *N+1* the order approximation is given by

$$f_{n+1}(x) = \sum_{n-0}^{N+1} f_n p_n(x)$$

f_n is the inner product of *f(x)* and $p_n(x)$.

The above approximated function has *N* maxima and minima points that are relatively far away from each other.

Before applying Legendre polynomials in extracting landmarks we first re-parameterize $C(s)$ to $C(p)$ where $p \in [-1,1]$.

Finally, $C'(p) = 0$ is used to find N landmarks.

5.3 Classification of un-occluded objects

We have used the Kendall's formulae of finding the geodesic distance between two sets of points given there sequence. Ansari and Delp [7] used the sum of the invariant differences between the local triangles to find he difference between two objects. According to Kendall [27], the distance between the two sets of points is given by $D(x^*, y^*) = \arccos(|< x', y' >|/\| x' \|.\| y' \|)$, where x' and y' are the centered objects and $<.,.>$, $|.|$, and $\|.\|$ are, respectively, the inner product of the two complex vectors, the norm of a complex number and the norm of the complex vector.

Let $x=[x_1,x_2,\ldots,x_n]$ are the candidate points of model object and $y=[y_1,y_2,\ldots,y_n]$ is are the candidate points of the test object.

$$k_0 = \min_{0 \le m \le 9, 1 \le k \le K} d((C^m y)^*, x_k^*)$$

Where $\boldsymbol{x_k}$ is the k^{th} model object.

During the recognition, we need to consider all the circular shifts (C^m) when comparing y with model x.

The distance between the test and the model objects in the database is calculated by the above formulae which gives the model which is the nearest in distance to the test object.

5.4 Classification of Occluded objects

Occlusion is the severe shape distortion. In case of occlusion, some of the candidate points are false and some are additional. So, we have to take care of those candidate points which are false.

So, we have used less number of candidate points than the actual number of candidate points *(k<n)*. We assumed that the occlusion occurred on contiguous subscripts. So, using the contiguous subscript constraint we have *n* ways of taking k candidate points from test and model object.

Let $x=[x_1,x_2,\ldots,x_n]$ are the landmarks of model object and $y=[y_1,y_2,\ldots,y_n]$ is are the landmarks of the test object of which $y=[y_1,y_2,\ldots,y_q]$ are correct landmarks. Where $q<n$. we can take q landmarks from each from x and y and compute the procrustean distance. But, this creates the high combinatorial complexity. So, we assume that occlusion always cuts off a continuous curve from the objects contour. So, this constraint drastically reduces the number of searches.

Now, for each $q <= n$, we define a partial distance between an observed object and model object. We define the partial distance as

$$d_q(x,y) = \min_{1\le i,j\le n} d((x_q^i)^*,(y_q^j)^*)$$

x_q^i or y_q^i denote the i^{th} subset of q contiguous landmarks. Minimization is taken over all possible values of i , j.

The above formulae is applied to all the objects of the database using the below formulae.

$$d_o(x, y) = \min_{1 \le q \le k} d_q(x, y)$$

Where k is the total number of objects of the database.

5.5 Experimental Description and intermediate results

The Object's boundary is represented in terms of centroidal distance representation as shown in Figure 5. It is the distance from center of the object to the all the points on the boundary.

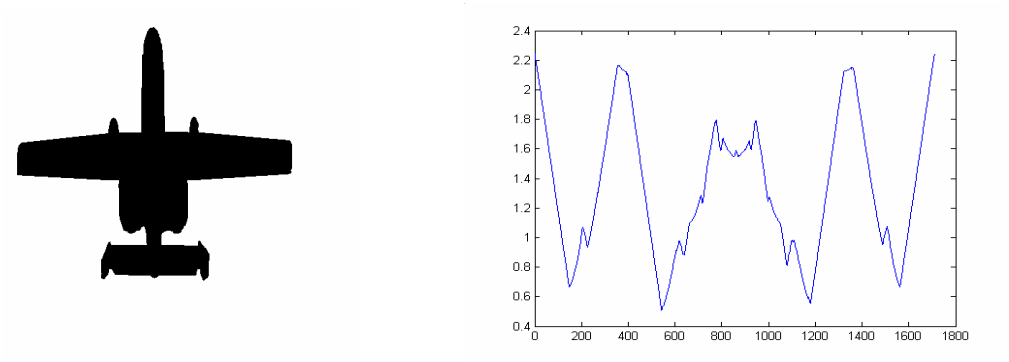


Figure 5: An object and its centroidal distance representation

The centroidal distance representation in Figure 5 is approximated by applying the polyfit function of matlab as shown in Figure 6.

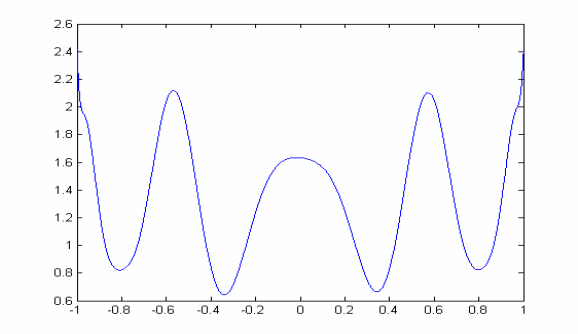


Figure 6: Polynomial approximation of centroidal distance representation

The fitted polynomial is now approximated to a known degree by using the Legendre's polynomials generated as shown in Figure 7.

$$f_{n+1}(x) = \sum_{n-0}^{N+1} f_n p_n(x)$$

f_n is the inner product of $f(x)$ and $p_n(x)$. After approximating, we find the maxima and minima that are existing in it by finding the roots of $f_{n+1}(x)$.

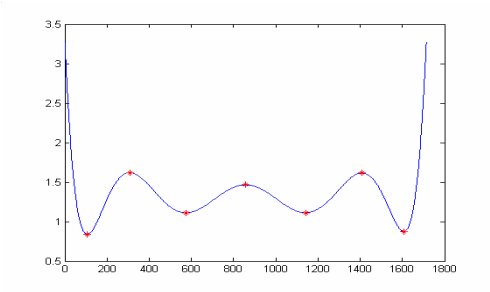


Figure 7: Approximation using Legendre's polynomial of degree seven

The landmarks of Figure 7, marked on the object boundary are shown in Figure 8.

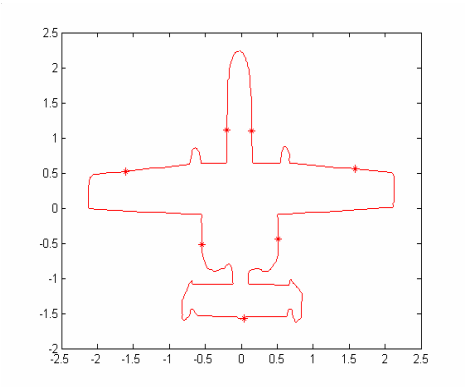


Figure 8: Representation of landmarks on object's boundary

We used salt and pepper function of matlab to generate noisy airplanes, an example of an object with salt and pepper noise added is shown in Figure 9 and in Figure 10 the salt and pepper noise is removed using median filter. We used median filter to remove the noise outside of the object, so that noise is present only on the border as shown in Figure 11. Figure 12 shows the landmarks extracted after approximation with Legender's polynomials.

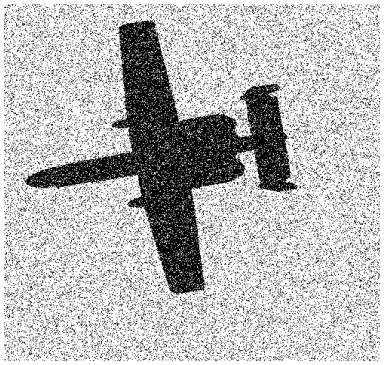


Figure 9: Object with noise

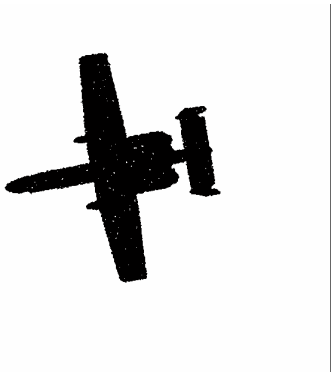


Figure 10: Object after applying median filter

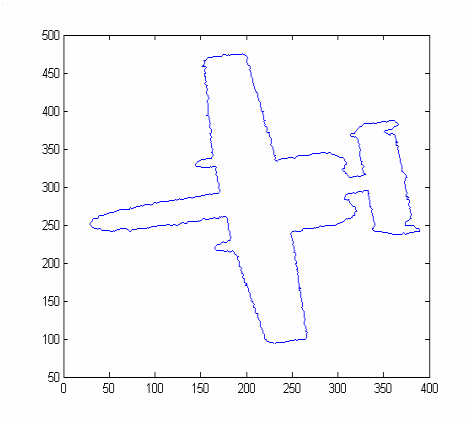


Figure 11: Noisy Boundary

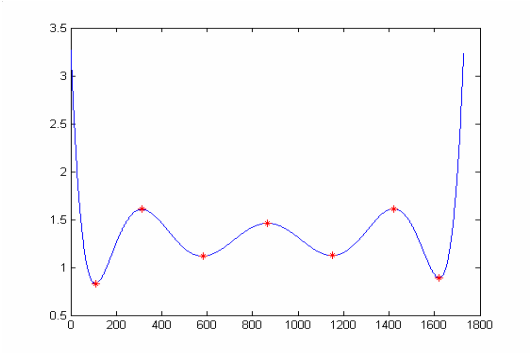


Figure 12: Landmarks after approximation

The landmarks of Figure 12 when marked on the object boundary looks like in Figure 13.

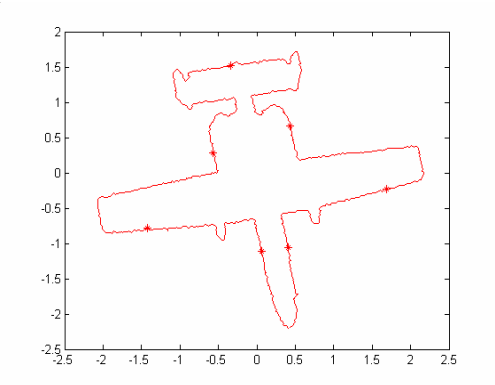


Figure 13: Representation of landmarks on objects boundary

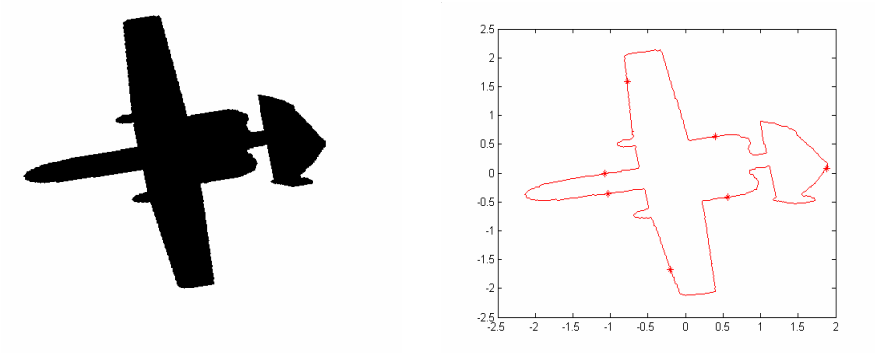


Figure 14: Representation of landmarks on occluded objects, with little change in center of gravity.

