RECOGNITION OF PARTIALLY OCCLUDED COMPLEX INDUSTRIAL OBJECTS USING FOURIER DESCRIPTORS

A Thesis Presented to the

Faculty of Graduate Studies

University of Manitoba

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Mechanical Engineering

presented by

Ravi Rai

(c) August 1990

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-71771-8

Canada

RECOGNITION OF PARTIALLY OCCLUDED COMPLEX INDUSTRIAL OBJECTS USING FOURIER DESCRIPTORS

BY

RAVI RAI

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

© 1990

Permission has been granted to the LIBRARY OF THE UNIVERSITY OF MANITOBA to lend or sell copies of this thesis. to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film, and UNIVERSITY MICROFILMS to publish an abstract of this thesis.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

ACKNOWLEDGEMENT

I thank my advisors Dr. S. Balakrishnan and Dr. J. Jenness for their support and guidance during the course of this thesis.

I would like to acknowledge my wife's efforts during the last few years which enabled me to undertake graduate studies. I would also like to acknowledge the encouragement I received from my parents.

Lastly I wish to thank the Lord for providing me the opportunity to pursue studies in a world where so many people have so little opportunity.

ABSTRACT

A method is presented for the recognition of complex 2D-industrial objects by matching extended features and internal features in Fourier Space. The technique places no limitation on the shape of complex objects. The technique enables recognition of complex objects when they are partially occluded with touching or overlapping boundaries.

To implement the proposed technique, a new type of extended features are proposed. These extended features are unique in that they encompass structural information on the location of internal features in an object. Also a slightly modified representation of existing Fourier Descriptors is defined. Unlike Fourier Descriptors defined by some other investigators, the proposed Fourier Descriptors are not magnitude invariant and hence allow size discrimination between objects.

A preliminary microcomputer based part recognition system has been implemented for the recognition of parts with multiple internal features such as holes, slots etc.. The system has two modes of operation (i) a "training" mode for the automatic generation and storage of models in the data base and (ii) a "runtime" mode where object models are

automatically generated and compared with candidate models in the database.

The main emphasis in this thesis is on the introduction of a technique for the recognition of partially occluded, complex, industrial objects using Fourier Descriptors, as opposed to the actual mechanics of the matching process or the functioning of the database storage and retrieval tasks.

TABLE OF CONTENTS

CHAPTER

ACKNOWLEDGEMENT i

Abstract ii

CHAPTER 1 : INTRODUCTION 1

CHAPTER 2 : LITERATURE REVIEW 4

2.1 Area Methods 4

2.2 Boundary Methods 6

2.2.1 Polygonal Approximations 6

2.2.2 Hough Transforms 12

2.2.3 Fourier Descriptors 16

2.2.4 Other Boundary Methods 21

2.3 Special Axes Methods 25

2.4 Existing Systems for Complex Industrial Objects .. 28

2.4.1 Model Based System 28

2.4.2 Local Feature Focus Method 30

2.4.3 Boundary Matching Using Footprints 31

2.5 Summary 33

CHAPTER 3 : RECOGNITION USING FOURIER DESCRIPTORS OF EXTENDED FEATURES 36

3.1 Proposed Fourier Descriptors 36

3.1.1 Properties of FD With Respect to Shape 44

3.1.2 Properties of FD With Respect to Rotation 53

3.1.3 Properties of FD With Respect to Scale 58

3.1.4 Properties of FD with Respect to Image Reconstruction 62

3.1.5 Summary of Proposed Fourier Descriptors 62

3.2. Recognition of Complex Industrial Objects 66

3.2.1 Heuristic #1 for the Generation of Extended Features and Fourier Descriptors for Objects With Two or More Internal Features 71

3.2.2 Heuristic #2 for the Generation of Extended Features and Fourier Descriptors for Objects With Less Than Two Internal Features 77

3.3 Recognition of Overlapping and Touching Objects .. 81

3.4 Summary: Recognition of Partially Occluded, Complex Industrial Objects 87

CHAPTER 4 : EXPERIMENTAL RESULTS 89

4.1 Operator Interface 90

4.2 Format of Data in Object Model 95

4.3 Recognition of Partially Occluded Objects 98

4.4 Recognition of Basic Shapes 114

4.5 Dimensional Analysis of Repeatability 116

4.6 Summary of Experimental results 118

CHAPTER 5 : DESCRIPTION OF EXPERIMENTAL SYSTEM 120

5.1 Image Processing Hardware 120

5.2 Software Developed 120

5.2.1 Thresholding 121

5.2.2 Edge Detection 121

5.2.3 Detection of Internal Features in Edge Image 124

5.2.4 Area and Center of Geometry of Internal Feature 125

5.2.5 System Calibration 125

5.2.6 Matching of Internal and Extended Features in "Training" and "Runtime" Modes 126

5.2.7 Matching of Partially Occluded Object With Database Objects 127

CHAPTER 6 : CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK 128

6.1 Limitation of Current Implementation and Future Work 129

6.2 Future Work to Improve the Basic Technique 130

REFERENCES 131

APPENDICES

A1: Listing of Interaction with System Software for Adding an Object (serial #2222) to the Database 134

A2: Listing of Interaction with System Software for Recognition of an Object (serial #2222) in the Database 137

A3: Database Containing Models of Six (serial #1111, #2222, #3333, #4444, #5555, #6666) Objects 142

A4: Object Model for Serial #2222 Obtained during "Runtime" Mode 159

LIST OF FIGURES

FIGURE		PAGE
2.1	(a) Definition of θk and the Incremental Curvature δk at Node k, Given an Aperture Size w = 5 (b) The Plot of Incremental Curvature δk at Node k (w = 5) for a Chain Coded Curve Possessing a 90° Turn	8
2.2	Iterative Technique for Generating a Polygonal Approximation to a Curve	11
2.3	(a) Three Collinear Points on a Line y = 2x + 1 (b) Plot of m = y/x - c/x for the Three Collinear in Figure 2.3a	13
2.4	Parametric Representation of a Plane Curve with Tangential Direction θ(l) and Cumulative Angular Bend Function ø(l)	17
2.5	Parametric Representation of a Plane Curve with Radius-vector as a Function of Angle	19
2.6	Limitations of the Radius-Vector vs. Angle Technique (a) Intersection of Radius-vector with Two Boundary Points (b) Intersection of Radius-Vector with Three Boundary Points	22
2.7	Creation of a Stick Figure Using " Grass Fire" Analogy	26
2.8	Creation of a Stick Figure Using " Maximal Disk Technique	27
3.1	(a) Representation of a Closed Curve Where the Centroid Lies Within the Curve, (b) Representation of a Highly Concave curve where the Centroid Lies Outside the Curve.........	38
3.2	4" x 8" Rectangle	40
3.3	A Plot of r(θ) vs. θ for 4" x 8" Rectangle in Figure 3.2	42
3.4	4" Square	45

FIGURE		PAGE

3.5 A Plot of r(θ) vs. θ for 4" Square in Figure 3.4 47

3.6 4" Diameter Crcle 50

3.7 A Plot of r(θ) vs. θ for the 4" Diameter Circle 51

3.8 4"x 8" Rectangle of Fig. 3.2 Rotated by 22.5 Degrees 54

3.9 2" x 4" Rectangle 59

3.10 Reconstructed (from Fourier Domain) a Plot r(θ) vs.θ for 4" x 8" Rectangle in Figure 3.2 65

3.11 Complex Industrial Object With Four Internal Features 68

3.12 Location of the Focus and Extended Features for Object in Fig. 3.11 69

3.13 Complex Object With Internal Features Having More than One Critical Point 74

3.14 A Complex Part with Six Internal Features 76

3.15 Object with No Internal Features 79

3.16 Object with One Internal Feature 80

3.17 A Collection of 2-D Objects 84

3.18 Puzzle Cnsisting of Overlapping and Touching Objects of Fig. 3.17 85

4.1 Complex Object with Serial #2222 in Database 91

4.2 Format of Data in the Model of an Object with Four Internal Fatures 96

FIGURE		PAGE
4.3	Data Format of Every Feature in an Object Model	97
4.4	Complex Object with Serial # 1111 in Database	100
4.5	Complex Object with Serial # 3333 in Database	101
4.6	Complex Object with Serial # 4444 in Database	102
4.7	Complex Object with Serial # 5555 in Database	103
4.8	Complex Object with Serial # 6666 in Database	104
4.9	Partially Occluded Object	105
4.10	Partially Occluded Object	106
4.11	Partially Occluded Object	107
4.12	Partially Occluded Object	108
4.13	Partially Occluded Object	109
4.14	Partially Occluded Object	110
4.15	Partially Occluded Object	111
4.16	Partially Occluded Object	112
5.1	3 x 1 Pixel Filters Used for Edge Detection	122
5.2	Hexadecimal Coded Edge Image of a Portion of the Object in Figure 4.1	123

LIST OF TABLES

TABLES		PAGE
2.1	List of Features That Can be Extracted From a Plot of Incremental Curvature	10
3.1	r(θ) vs. θ for 4" x 8" Rectangle in Figure 3.2	41
3.2	Fourier Descriptors for 4"x 8" Rectangle in Figure 3.2	43
3.3	r(θ) vs. θ for 4" Square in Fig. 3.4	46
3.4	Fourier Descriptors for 4" Square in Fig. 3.4	49
3.5	Fourier Descriptors for 4" Circle in Fig. 3.6	52
3.6	r(θ) vs. θ for 4" x 8" Rotated Rectangle in Figure 3.8	56
3.7	Fourier Descriptors of 4"x 8" Rotated Rectangle in Fig. 3.8	57
3.8	Fourier Descriptors of 2" x 4" rectangle in Fig. 3.9	60
3.9	Comparison of Fourier Descriptors of 4" x 8" and 2" x 4" Rectangles in Fig. 3.2 and 3.9 respectively	61
3.10	Fourier Descriptors of Fourier Descriptors of 4"x 8" Rectangle in Figure 3.2	63
3.11	Negative and Positive Frequency Terms Reconstructed From Table 3.10	64
3.12	The Impact of Segmentation of Object Boundary On Object Recognition	86
4.1	Recognition Exercise Involving Partially Occluded Objects	113
4.2	Fourier Descriptors of Basic Shapes, Extracted from Models of Figure 4.1 and 4.3 in the Database in Appendix A3	115

TABLES PAGE

4.3 Statistical Analysis for Repeatability on the Dimensions of Internal Features of Objects in Figure 4.1 117

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

One of the challenges facing factory automation is the automatic retrieval of industrial objects that have complex shapes by autonomous robots. Currently, commercially available systems can recognize isolated two dimensional parts against contrasting backgrounds. However the problem of object recognition is considerably more complex when objects are partially occluded and are touching or overlapping. This type of problem can be divided into four sub-problems [Bolles and Cain, 1982] as follows:

(a) portion of an object.

(b) collection of objects when one or more are touching.

(c) collection of objects when one or more are overlapping.

(d) collection of one or more objects that are defective dimensionally.

1.2 Research Objectives

Object size is important in discriminating between industrial objects, and hence Fourier Descriptors used in the recognition process should not be invariant to size. The location of internal features with respect to the global object boundary is also important in discriminating between objects. Thus structural information should be incorporated in the shape model.

With the above technical objectives, a system is deviced comprising of (i) an unconventional set of Fourier Descriptors which are not size invariant and (ii) a newly proposed extended feature which encompasses structural information.

1.3 Thesis Overview

Chapter 2 reviews existing literature and consists of the following; (i) basic shape recognition techniques such as the method of moments, polygonal approximation, Hough Transform and Fourier Descriptors and (ii) existing systems for the recognition of complex industrial parts. Chapter 3 describes (i) the proposed Fourier Descriptors, (ii) the proposed extended features, (iii) the basic approach for recognizing complex industrial objects and how the basic approach can be extended to address some of the aforementioned sub-problems. Chapter 4 presents experimental results from a preliminary system for the recognition of partially occluded objects. Statistical analysis on the repeatability of dimensional data, for the identification of defective objects is also presented. In chapter 5, hardware and software employed in the experimental work is described. A brief description of (i) primitive image processing functions such as thresholding, edge detection, segmentation of edge points into internal features and (ii) the method employed in matching "runtime" and "training" mode object models is given. Chapter 6 presents conclusions and recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

Shape recognition is based upon shape descriptors. Two simple shape descriptors are an object's area and perimeter. A slightly more complicated shape descriptor for example is the smallest circumscribing rectangle. As yet there is no general agreement on a minimum set of shape descriptors to quantify a given object.

We shall first review basic shape recognition techniques based upon (a) area methods, (b) boundary methods and (c) special axes methods. Then we shall review some integrated object recognition systems, based upon some of the aforementioned shape recognition techniques.