5.6 Time Complexity

Suppose each model object has *n* landmarks. Then to compute the shape space distance between input object and model object, we need *n* distance calculations. Suppose there are K object models. Then, total number of distance computations required is K*n*.

In case of occlusion, suppose the true landmarks considered are m<n. To compute the distance between test and modal object we need to perform n^2 computations .Therefore when there are K object models, the total number of distance computations required is Kn^2. The computational complexity in this approach in case of non-occlusion and occlusion seems to be manageable, but in case of large size of object database and if the number of landmarks chosen are very large then this method suffers and some additional strategy such as hierarchical classification will help to achieve efficiency.

5.7 Comparison with Other Experiments

The experiment is carried out for twenty model objects to demonstrate its efficiency of recognition as it was taking pretty long time for recognition. The test objects were

generated by translating, scaling and rotating the model objects. The test objects were randomly rotated and translated, but scaled to factor of around $\frac{3}{4}th$, $1\frac{1}{4}th$ and some without scale of their model sizes. In case of testing occlusion, the model objects translated, scaled, rotated and occluded. The occluded objects are generated by occluding the model objects with little or no change in center of gravity. In case of testing the noisy objects, the noise is added using the imnoise function of matlab with a given percentage density. The remaining procedure of adopted for creating noise on the boundary of the object is same as in the previous experiments. The intermediate results such as test object under different transformations with landmarks are already shown in section 5.5. Twenty airplane objects were used in the database for this experiment and the same number is used for the previous two for comparison.

Table 6: Comparison of results for 20 model objects

Tests / methods	Similarity transformations (50 test objects)	Noise+Similarity transformations (50 test objects)	Occlusion (50 test objects)	Noise+ occlusion (50 test objects)
Using Moment invariants	50	50	0	0
Using Fourier transforms	50	42	2	0
Using shape space approach	47	46	41	41

The comparison of experimental results using moment descriptors, Fourier descriptors and Shape Space approach is depicted in Table 6. Figure 15 shows that the recognition rate is lower when compared to the classical methods of using moment and Fourier descriptors is lower in case of similarity transformations. The tested objects are generated by translating, rotating, and scaling the model objects

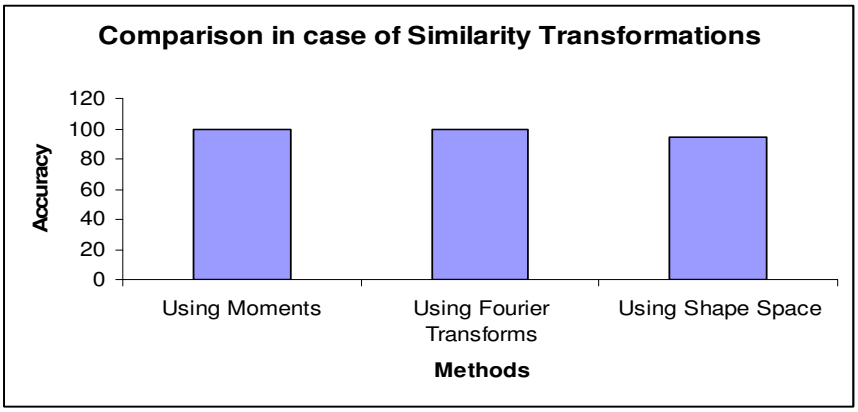


Figure 15: Effect of similarity transformations on accuracy

Figure 16 shows that the performance is slightly lowered in case of noise added to the boundary. But, the performance of using moment descriptors performed reasonable well.

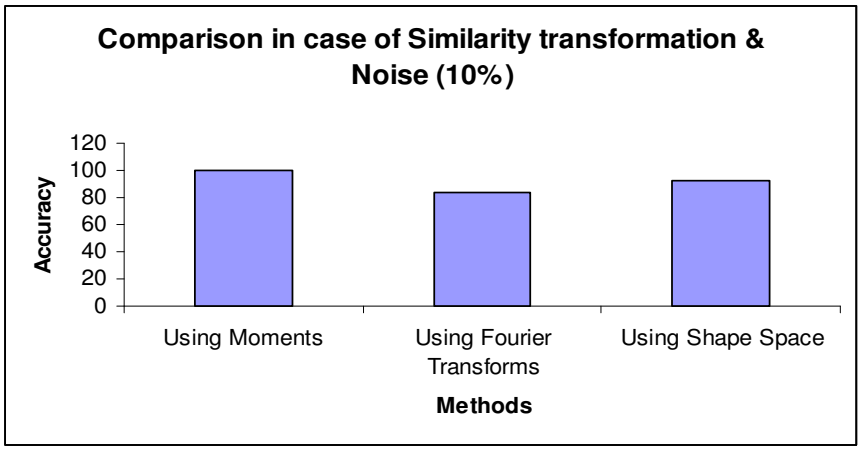


Figure 16: Effect of similarity transformations and noise on accuracy

In Figure 17 the effect of similarity transformations with occlusion on recognition rate in case of all three experiments is depicted. The classical methods performed almost nil, while the shape space approach performed fairly well. Again the point to mention here is the occlusions were not random, but they are occluded with very little change in center of gravity.

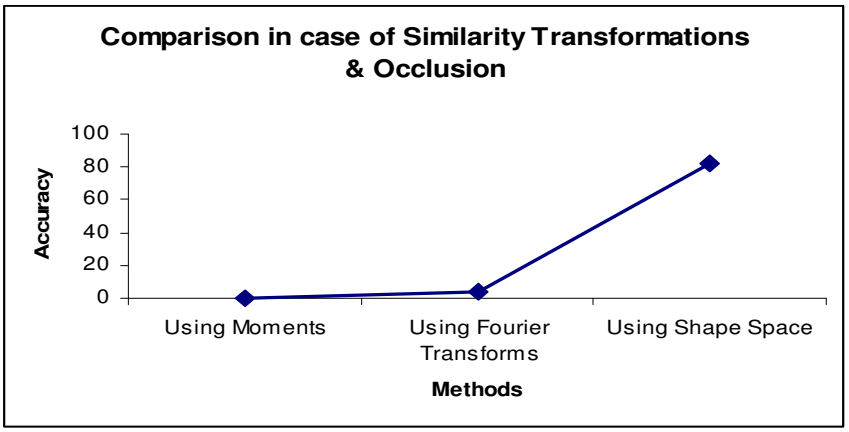


Figure 17: Effect of similarity transformations and occlusion on accuracy

Figure 18 depicts the effect of similarity transformation, noise and occlusion on the recognition rate. The graph looked almost same as in Figure 17. The conditions of generating the test objects are same as the previous experiment.

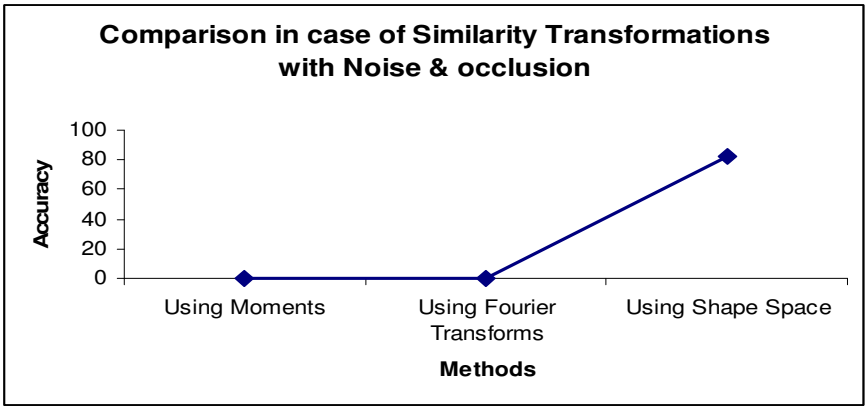


Figure 18: Effect of similarity transformations, noise and occlusion on accuracy

CHAPTER SIX

OBJECT RECOGNITION USING INDEXING APPROACH

6.1 Introduction

This proposed approach addresses the problem of similarity transformations as well as occlusion. The recognition problem of partially occluded objects is solved using the local feature descriptors for each contour segment and an indexing strategy is used to classify the object.

6.2 Introduction to Indexing approach

The partial shape recognition techniques [12] & [13] utilized the local features described by Fourier descriptors but used the sequential approach in classification. [15] tried to match the control points in the unknown shape to those of a shape from the template data bank, and estimates the translation, rotation, and scaling factors to be used to normalize the boundary of the unknown shape. The local features and neural networks for partial shape recognition is used by [16]. The problem of shape retrieval by shape similarity, using local features and metric indexing is addressed by [36].

If the number of objects is very large and the problem cannot be solved using feature space partitioning, then indexing techniques become attractive. The matching approach

discussed in the previous section is a sequential approach and requires that the unknown object be compared with all objects. This sequential nature of the approach makes it unsuitable with a number of objects. In such a case, one should be able to use a hypothesizer that reduces the search space significantly. Joong-Hwan Baek, Keith A. Teague [18] used local features such as corners, arcs, parallel-lines, and corner-arcs which were extracted from the preprocessed image and the hashing method was used in order to match the hypothesized objects. The local shape descriptors were used by [19] to carry out an efficient indexed search over the models so as to reduce the search space. An efficient coarse-to-fine recognition is proposed by [20] in which hypotheses are only generated for a subset of contours with enough discriminative information.

The higher-dimensional spaces in indexing are used by [17]. Their analysis indicates a dramatic reduction in recognition time by increasing the size of the feature vectors. Feature indexing approaches use features of objects to structure the model base. [14] proposed hierarchical structure (whole object to component sub-parts) and used these hierarchies to achieve robust recognition based on the optimization that allow him to use an indexing scheme for his model library. When a feature from the indexing set is detected in an image, this feature is used to reduce the search space. More than one feature from the indexing set may be detected and used to reduce the search space and in turn reduce the total time spent on object recognition.