2.1 Area Methods

The most common area method is the method of moments. This was first proposed by Hu [1962]. The $(p,q)^{th}$ moment of a shape is defined as follows;

$$m(p,q) = \int x^p y^q r(x,y)\, dx\, dy \qquad p,q = 0,1,2.. \quad (2.1)$$

where $r(x,y)$ is equal to 1 in the bounded region of the shape and 0 outside.

This method has produced moments that are invariant under image rotation, translation and scaling and this makes the technique desirable for object recognition. The technique also has some disadvantages and these are as follows;

(1) A simple shape may require just the first few moments but a complicated shape will require many more. This puts a severe limitation on practical applications as the calculation of moments is computational intensive.

(2) The distortion of the image by noise and poor segmentation is not easily modelled using moments [Faugereras, 1983].

(3) When dealing with partially occluded or agglomerated objects, the moments of the resulting shape are radically different and bear no structural resemblance to those of the original shape.

Another approach for shape modelling includes segmenting the shape into convex subregions [Kurozuml, 1982, Pavlidis, 1977]. Given a set of object points, S, a convex region is defined as one in which (i) for every two points, p and q in S, the line segment from p to q lies entirely in S or (ii) for every point p and q in S, the midpoint of the line segment from p to q lies in S. The problem associated with this method is that there is no unique convex segmentation of S [Kurozumi, 1982] and for this reason the technique has limited application.

2.2 Boundary Methods

A family of shape recognition techniques use the object boundary in the recognition task. These are known as "boundary methods". Among these Polygonal approximation, Hough Transform and Fourier Descriptors are the three prominent methods and are described below;

2.2.1 Polygonal Approximations

Aggregation of edge elements in an image resulting in a piecewise linear approximation is termed "polygonal approximation". There are two major ways by which this is accomplished. The first of these is based upon defining a number of critical points (corner points) along the boundary. The second is based upon the use of successive approximations to iterate the best-fit polygonal representation. Both methods require the quantization of the edge image. The common method for this is the superimposition of a grid (usually squares) on the edge image [Freeman, 1961a, Freeman, 1961b]. The nearest grid point, to an intersection of the grid and the edge image, is taken as a point/node on the curve.

The line scan method [Freeman, 1977a, Freeman, 1977b] can be used to define critical points. It is based upon the computation of the discrete average slope at each point/node on the objects boundary. The slope is generally based upon a moving average involving 4 to 9 nodes. The number of nodes in the moving average is called the aperture size, w. The aperture size determines the degree of noise filtering. With reference to figure 2.1a, the incremental slope δ_k is defined as follows:

$$\delta_k = \theta_{k+1} - \theta_{k-1} \quad \text{--------------} \quad (2.2)$$

where
$$\theta_k = \begin{cases} \arctan\,(yy_k/xx_k) & \text{for } |xx_k| \geq |yy_k| \\ \operatorname{arccot}\,(xx_k/\,yy_k) & \text{for } |xx_k| \leq |yy_k| \end{cases}$$

and where
$$xx_k = \sum_{l=k-w}^{k-1} x_l$$

$$yy_k = \sum_{l=k-w}^{k-1} y_l$$

$$w = 5$$

xl = 1,0,-1 {being x component of chain link vector}

yl = 1, 0, -1 {being y component of chain link vector }

Figure 2.1b shows $\delta_k(w=5)$ vs node number for a chain coded curve having a 90° corner. It can be seen that δ_k is zero

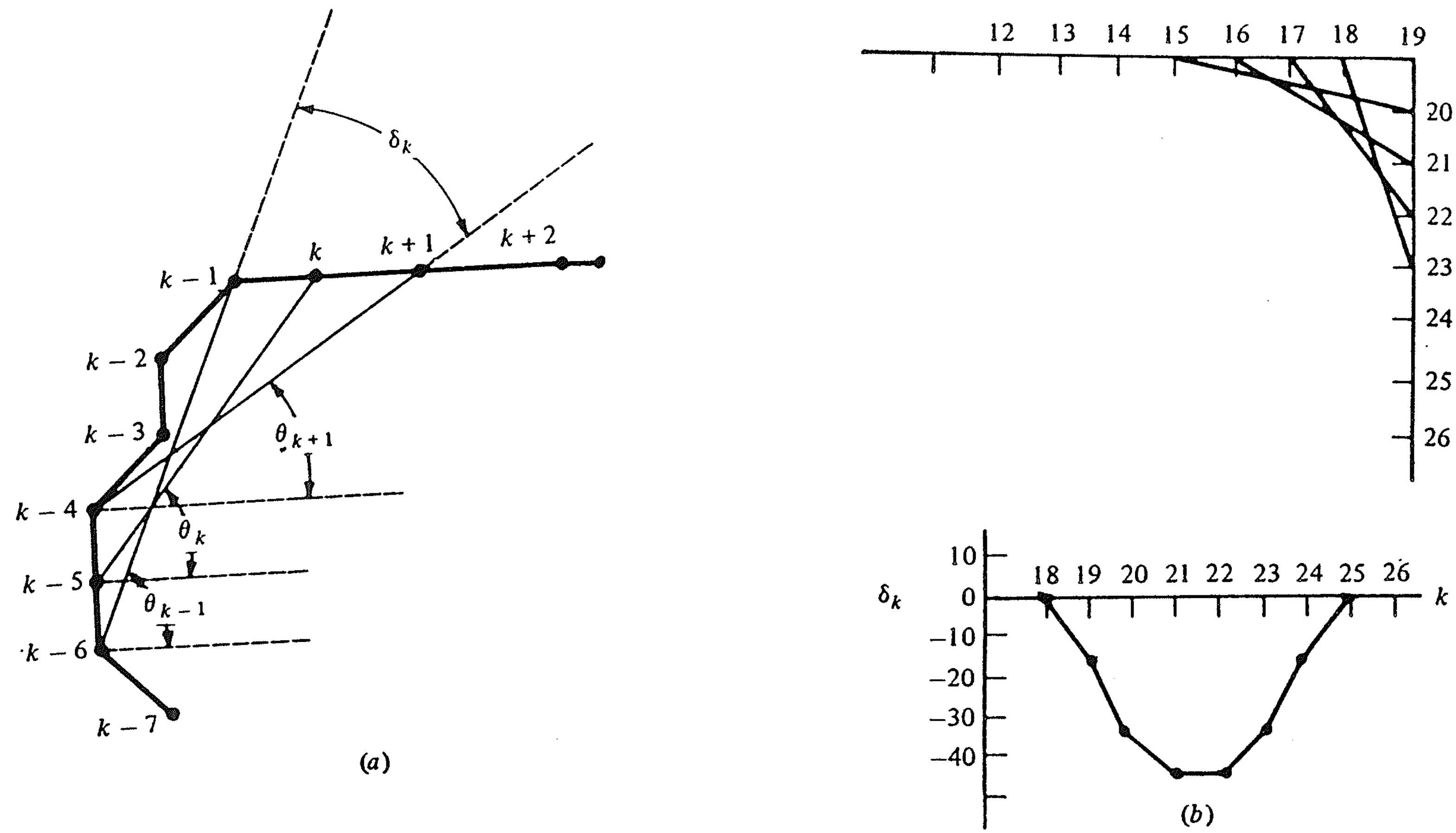

Figure 2.1: (a) Definition of θk and the Incremental Curvature δk at Node k, Given an Aperture Size w = 5 (b) The Plot of Incremental Curvature δk at Node k (w = 5) for a Chain Coded Curve Possessing a 90o Turn [Freeman, 1980].