Once the candidate object set has been formed, the next step is to compare the models of each object in the reduced set with the image to recognize the object. The verification phase should be used for selecting the best object candidate.

Typically, the following are the phases in this research approach:

1. Corner Detection
2. local Feature extraction
3. Hypothesis generation
4. Verification

6.3 Corner Detection

The algorithm we used is a modified form of [2] and used by [28]. Corner point is defined as a point where triangle of specified angle can be inscribed within specified distance from its neighbor points. The number of neighbor points to be checked is predefined.

First Pass: For each point pi it is checked if triangle of specified size and angle is inscribed or not. Following three conditions are used

$$d_{\min}^2 \le | p - p_k^+ |^2 \le d_{\max}^2$$

$$d_{\min}^2 \le | p - p_k^- |^2 \le d_{\max}^2$$

$$\alpha \le \alpha_{\max}$$

Where

p is point under consideration for corner point.

p_k^+ is the k^{th} clockwise neighbor of p.

p_k^- is the k^{th} anti-clockwise neighbor of p.

Taking

$$a = | p - p_k^+ |$$

$$b = | p - p_k^- |$$

$$c = | p_k^+ - p_k^- |$$

The angle alpha can be computed by using cosine law

$$a^2 + b^2 - c^2 - 2ab\cos\alpha = 0$$

$$\alpha = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

All the three conditions described in equations are necessary for first pass. Now each point *p* may have zero, one or more than one alpha values. Among all alpha values, minimum value is taken as the alpha value of that point *p*.

Second Pass: The corner algorithm can detect adjacent points as corners. Because we specify threshold value of alpha and corner points as well as points neighbor to corner points can have value less than alpha. Second pass removes such points. If the difference of contour indices of corner points cp_i and cp_{i+1} is less than IndexLimit then the corner point whose alpha is high is removed from the candidate corner list. It means that if corner point cp_i has contour index *i* and corner point cp_{i+1} has contour index *j* then (*j-i* >= IndexLimit). The procedure of detecting corner points is given in the Figure 19.

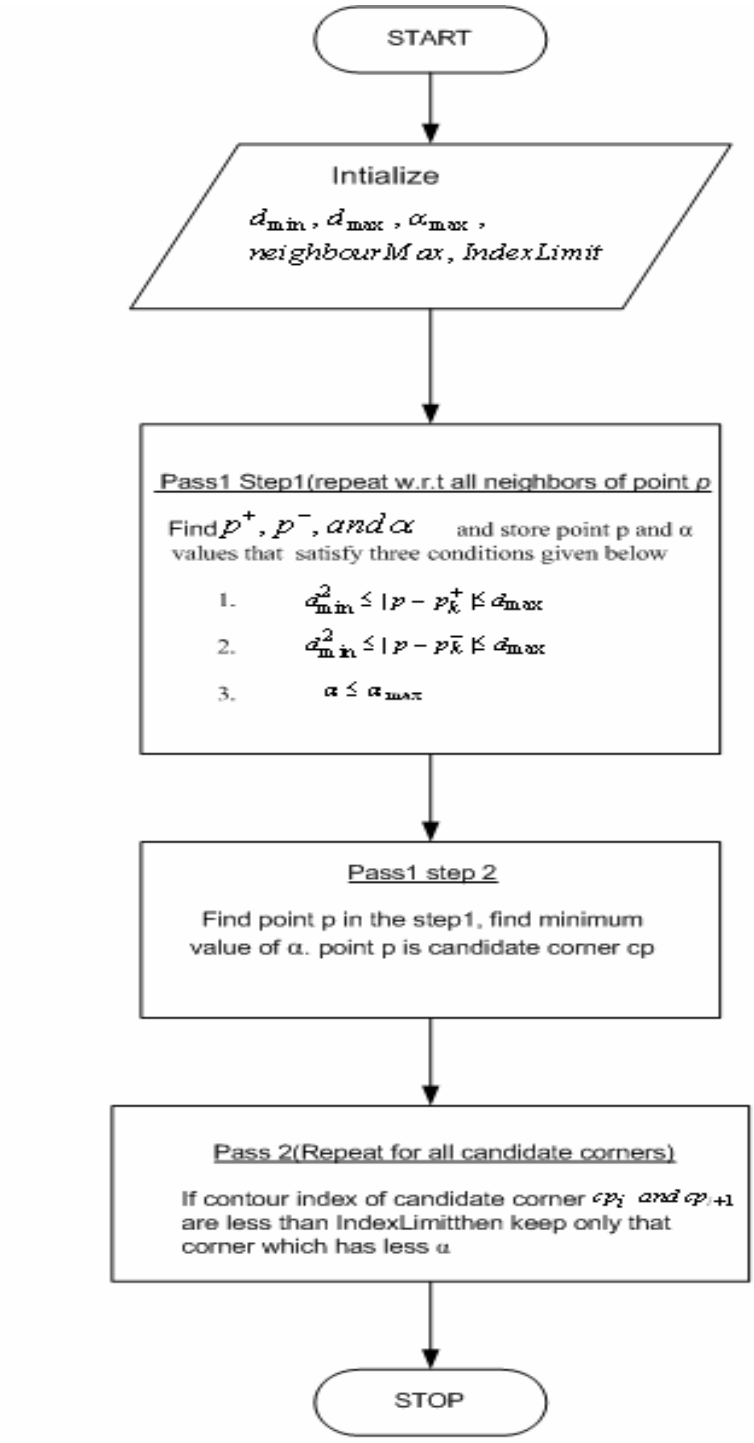


Figure 19: Flow Chart of Corner Detection Algorithm

The part of the contour between every two corner points is taken as one segment. Parameters used for corner detection algorithm proposed by [28] in our experiment are as follows

$d_{\min} = 5,\ d_{\max} = 8,\ \alpha_{\max} = 120$, $IndexLimit = 10$

Few results after applying corner detection algorithm

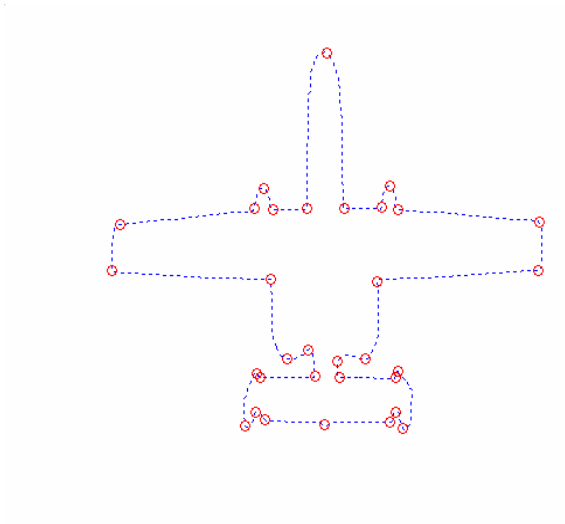


Figure 20: Object one with corner points

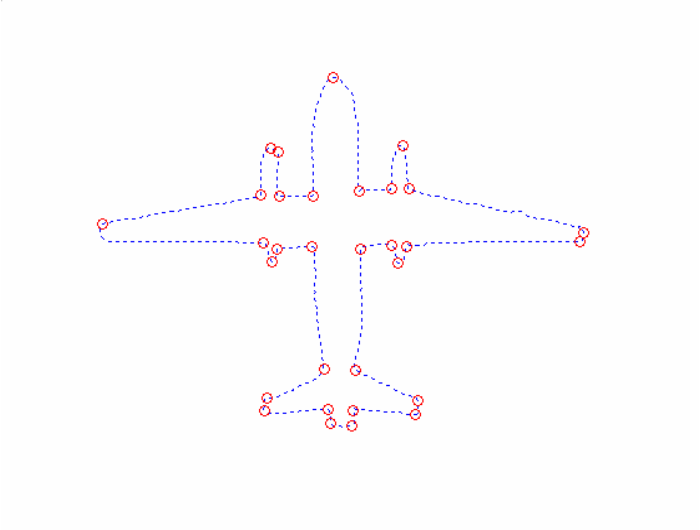


Figure 21: Object two with corner points

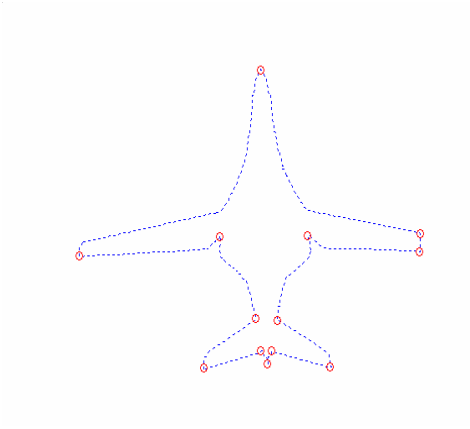


Figure 22: Object three with corner points

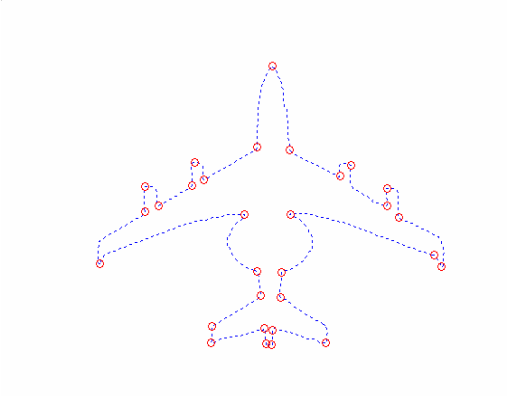


Figure 23: Object four with corner points

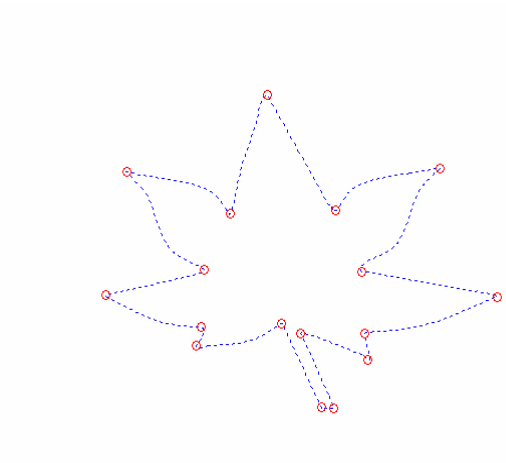


Figure 24: Object five with corner points

6.4 Feature Extraction

A set of descriptors for each segment that are invariant with respect to scale transformations are extracted from the geometric features of the segments as shown in Figure 25.

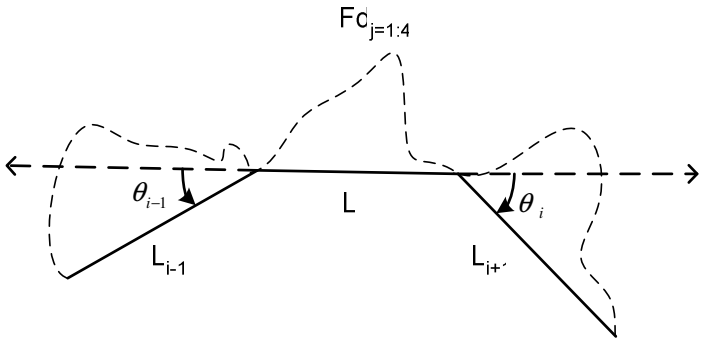


Figure 25: Segment Descriptors

Feature set of segment *i* is $\left(\frac{L_i}{L_{i-1}}, \frac{L_i}{L_{i+1}}, \theta_{i-1}, \theta_i, FD_{j=1:4} \right)$

A feature descriptor set has eight descriptors which describe the segments shape as well as segments length and exterior angle with respect to its adjacent segments.

6.5 Building the model library

Given the descriptors of all the segments for all the objects in the database, we define the set of index tables $\{Inx_k\}$ each associated to the distinct descriptor D_k. Typically, index table entry contains descriptor followed by its object number. For descriptor set of eight, we will have eight index tables.

6.6 Hypothesizer

Fragments of contours of the test object are matched by using local shape descriptors as shown in Figure 26. Let $D(O_i)$ *and* $D(O_j)$ be the set of descriptors of objects O_i and O_j respectively .The contour segment of object O_i is said to be matched contour segment of object O_j, if for every descriptor D_k of there corresponding segments of objects satisfies $\left|D_k(O_i) - D_k(O_j)\right| \le \Delta D_k$, where ΔD_k an accepted tolerance for D_k. Due to digitization noise, some tolerance must be established on the value of descriptors to ensure reliable indexing, and at the same time the bound should be tight enough to preserve the descriptor discriminating information by allowing some degree of sharing among similar shapes [19]. The effect of digitization of noise on corner points is shown in Table 7.

Given the descriptor set of test object for a particular segment, a heuristic search function H consists of finding the cluster of objects whose segments share the values of test object descriptor set. The objects whose segments got matched will be voted. Similarly the model objects are voted for all the segments of the test object. The few model objects which received the highest votes will be taken as hypothesized objects.

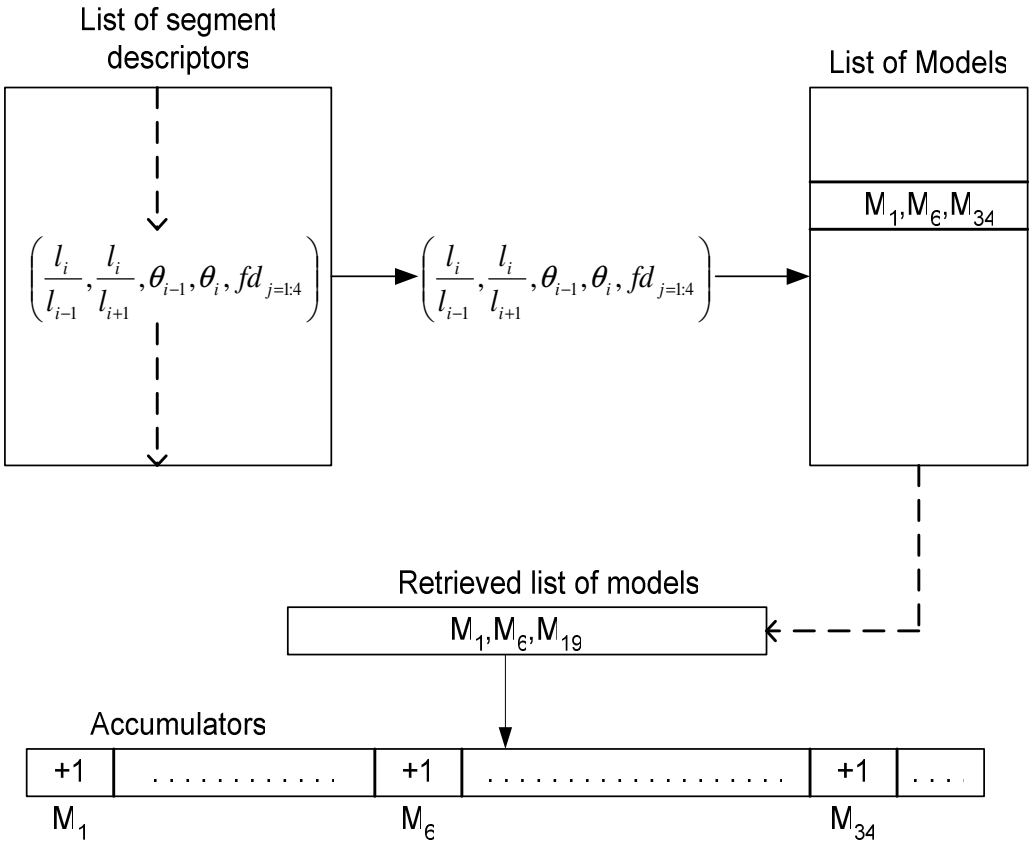


Figure 26: Generation of Hypothesis

Pruning of large portions of the model objects is carried out by keeping only some matched objects which received the highest vote.

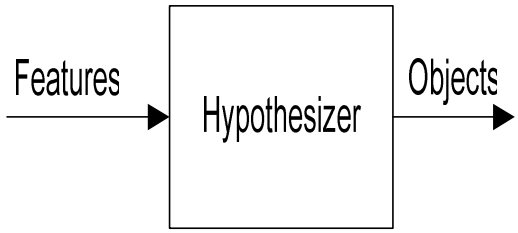


Figure 27: Block Diagram of Hypothesizer

6.7 Verifier

Pruning of large portion of the database reduces the search and enables the use of finer operators such as comparing the positioning of segments of the hypothesized objects. A library of index tables eight for each object, sorted in ascending order is built. Each of the index table entry consists of descriptor value and its corresponding segment number. We maintain a bucket of size equal to the number of contour segments for each hypothesized object. The contour segments of test object will be matched by searching in the indexed tables of the hypothesized objects. The block diagram of verifier is shown in Figure 29 and the block diagram of the whole algorithm is shown in Figure 30.

The segments matched will be marked with one and the other segments will be marked as zero's as shown in Figure 28.

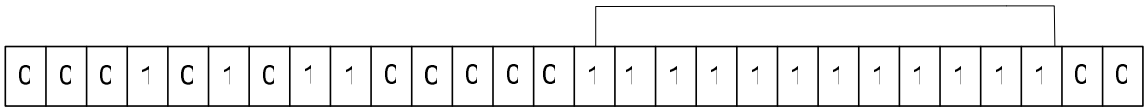


Figure 28: Marking of Segments

The hypothesized object which contains longest chain of one's will be a recognized object.

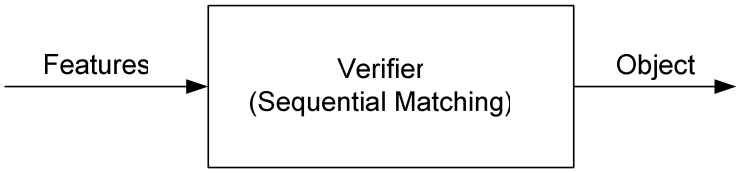


Figure 29: Block Diagram of Verifier

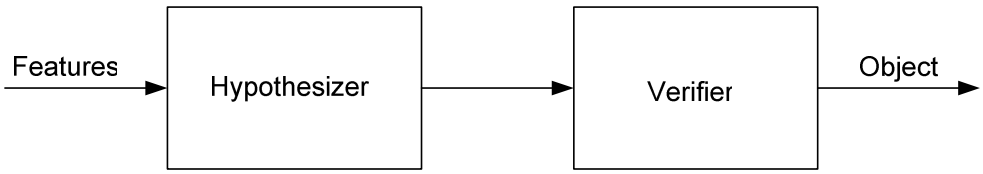


Figure 30: Block Diagram of Algorithm

The dependency of the recognition time over the size of the models is significantly reduced.

6.8 Results and Analysis

The test objects for testing similarity transformations were generated by translating, scaling and rotating the model objects same as in the previous experiments. The test objects were randomly rotated and translated, but scaled to factor of around $\frac{3}{4}th$, $1\frac{1}{4}th$ and some without scale of their model sizes. In case of generating the occluded test objects, the degree of occlusion is measured by the percentage of an object's boundary (in terms of curve length). The noisy test objects were generated in the same way as in the previous experiments. The objects used in database are kept in appendix A.

To test the time taken with the increase in the database size is conducted. Ten objects from the pool of sixty objects were randomly selected and the test objects were generated. Then the extensive random testing is done and the average time to recognize the object in case of similarity transformations was computed. Similarly, in the next stages twenty, thirty, forty, fifty, and sixty objects were selected from the pool of sixty objects and the random tests were conducted. The average time taken to the corresponding database size is shown in Figure 31. It is clear that the time taken does not vary linearly with increase in database size. The average time computed at each stage as the database size increased from ten to sixty was 5.25, 5.36, 5.47, 5.58, 5.76 and 5.88. The values show that the database size has almost negligible effect on recognition time as the database size increased from ten to sixty. Further, it can be inferred that the recognition time for a given test object and for particular database size depends on the number of segments of the test object.