for straight lines and passes through min or max turning points at corners. Table 2.1 shows some shape features that can be extracted from a plot of incremental curvature. For example a horizontal line in a plot of incremental curvature would indicate a shape of constant curvature, while a zero value would indicate a shape equivalent to a straight line.

There are a number of Iterative methods proposed for polygonal representation based on successive approximation. We shall review a typical Iterative technique proposed by Ramer (1972). The process is started by selecting two points on the closed curve. The initial point is joined to the final point by a straight line. The largest perpendicular distance of a boundary point to the straight line is computed. If this value is more than the pre-determined threshold then the curve is bisected. This process is continued in an Iterative fashion. Figure 2.2 demonstrates this process for a simple curve.

The advantages of Polygonal Approximation are (i) it can model any shape, (ii) it can successfully recognize partially occluded objects as some of the line segments would be fully visible. The disadvantage of this technique is that the matching of a large amount of line segments becomes a complex task.

δ_k vs. k plot	Shape interpretation
Horizontal line	Constant curvature
Large-magnitude value	High curvature
Positive value	Curvature toward left (bay)
Negative value	Curvature toward right (peninsula)
Zero value	Straight line
Zero-crossing	Point of inflection
Peak or valley of width $w + 1$ and sum value D	Curvature discontinuity $w/2$ units toward right of peak (valley) center and of angular change of $D/2$ degrees
Pairs of opposite-sign peaks of width 2 and magnitude arctan $1/w$, separated by $w - 2$ points of constant value	Straight line or gentle curve
Increasing (decreasing) mean slope	Inward (outward) spiral

Table 2.1: List of Features that can be Extracted from a Curve of Incremental Curvature [Freeman, 1980].

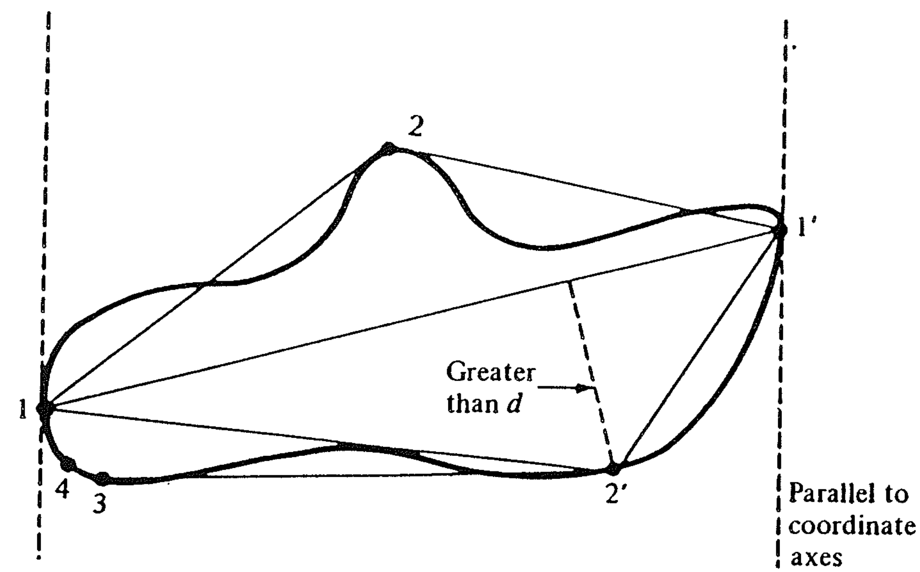

Figure 2.2: Iterative Technique for Generating a Polygonal Approximation to a Curve (Levine, 1985).

2.2.2 Hough Transform

Hough transform methods are based upon transforming the pixel data from space domain in image plane to a variety of different domains where the global characteristics are more easily recognized [Hough, 1962]. Hough transform is used extensively in biomedical applications.

To identify straight lines, Hough initially proposed a slope-intercept formulation. In its standard form this can be written as follows;

$$y = mx + c \qquad \text{-----} \ (2.3)$$

where x and y are coordinates of edge points in the space domain and "m" and "c" are the slope and intercept of a given line. Equation 2.3 could be rewritten as

$$m = y/x - c/x \qquad \text{-----} \ (2.4)$$

For a constant x and y (say x_o, y_o) a plot of "m" versus "c" can be made. This plot essentially shows all the straight lines that can pass through the point (x_o, y_o). To illustrate the complete procedure let us consider three potentially collinear points (x_1, y_1), (x_2, y_2), (x_3, y_3) shown in figure 2.3a. The objective of the exercise is to determine if the three points are collinear and if so what are the parameters of the straight line passing through these three points.