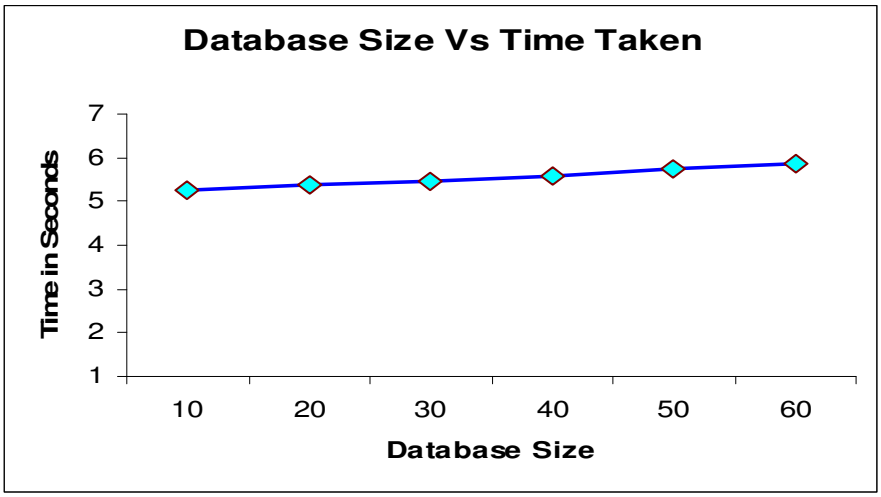


Figure 31: Effect of database size on recognition time

The experiment to determine the accuracy of recognition in case of similarity transformations is conducted by increasing the database size to ten in each stage. The objects for the database are selected from the pool of sixty objects in each stage randomly, and the test objects were generated and tested. Testing in each stage was repeated five times by randomly choosing different objects for the database except in case of fifty and sixty database size. The database size chosen in each stage was ten, twenty, thirty, forty, fifty, and sixty. The database size versus accuracy is plotted as shown in Figure 32. Each plotted point in graph represent the average accuracy at each stage after the experiment is repeated five times. The recognition rate dropped slightly with the increase in the database size at each stage. To represent random error, an *error bar,* consisting of a vertical line that extends from the mean value in proportion to the magnitude of the error was used. The error range in each stage is calculated by taking the standard deviation of results at that stage.

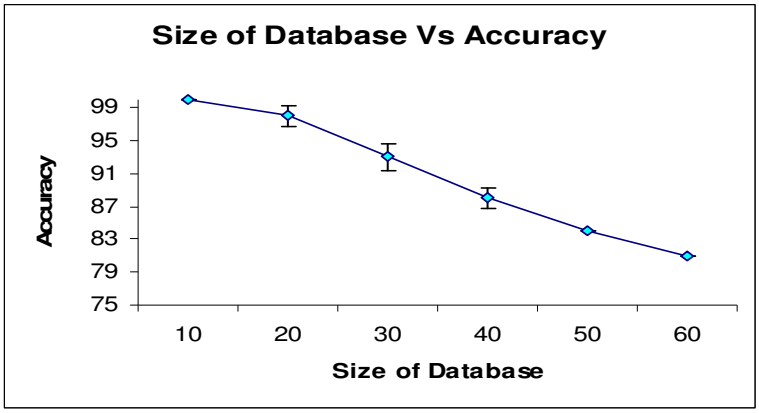


Figure 32: Effect of database size on accuracy in case of similarity transformations

The accuracy obtained as the database size increased from ten to sixty in case of just similarity transformations as shown in Figure 32 was hundred percent, ninety eight percent, ninety three percent, eighty eight percent, eight four percent and eighty one percent. The random errors shown as an error bars in Figure 32 have the vaues 0, 1.2, 1.7, 1.3, 0 and 0.

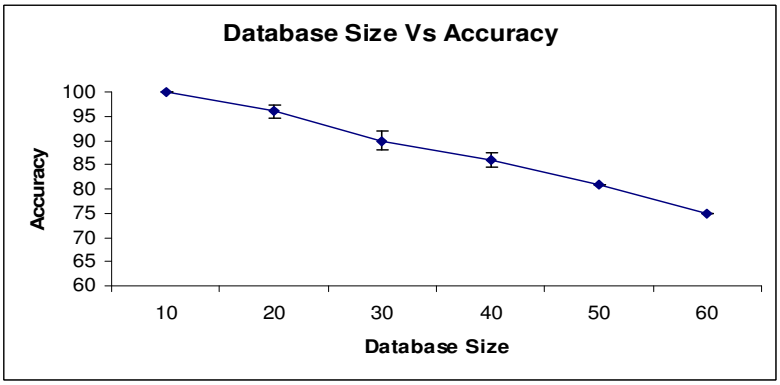


Figure 33: Effect of database size on accuracy in case of similarity transformations and occlusion

The accuracy obtained as the database size increased from ten to sixty in case of similarity transformations with occlusions up to twenty percent is shown in Figure 33. The accuracy was hundred percent, ninety six percent, ninety percent, eighty six percent, eight one

percent and seventy five percent. The random errors shown in Figure 33 as error bars have the values 0, 1.4, 1.9, 1.5, 0, 0.

The recognition rate in case of similarity transformation and noise of 10 % (Salt & Pepper noise) can be seen in Figure 34. The less recognition rate in case of noise may be due to the some faulty corner points. The effect of noise on the corner points is shown in Table 7. The effect of noise on the recognition rate is illustrated in Table 8.

Table 7 : Effect of noise on corner points

Corner Points 0% Noise		Corner Points 5% Noise		Corner Points 10% Noise		Corner Points 15% Noise	
x	y	x	y	x	y	x	y
395	240	394	239	393	237	394	237
394	217	393	217	396	215	391	212
286	237	287	232	289	234	285	235
257	127	257	127	256	123	253	123
308	66	306	66	301	63	305	63
251	87	251	83	247	82	245	81
247	70	246	72	244	74	242	67
241	87	240	83	237	83	236	82
186	65	186	65	183	62	188	63

236	130	234	128	231	124	235	127
201	236	198	231	200	233	198	231
66	211	65	209	63	214	64	207
241	453	240	450	237	447	237	448

The corner points shown in Table 7 indicate the faulty corners generated due to the addition of noise to the contour. With the increase in the noise level, the corner points were deviated more from its original position. The decrease the recognition rate with the increase in the noise level is depicted in Table 8 below. The recognition rate gradually decreased with the increase in the noise level.

Table 8: Effect of noise on recognition rate

Percent Noise	**Recognition rate**
0%	81%
5%	76%
10%	68 %
15%	66.66%

The accuracy obtained as the database size increased from ten to sixty in case of similarity transformations with noise up to ten percent (Salt & Pepper) is shown in Figure 34. The

accuracy was eighty seven percent, eighty three percent, eighty percent, seventy five percent, seventy one percent and sixty eight percent. The random errors shown in Figure 34 as error bars have the values 0, 1.3, 2.3, 1.6, 0, 0.

Figure 34: Effect of database size on accuracy in case of similarity transformations and noise

The recognition rate in case of similarity transformation, noise 10% (Salt & Pepper) and occlusion up to twenty percent can be seen in Figure 35. The accuracy was eighty four percent, seventy nine percent, seventy four percent, seventy one percent, sixty seven percent and sixty two percent. The random errors shown in Figure 35 as error bars have the values 1.5, 2.3, 2.8, 2.4, 0, 0. The considerable drop in recognition rate when compared to previous cases is due to the faulty corner points generated because of noise and occlusions. The accuracy in case of similarity transformations, noise and occlusion indicates that the approach has considerable advantage when dealing with occlusion and noise. Further, the non linear search makes the approach better in performance when compared to other sequential search approaches.

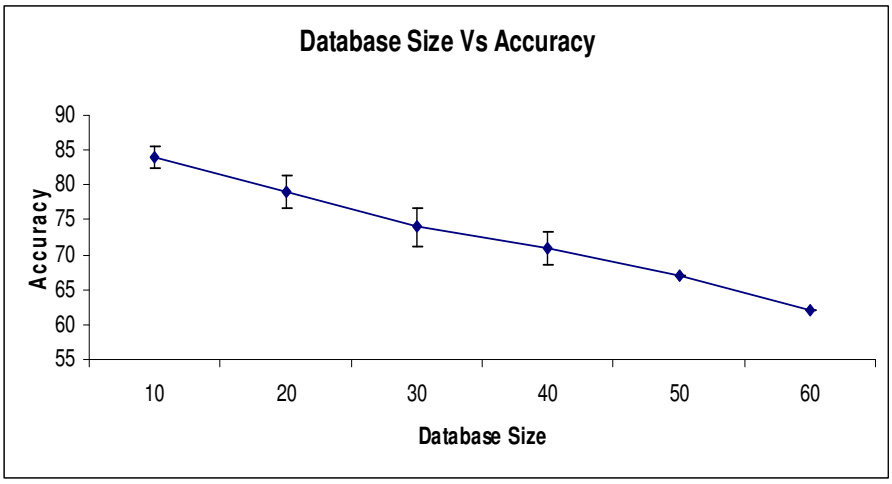


Figure 35: Effect of database size on accuracy in case of similarity transformations, noise and occlusion

6.9 Results comparison

The recognition rates of the three experiments, using moment invariants, Fourier descriptors and indexing methods are compared. The recognition rate in case of similarity transformations using indexing approach excelled other methods and it has performed well in case of occlusion. The number of test objects matched in each experiment is depicted in Table 9 and the percentage recognition in case of similarity transformations in Figure 36 and percentage recognition rate in case of similarity transformations and noise in Figure 37.

Table 9: Comparison of recognition rates of various experiments

Tests / methods	Similarity transformations (100 test objects)	Similarity transformations+ Noise (10%) (100 test objects)	For 20% Occlusion (for 100 test objects)	Similarity transformations+ Noise (10%) + Occlusion (20%) (100 test objects)
Using Moment invariants	61	59	–	–
Using Fourier transforms	85	83	–	–
Using Indexing Approach	81	68	75	62

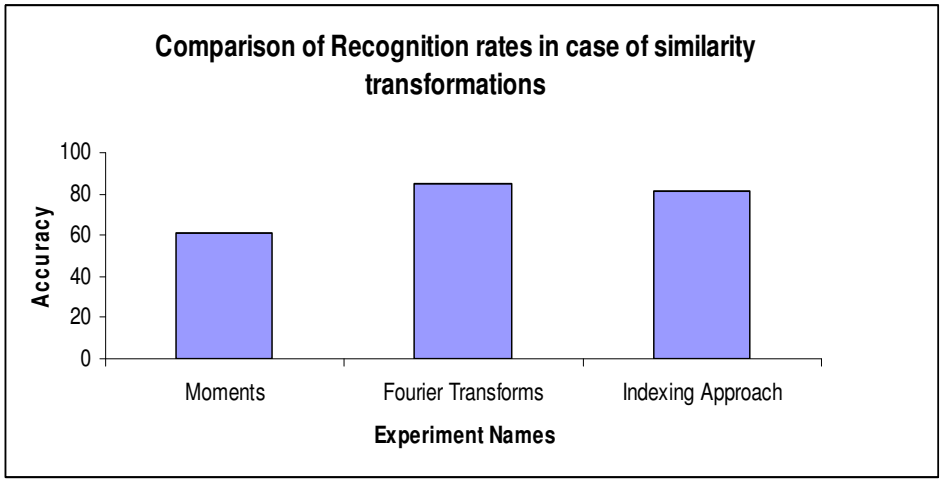


Figure 36: Comparison of recognition rates of different experiments in case of similarity transformations

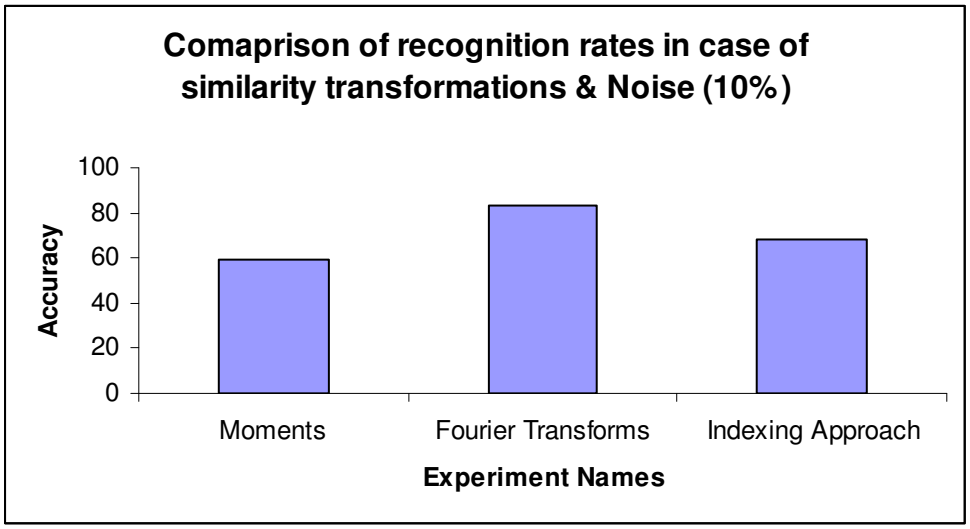


Figure 37: Comparison of accuracies in case of similarity transformations & noise

6.10 Performance Comparison

The performance of experiments using moments, Fourier transforms and indexing approach were compared in case of similarity transformations shown in Figure 38. The experiment of using Fourier descriptors out performed the other experiments with an average recognition time of 0.51 seconds, even though the feature vectors are of almost same size. The reason behind this large difference is due to the large values of moment descriptors as compared to Fourier descriptors.

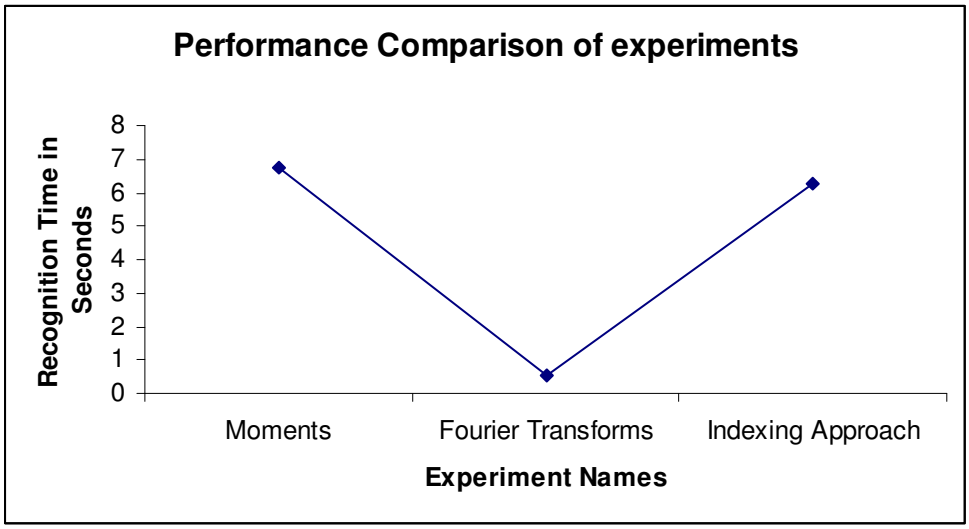


Figure 38: Performance comparison of different experiments

6.11 Time Complexity

Since the time taken in hypothesis generation and verification is constant, the time complexity is linear with respect to the number of segments in the test object.

Suppose there are fifty objects in the database and each has twenty segments on an average then the number of entries in each index table will be 50x20=1000 entries. So, to search the list of above size for a given descriptor within some threshold will take 10 units of time. So, the time taken is constant for searching any descriptor.

Suppose there are n numbers of segments in the test object and if each segment takes k units of time, then the total time taken is kn. Again in the verification phase, we will verify constant number of hypothesized objects each time and the time taken is constant. Therefore the total time is of order kn+k. So, the time complexity is O(n).

CHAPTER SEVEN

CONCLUSION

7.1 Introduction

In this chapter, we will present the summary of the work and how it can be improved in future.

7.2 Summary and Contributions

This thesis presented two research approaches for 2-dimensional object recognition that are invariant to similarity transformations and noise. Another attempt of two different methods are proposed to overcome the problem of occlusion. The first research approach (Shape Space) failed to recognize the random occlusions and time complexity of the approach is very high. In the second research approach, the segmentation of contour is accomplished by detecting corner points, and then the local shape descriptors were used to describe the segments. A library of index tables was used to generate the hypothesis and verification phases. The generation of hypothesis reduced the time of recognition considerably by pruning the large number of database objects. Further, the indexing

approach's recognition time is not dependent on the size of the database, which is the greatest advantage over the other methods.

The percentage recognition in case of similarity transformations, noise and occlusion was compared with other experiments. The overall results showed that the indexing approach excelled other methods.

The performance of indexing approach when compared to shape space approach is far better. Typically, the time required to recognize is dependent on number of segments rather than database size in indexing approach, where as in shape space the time required is dependent on the number of landmarks as well as database size. This concludes that the proposed approach of indexing is both effective as well as efficient in all cases when compared to other methods proposed.

7.3 Limitations

The limitations of the proposed indexing approach are as follows:

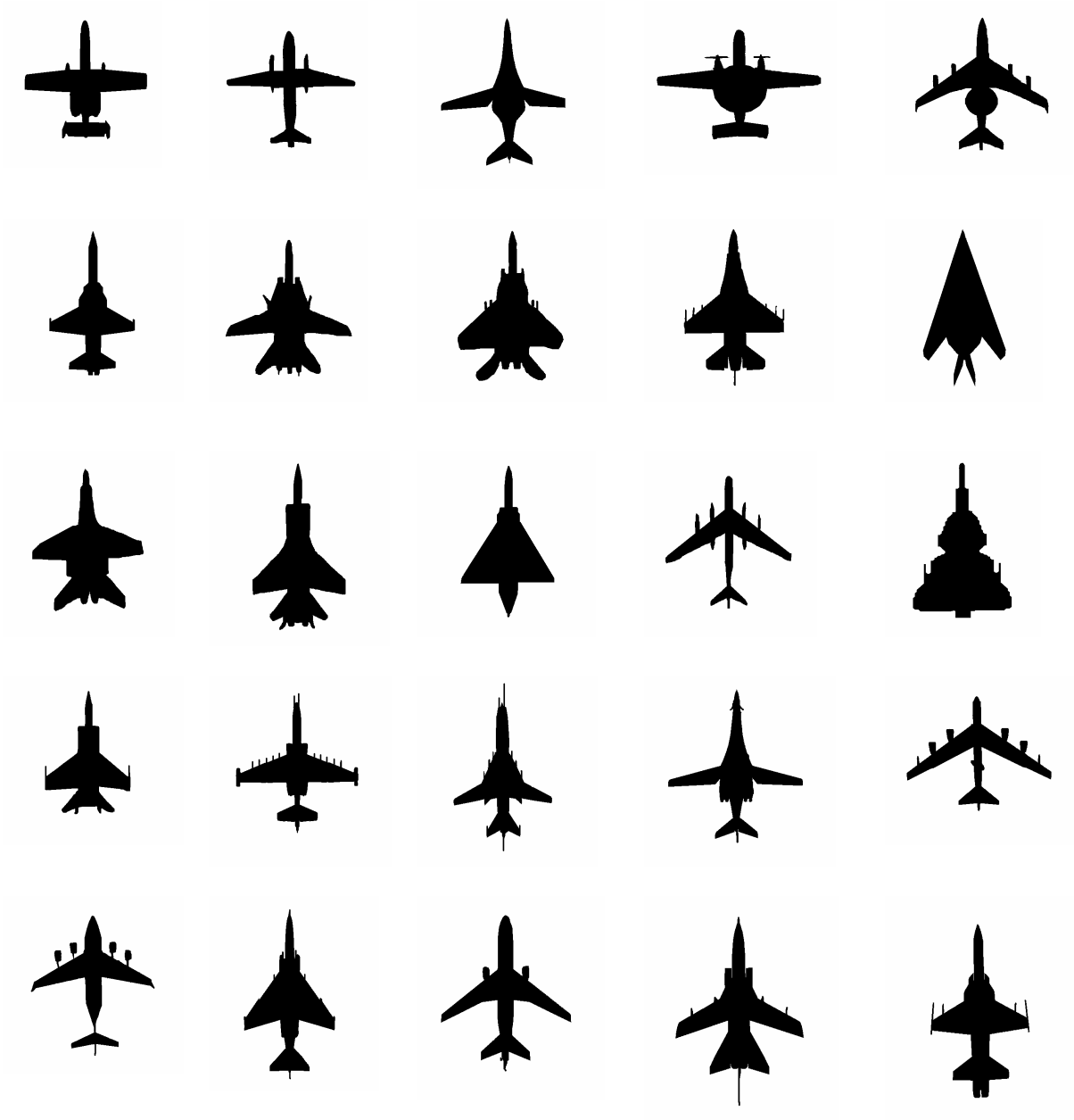
- The approach considered only the outer contour of the objects for recognition. So, the outer contours of different objects may not be more discriminative and further occlusion may cause false recognition.
- The proposed method of contour segmentation of using corner points is susceptible to error in case of more noise on the boundary.