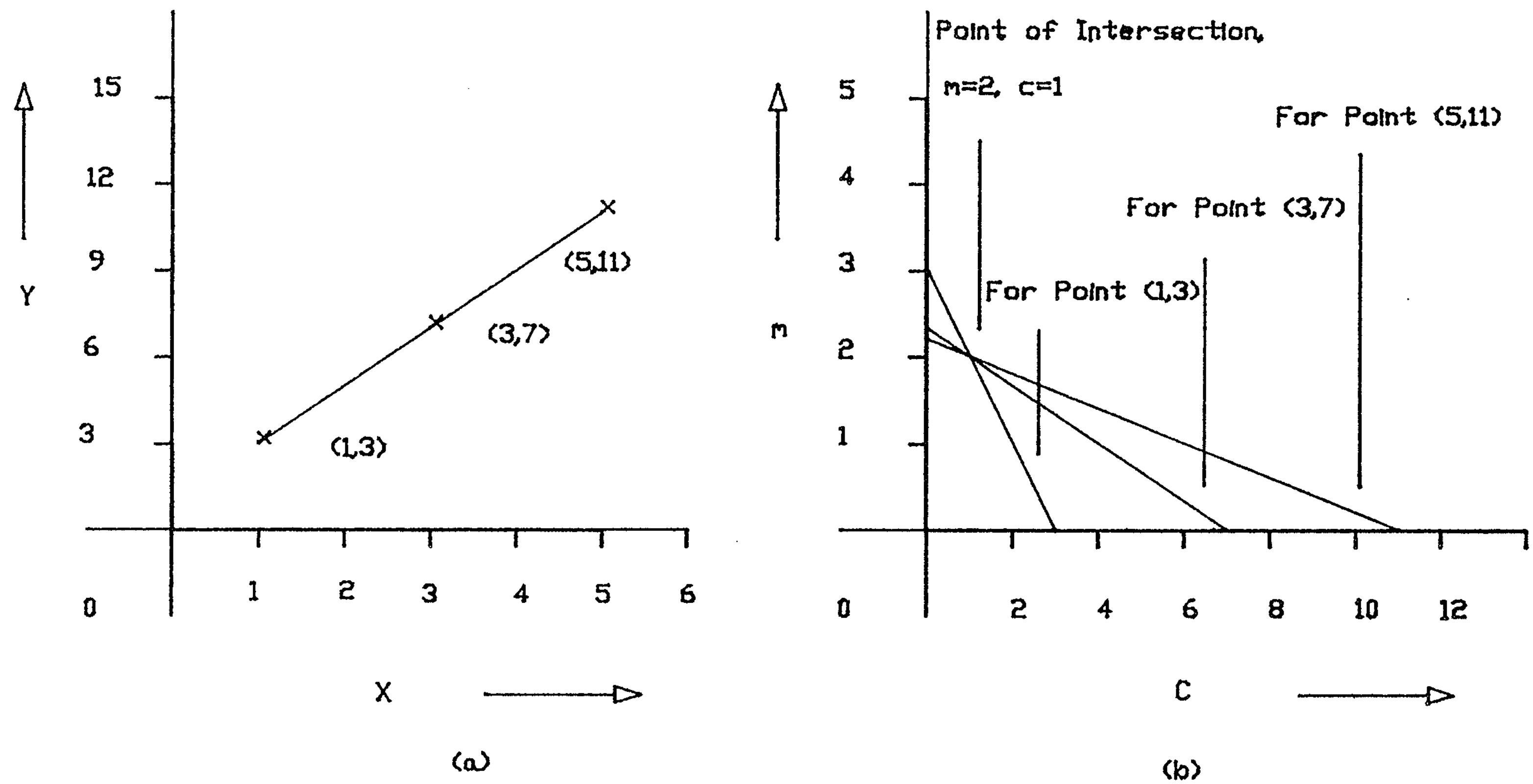

Figure 2.3: (a) Three Collinear Points on a Line $y=2x+1$, (b) Plot of $m = y/x - c/x$ for the Three Collinear Points in (a) Above

The procedure is started by plotting three curves of "m" versus "c" such that each curve corresponds to one of the three aforementioned points. Figure 2.3b shows the three curves corresponding to the three points. As can be seen in figure 2.3b, the three curves intersect at one point. This indicates that the three points are collinear. The values of "m" and "c" corresponding to the point of intersection are the slope and intercept of the straight line joining the aforementioned collinear points. Thus the problem of finding a line is reduced to a matter of detecting a cluster in the parameter (m,c) space.

In complex images with many lines, an n_m x n_c accumulator array is set up. Here n_m corresponds to the number of equally spaced intervals to be used in digitizing "m" and n_c the number of equally spaced intervals to be used in digitizing "c". Now for every edge point in the space domain, a series of points are computed in the (m,c) parameter space. Rather than plotting these points on a graph the corresponding cells in the accumulator array are incremented. A threshold value is now applied to identify accumulator cells with a high amount of activity. Each accumulator cell with a high amount of activity corresponds to a line in the space domain.

One of the complications with this type of parametrization is that the values of slope and intercept are unbounded. Duda and Hart (1972) suggested (θ, p) as alternative parameters as follows;

$$x \cos \theta + y \sin \theta = p \quad \text{------} \quad (2.5)$$

Where θ is the angle of a line passing through origin (0,0) and being normal to another line l' passing through an image point (x', y') and p being the length of the normal. The advantage of this parameterization is that θ and p are bounded as follows;

$$0^\circ \leq \theta \leq 360^\circ$$

$$\text{and} \qquad |p| \leq (\text{Image length}^2 + \text{Image Width}^2)^{1/2}$$

The advantages of Hough's Transform are (i) the technique can be extended to any analytical curve and has been successfully applied to the detection of circles, ellipses, parabolas and corners and (ii) the technique is relatively unaffected by noise and gaps in the image. The main disadvantage of this technique is that the computational complexity increases exponentially with the number of parameters required to define the curve. This limits the technique's application.

2.2.3 Fourier Descriptors

A number of different representations of Fourier Descriptors have been defined. Zahn and Roskies (1972) have defined an angle vs. length metric as follows;

> Assume Φ is a clockwise oriented closed curve (see figure 2.4) with parametric representation $(x(l), y(l)) = z(l)$ where l, the arc length, is $0 \leq l \leq L$. The angular direction of Φ at any point l is given by $\theta(l)$. The cumulative angular bend at any point l
>
> is given by $\phi(l)$ where $\phi(l) = \theta(l) - \theta(0)$. A normalized variant $\phi^*(t)$ is defined as follows:
>
> $$\phi^*(t) = \phi(Lt/2\pi) + t$$
>
> The domain of $\phi^*(t)$ is $[0,2\pi]$ such that $\phi^*(0) = \phi^*(2\pi) = 0$. Hence the Fourier Descriptors can be defined by the equation
>
> $$\phi^*(t) = \sum_{k=-\infty}^{\infty} c_k e^{jkt}$$
>
> $$c_k = \frac{1}{2\pi} \int_0^{2\pi} \phi^*(t) e^{-jkt}\, dt$$
>
> where the set $c_k = a_k - jb_k$ are the real and imaginary component of the descriptor coefficients.
>
> The magnitude of the coefficients of the above Fourier Descriptors are invariant to position, size and orientation. The phase angle is a function of the orientation. Information regarding the orientation of an object can be obtained from the phase angle.