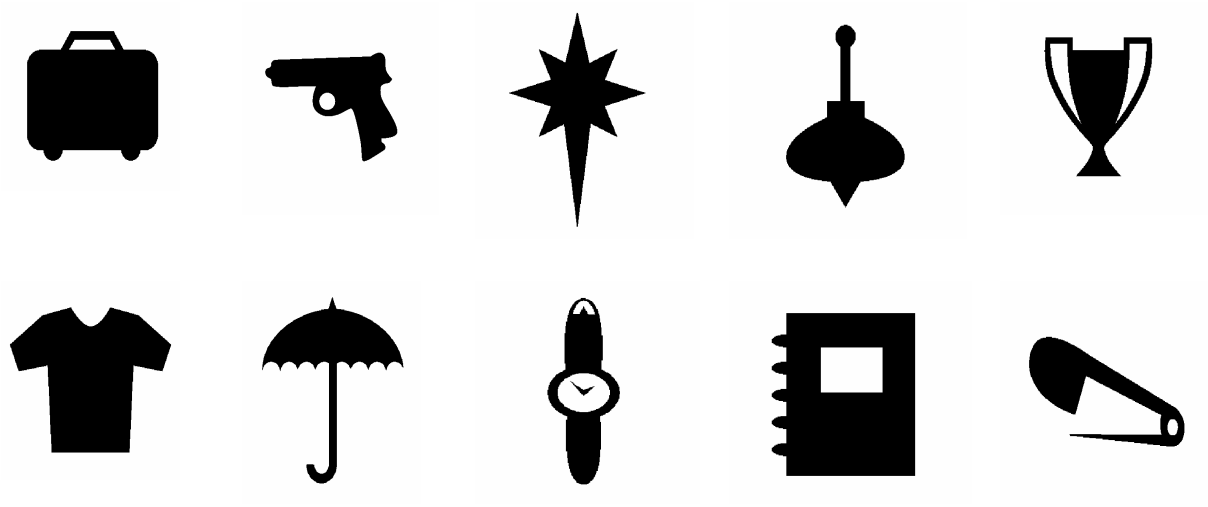
7.4 Future work

As a future work, we can develop some robust contour segmentation algorithm which should be consistent in case of occlusion with noise. We can also use other information of the objects, such as inner contours for more discriminative information.

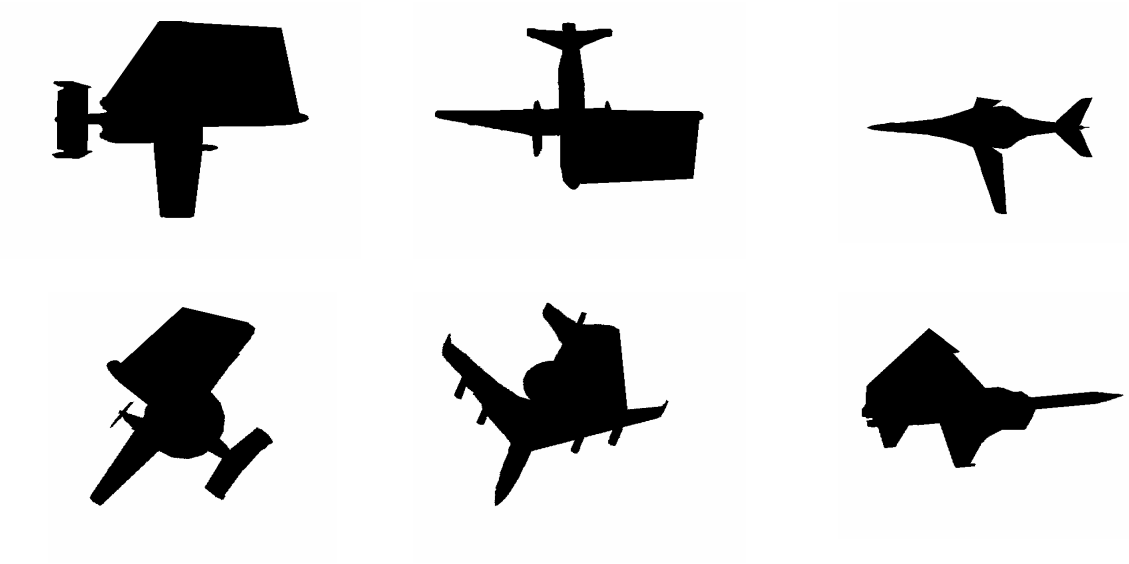
Appendix A

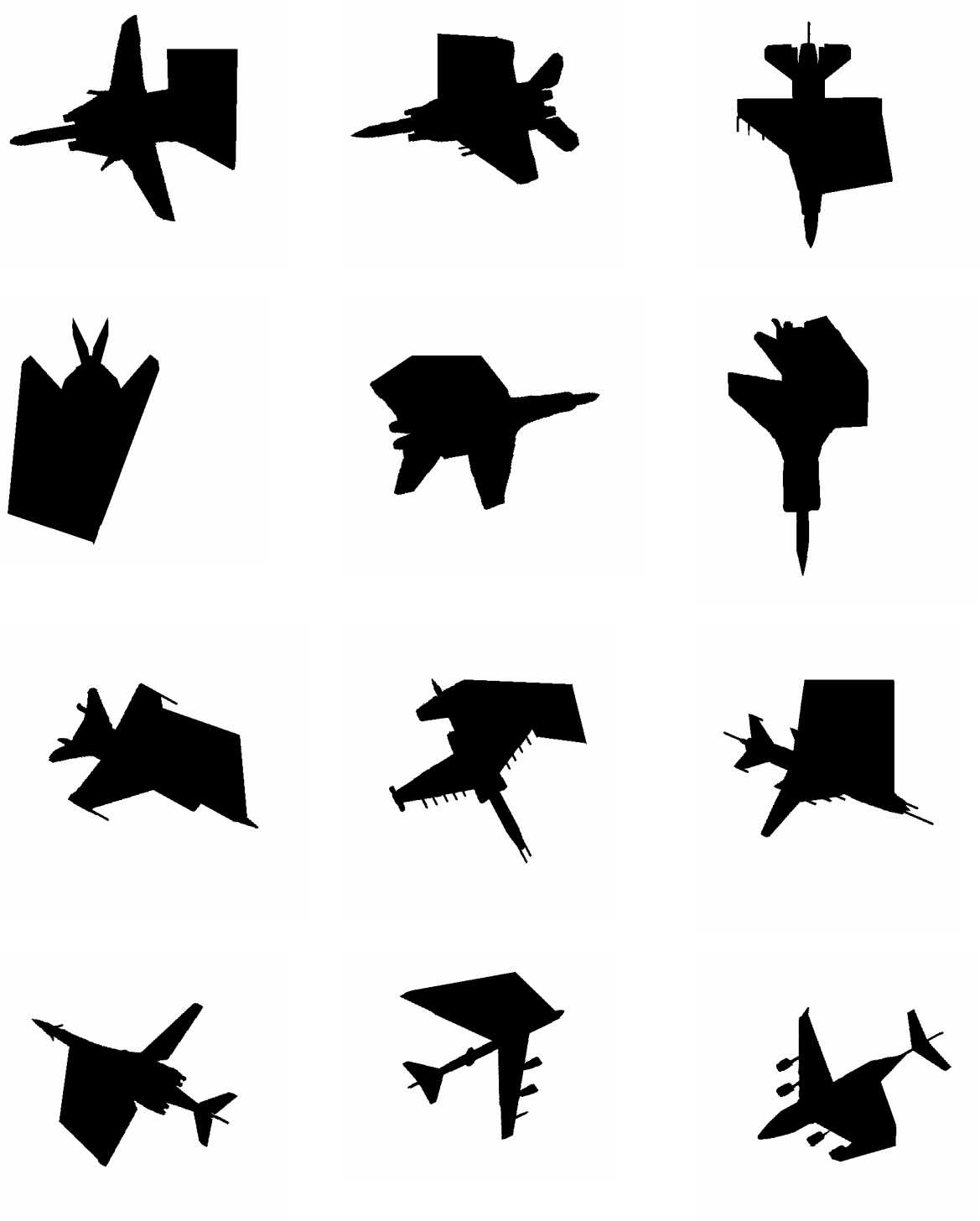
Database Objects

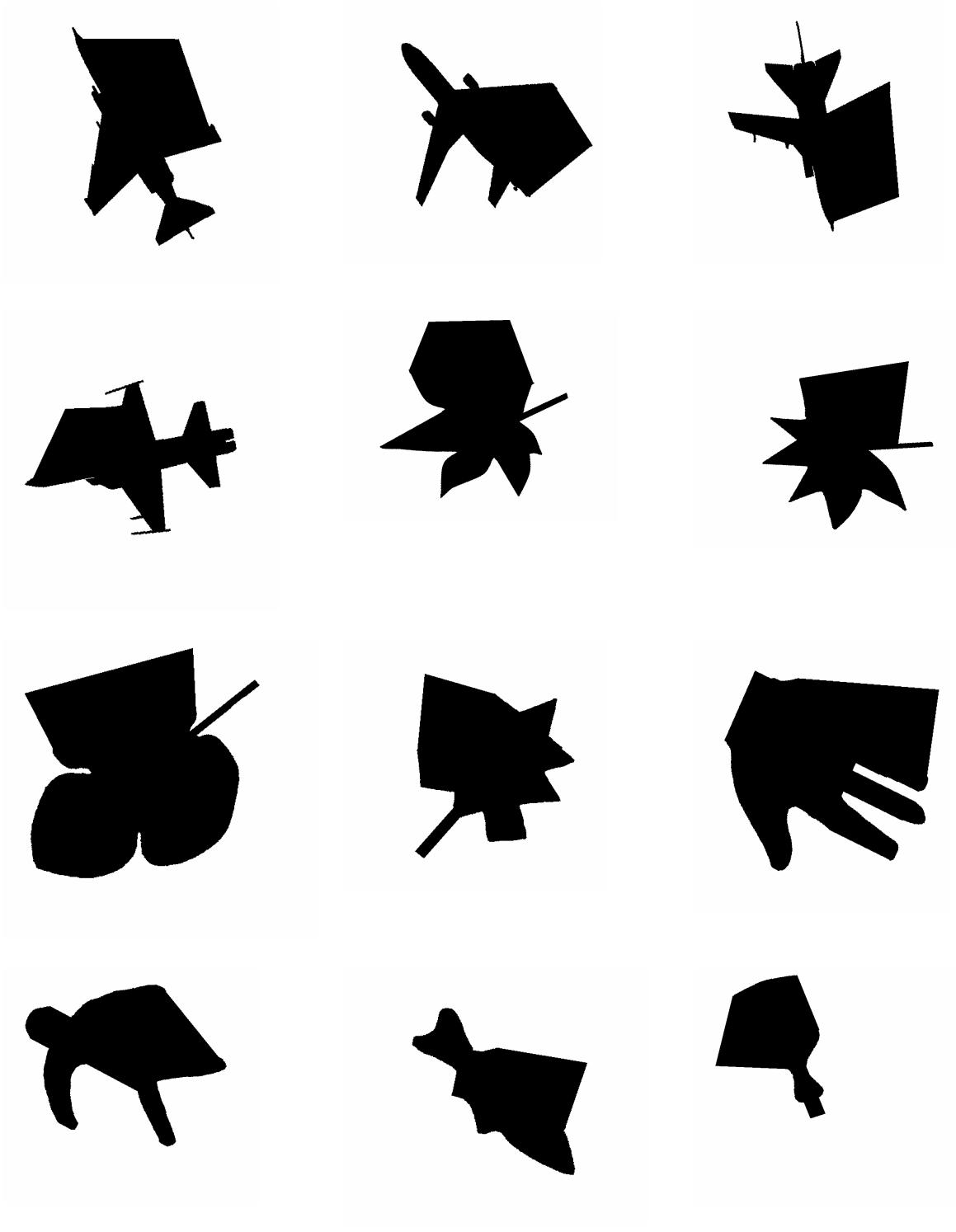


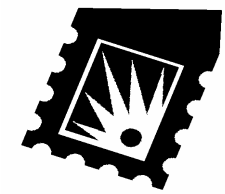
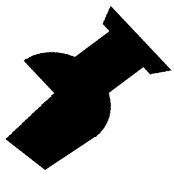