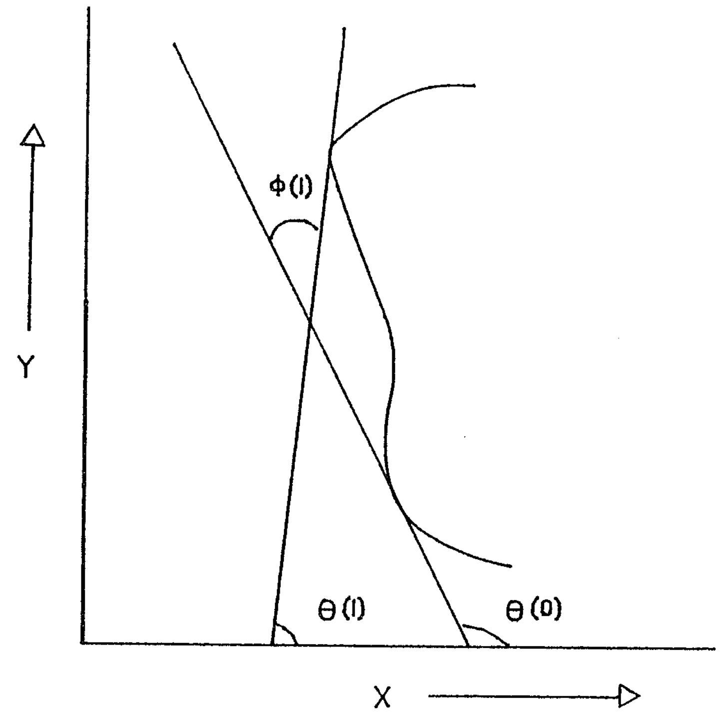

Figure 2.4: Parametric Representation of a Plane Curve With Tangential Direction Θ(l) and Cumulative Angular Bend Function Φ(l).

Granlund (1972) proposed alternative Fourier Descriptors based upon a complex parametric function as follows;

> Assume Φ is a clockwise oriented closed curve (see figure 2.4) with parametric representation (x(l), y(l)) = z(l) where l, the arc length, is such that $0 \le l \le L$. An imaginary point moving along the boundary generates a complex function u(l) = x(l) + y(l). This complex function is periodic with a period L. The Fourier Descriptors in this instance are described as follows;
>
> $$A_n = \frac{1}{L} \int_0^L u(l)\, e^{-j(2\pi/L)nl}\, dl$$
>
> $$u(l) = \sum_{-\infty}^{\infty} A_n\, e^{jn(2\pi/L)l}$$
>
> The magnitude of the coefficients of the above Fourier Descriptors are not invariant to size. They are however invariant to translation and orientation (note the direct current (DC) term is sensitive to translation). Information regarding the orientation of an object can be obtained from the phase angle data.

Another Fourier Descriptor [Rohlf, 1984, Kiryati, 1989] is based upon the length of a radius-vector from the centroid (also defined as the origin) to boundary points as a function of the angle the radius-vector makes with the horizontal axis (see figure 2.5). This descriptor is described as follows;

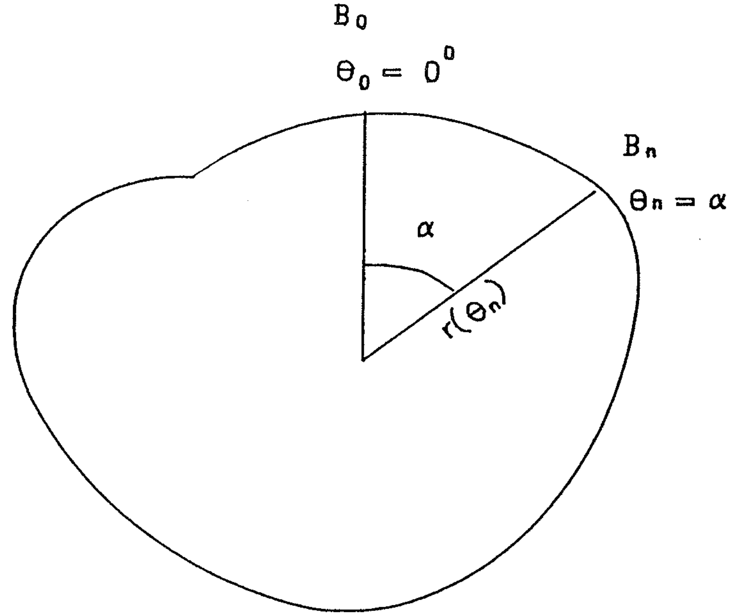

Figure 2.5: Parametric Representation of a Plane Curve with Radius–Vector as a Function of Angle.

$$r(\theta) = \sum_{n=-\infty}^{\infty} r_n e^{jn\theta}$$

$$r_n = \frac{1}{2\pi} \int_0^{2\pi} r(\theta)\, e^{-jn\theta}\, d\theta$$

The magnitude of the coefficients of the above Fourier Descriptors are not invariant to size. They are however invariant to position and orientation. Information regarding the orientation of an object can be obtained from the phase angle.

A comprehensive comparison of the relative merits of the three Fourier Descriptors is not available in literature. However some advantages and disadvantages of each are as follows;

1) In Fourier Descriptors based upon angle vs. length metric, $\phi^*(t)$ contains discontinuities for polygonal curves. As a result of this A_n decreases slowly as n increases [Person and Fu, 1977]. Fourier Descriptors based upon the complex parametric functions and those based upon the radius vector metric do not have this problem.

2) The magnitude of the coefficients of the Fourier Descriptors based upon the angle vs. length metric are size invariant. In contrast the magnitude of the coefficients of the other two types of Fourier Descriptors are not size invariant.

3) The magnitude of the coefficients of all three Fourier Descriptors are invariant to position and orientation. Information regarding orientation is resident in the phase angle data.

4) The Fourier Descriptors based upon angle vs. length metric are more sensitive to noise as compared to those based on the radius-vector vs. angle metric [Rohlf, 1984].

5) The radius-vector vs. angle technique is limited to applications where any radius-vector intersects the global boundary at only one point (see figure 2.6).

6) In all three instances, Fourier Descriptor coefficients of the partially occluded object model bear little structural similarity to the coefficients of the partially occluded object model.

7) In all three instances, only the global boundary of the shape is used in computing the shape descriptor. Structural information on the location of internal features is not incorporated in the shape descriptor. This is important as all available information is not being used.

2.2.4 Other boundary Methods

There are additional shape recognition methods based upon Contour Complexity, Shape Regularity and Global Geometric Shape. These are discussed below.

1) Contour Complexity

Contour complexity is a measure of the jaggedness or texture of the object boundary. A simple measure of contour complexity is the number of vertices in a polygon approximation of the boundary of an object. Another measure of contour complexity is angle regularity. Consider a closed polygon with n vertices and m boundary points. Then the angle regularity, A, is given as follows [Levine, 1985];

$$A_1 = \frac{1}{n} \left[(\theta_1 - \theta_m) + \sum_{k=1}^{m-1} (\theta_{k+1} - \theta_k) \right]$$

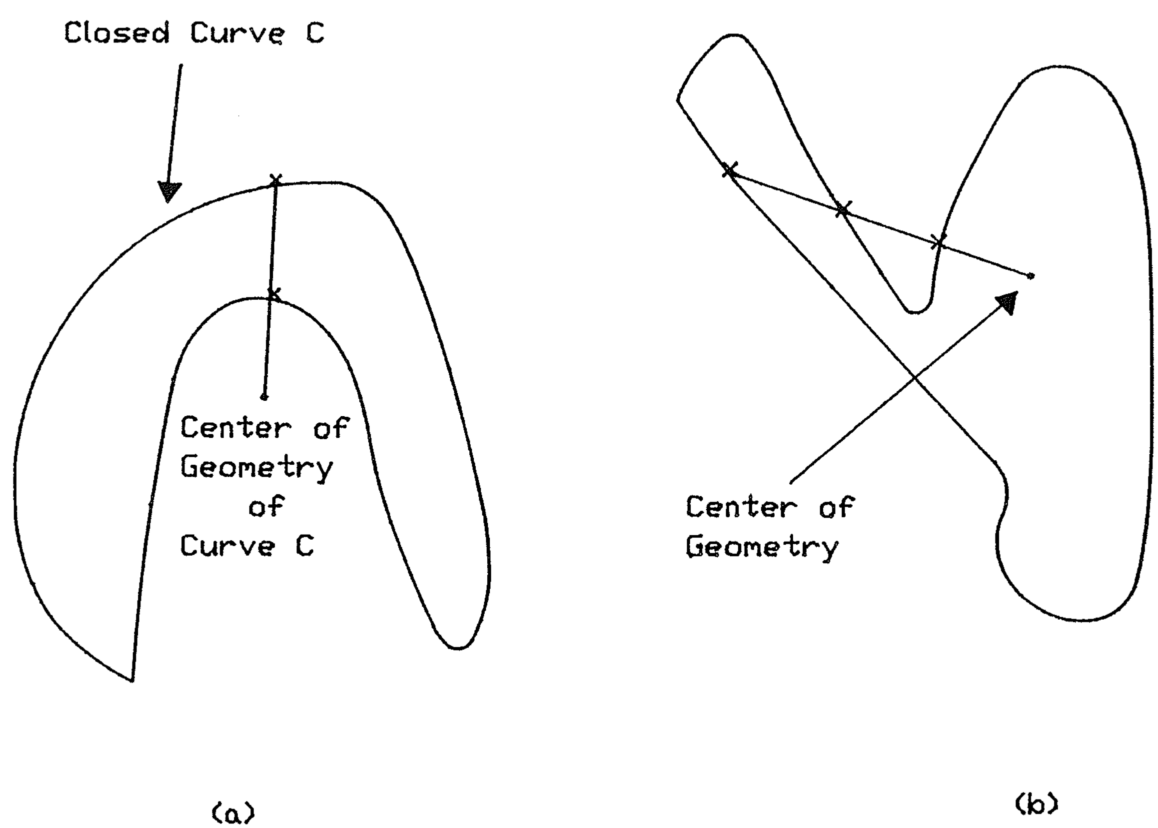

Figure 2.6: Limitations of the Radius-Vector vs. Angle Technique (a) Intersection of Radius-Vector With Two Boundary Points (b) Intersection of Radius-Vector with Three Boundary Points.

where θ_k = the interior angle at the k^{th} boundary point.

2) Shape Regularity

Shape regularity is based on the concept of uniformity of the lengths of the various segments of the polygonal curve. Shape regularity is determined by comparing the length of sides of an object with that of a regular polygon. As an example consider an object of n sides and of total perimeter length given by P. Then shape regularity, is defined as follows [Levive, 1985];

$$A_2 = \frac{\left[\sum_{k=1}^{n} (l_k - L)^2 \right]^{1/2}}{2L(n-2)}$$

where $L = P/n$
and where l_k = Length of the k^{th} segment of the polygonal curve.

3) Global Geometric Shape

Global geometric shape considers the global shape of the objects while ignoring boundary irregularities. There are a number of global geometric shape descriptors. These include the following [Levine, 1985];

$$\text{Compactness, } A_3 = \frac{P^2}{4\pi A}$$

where P = polygonal object perimeter
A = polygonal object area

$$\text{Average Bending Energy, } A_4 = \frac{1}{n} \sum_{k=0}^{m-1} \left| (R(k) \right|^2$$

where $R(k)$ = curvature at point k
P = perimeter of curve

$$\text{Elongation, } A_5 = \frac{|D-W|}{D}$$

where D = diameter of object(major axis)
w = width of object (minor axis)

Often a recognition task may deploy a number of descriptors based upon contour complexity, side regularity, and global geometric shape. An example of such a descriptor [Levine, 1985] may be as follows;

$$A_6 = 3\sqrt{\ } (A_1 \cdot A_2 \cdot A_4)$$

The advantage of the method presented in this section is that descriptors are simple and easy to compute and will provide sufficient discrimination for many simple applications. The major disadvantages of these descriptors are as follows;

(i) Structural information about the location of internal features is not used in the recognition process.

(2) Objects with similar global characteristics but with fine local differences could be wrongly classified.

(3) Descriptors of partially occluded objects will bear no resemblance to those of the fully visible object.

2.3 Special Axes Methods

Medial axis transform was first proposed by Blum (1972). This method is also known by other names such as Skeleton, Distance Transform, Stick Figures and Symmetric Transform [Levine, 1985]. The implementation of the process reduces two dimensional objects to a one pixel thick stick figure. There are a variety of methods available to implement this transform. Two of then are briefly discussed below;

1) The "Wave-Front" or the "Grass-Fire analogy" (see figure 2.7), approach considers each point on the boundary of the object as a source which propagates disk shaped wave fronts. Using the "grass-fire" analogy, the fire spreads radially until two or more wave fronts meet and the fire is extinguished. The fire extinction points are considered points on the skeleton.

2) Another approach is termed maximal disks [Blum, 1978]. In this approach disks of the largest size possible are fitted in the boundary curve. The locus of the center of these disks forms the skeleton (see figure 2.8).