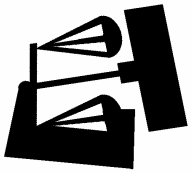
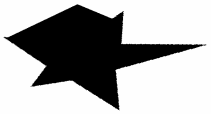
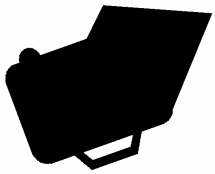
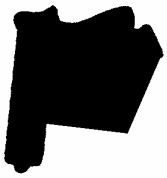
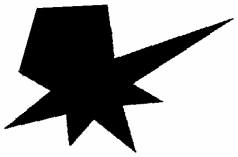
Occluded Objects

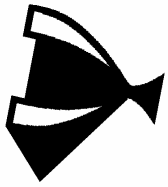
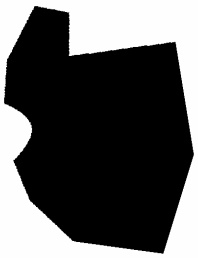
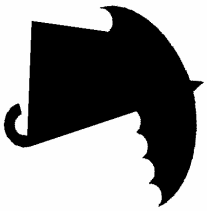


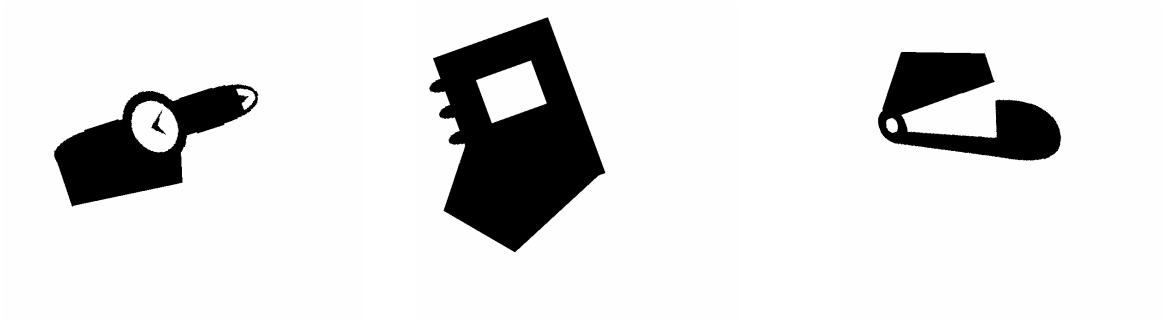






References

[1] M.K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inform. Theory 8 (1962) 179}187.

[2] Y. Li, Reforming the theory of invariant moments for pattern recognition, Pattern Recognition 25 (1992)723-730.

[3] W.H. Wong, W.C. Siu, K.M. Lam, Generation of moment invariants and their uses for character recognition, Pattern Recognition Lett. 16 (1995) 115-123.

[4] C.T. Zahn, R.Z. Rhoskies, Fourier descriptors for plane closed curves, IEEE trans. Compu. 21 (1972) 269-281.

[5] L. Stark, K. Bowyer, Generic Object Recognition using form and function, World Scientific, Singapore, 1996.

[6] Thomas Bernier, Jacques-Andre landry, A new method for representing and matching shapes of natural objects, Pattern Recognition 36(2003) 1711-1723.

[7] N. Ansari, E.J. Delp, Partial Shape Recognition: a landmark based approach, IEEE Trans. PAMI 12 (1990) 470-183.

[8] S. jaggi, Multiscale geometric feature extraction and object recognition, MIT Phd thesis(LIDS-TH 2384),1997.

[9] R.T. Chin, C.R. Dyer, Model-based recognition in Robot vision, ACM comput, Surveys 18 (1986) 67-108.

[10] Che-Bin Liu, Narendra Ahuja, A model for dynamic shape and its applications, In 20th international conference on computer vision and pattern recognition, Volume II-129(2004).

[11] Lena Gorilick, Meirav Galun, Eitan Sharon,Ronen Basri, Achi Brandt, Shape representation and classification using poisson equation, In 20th international conference on computer vision and pattern recognition, Volume II-129(2004).

[12] John W. Gorman, O Robert Mitchell, Frank P. Kuhl, Partial shape recognition using dynamic programming, IEEE Transactions on pattern analysis and machine intelligence, Vol.10,No.2, March 1988.

[13] John W. Gorman, Richard M. Ulmer, Partial shape recognition using simulated annealing, Proc. Of Energy and Information Technologies in the Southeast, 959-964 vol.3, 1989.

[14] G.J. Ettinger, Large hierarchical object recognition using libraries of parameterized model sub-parts, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1988, pp.32-41.

[15] Hong-Chih Liu and Mandyam D. Srinath, Partial Shape Classification using contour matching in distance transformation, IEEE Transactions on pattern analysis and machine intelligence, Vol.12, No. 11, November 1990.

[16] Nanning Zheng, Yaoyong Li, Wiek P.M. Houwers, Local feature-based recognition of partially occluded objects using neural network, 21st International Conference on Industrial Electronics, Control, and

Instrumentation,vol. 2 1301-1306 1995.

[17] A. Califano and R. Mohan, Multidimensional indexing for recognizing visual shapes, Proc. IEEE Conf. Computer Vision and Pattern Recognition, June 1991, pp.28-34.

[18] Joong-Hwan Baek, Keith A. Teague, Occluded Object Recognition using extended local features and hashing, Proc. 1994 IEEE Int. Conf. on Sys., Man, and Cybernetics, San Antonio, Texas, October 2-5, 1994.

[19] M. Al-Mouhamed, An efficient indexing scheme for image storage and recognition, IEEE Trans. Industrial electronics, vol.46, pp.429-439, April 1999.

[20] M. Al-Mouhamed, A Robust Gross-to-Fine pattern recognition system, IEEE Trans. On industrial electronics, vol. 48, No. 6, December 2001.

[21] R.J. Schalkoff, "Pattern Recognition: Statistical, Structural and Neural Approaches", John Willey and Sons, Inc., (1992).

[22] K.Y. Huan, "Syntactic Pattern Recognition for Seismic Oil Exploration", Series in Machine Perceptron and Artificial Intelligence, Vol. 46, pp. 1-3, (2002).

[23] Devijver and J. Kittler, "Pattern Recognition: A Statistical Approach", London: Prentice Hall, (1982).

[24] S.O. Belkasim, M. Sridhar and A. Ahmadi, "Pattern Recognition with Moment Invariants: A Comparative study and new results", Pattern Recognition,

Vol.24, pp.1117-1138, (Dec 1991).

[25] A. Khotanzad and Y.H. Hong, ”Rotation invariant image recognition using features selected via a systematic method”, *Pattern Recognition,* Vol. 23, no. 10, pp.1089-1011, (1990).

[26] A. Khotanzad and Y.H. Hong, ”Invariant image recognition using Zernike Moments”, IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 16, no. 10, pp. 976-986, (Oct 1994).

[27] D.G. Kendall, Shape Manifolds, Procrustean metrics, and complex projective spaces, Bull. London Math. Soc. 16(1984) 81-121.

[28] MS Thesis Work of Murtaza Ali Khan, King Fahd University of Petroleum and Minerals. Year 2000.

[29] Dudani, S. A., Breeding, K. J., McGhee, R.B., “Aircraft identification by moment invariants”, *IEEE-T on Computers*, Vol. 26, No.1, pp. 39-46, Jan. 1977.

[30] A.K .Jain, J. Mao and K. Mohiuddin, “Artificial Neural Networks: A tutorial”, IEEE Computer special issue on Neural Computing, pp.1-49, (1996).

[31] MS Thesis Work of Syed Nazim Nawaz, King Fahd University of Petroleum and Minerals. Year 2003.

[32] F. Mokhtarian, Silhouette-based isolated object recognition through curvature scale space, *IEEE Trans. Pattern Anal. Mach. Intell*. **17**, 1995, 539–544.

[33] http://www.ee.calpoly.edu/~fdepiero/STL/STL%20-%20Image%20-%20Median%20Filtering.htm

[34] http://www.cee.hw.ac.uk/hipr/html/median.html

[35] L. A. Torres-Méndez, *et all ,* Translation, Rotation, and Scale-Invariant Object Recognition, IEEE Transactions on systems, man , and cybernetics applications and reviews, Vol. 30, no. 1, February 2000 125

[36] S. Berretti, A. Del Bimbo, P. Pala, Indexed Retrieval by Shape Appearance, IEE Proceedings on Vision Image and Signal Processing, vol.147, no.4, pp.356-362, August 2000.

[37] Gonzalez, R. C. Wintz, P. Digital Image Processing. Addison--Wesley, Reading, MA.

[38] Granlund, G.H., 1972 Fourier Pre-processing for Hand Print Character Recognition, IEEE Transactions on Computers, Vol. C-21, 195-201.

[39] J.S. Beis and D.G. Lowe. Shape indexing using approximate nearest-neighbor search in high dimensional spaces. In Proc. IEEE Conf. Comp. Vision Patt. Recog., pages 1000--1006, 1997.

[40] D. D. Hoffman and W. A. Richards. Parts of Recognition. Cognition 18, 65--96, 1984.

VITA

Ahmed Abdul Mujeeb Qhusro, was born in Nalgonda, India, obtained Bachelor of Engineering (BE) in Computer Science & Engineering from Osmania University, Hyderabad, India in April 2000. Prior to attending King Fahd University of Petroleum & Minerals, he worked as a lecturer from January 2001 to January 2002 in Department of Computer Science & Engineering in Deccan College of Engineering & Technology, Hyderabad, India. He joined King Fahd University of petroleum & Minerals (KFUPM) in March 2002 as a research Assistant to pursue the masters in Information & Computer Science department. He successfully defended his thesis in December 2004. Ahmed Abdul Mujeeb's research interests include Pattern Recognition, Distributed Computing, Operating systems, Computer Vision, Computer Graphics, and Software Engineering. He can be reached at qhusro@yahoo.com