Generally disadvantages of these approaches are as follows;

1) They simplify the object to an extent whereby significant information about the object is lost such that (a) reconstruction of the original shape is impossible and (b) probability of improper classification of similar objects increases.

2) By using these techniques a two dimensional object is reduced to a one dimensional stick figure. The one dimensional stick figure still has to be modelled in some analytical fashion for shape recognition.

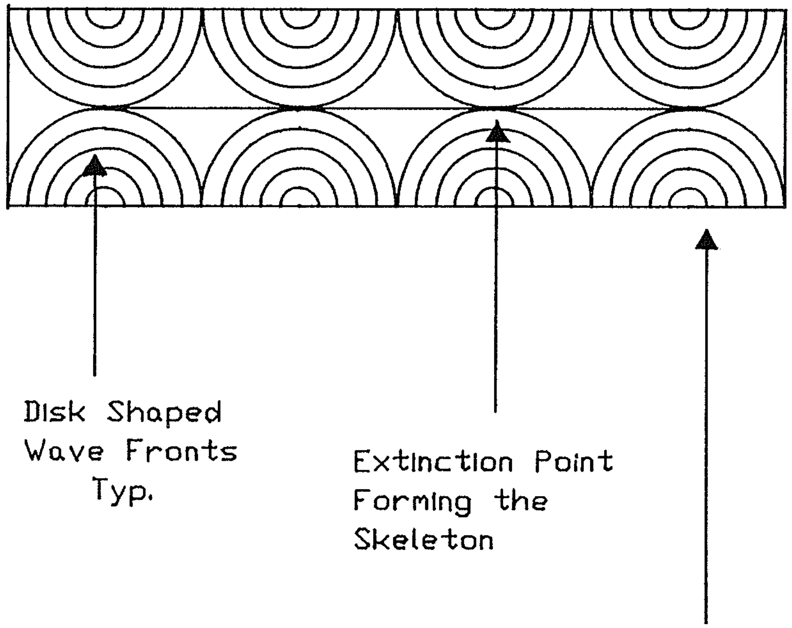

Figure 2.7: Creation of a Stick Figure Using 'Grass Fire' Analogy

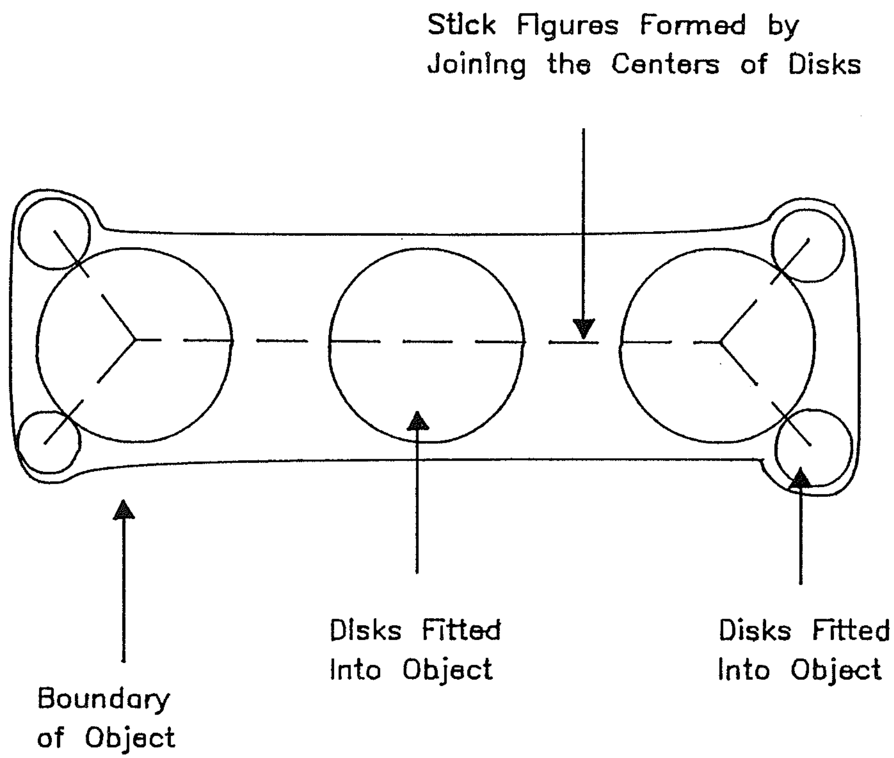

Figure 2.8: Creation of a Stick Figure Using "Maximal Disk" Technique.

2.4 Existing Systems for Complex Industrial Parts:

To date a number of approaches have been developed for the recognition of complex industrial parts. Some of these are discussed below:

2.4.1 Model Based System:

Perkins (1978) developed a model based vision system for the recognition of Industrial parts. A description of the system is as follows;

> A picture is digitized, and edge points are then extracted. The edge points are linked together to form "Chains". The chains are plotted in θ-S space where S is the distance of an edge point along the chain and θ the change in angular direction. Straight lines in real space become horizontal lines in θ-S space, while arcs become straight lines with slopes proportional to the arc's curvature. The objective of plotting data in θ-S space is to separate the edge points into groups to be fitted by different curves. This is achieved by computing the curvature (Dθ/DS) at each edge point. An abrupt change in curvature indicates a breakpoint and a new grouping of edge points. The chains are converted into concurves by fitting (using least squares) straight lines and arc to grouping of edge points.
>
> System operation consists of two modes. The first involves developing a model in "training" mode for updating the database. The second involves conducting the recognition exercise in "runtime" mode. The process of matching a runtime image to items in the database consists of three steps as follows:
>
> (1) Comparison of General Features: In this step the concurves in the model are compared to the concurves in the runtime image. This is done by comparing the following properties:

a) The general description of the concurve (straight line, circle, some complex shape).

b) The number and length of arcs.

c) The number and length of straight lines.

d) Bending energy.

e) Area of any internal holes.

f) For closed concurves, minimum and maximum moments of inertia, ratio of area/perimeter^2.

Based on the above comparison, the concurves from the model are matched with concurves in the runtime image. This is done by computing the likelihood of matching each concurve in the model against each concurve in the runtime image. For example, if there are five concurves each in the model and the runtime images, then the likelihood of the resulting 25 combinations would be computed. Combinations with the highest likelihood would be selected for the tentative transformation .

(2) Forming Tentative Transformations: The next step in the recognition process is to translate and rotate a concurve in the runtime image to duplicate the corresponding model concurve. For this purpose, the first combination (with the highest likelihood) of model and runtime image concurves are taken. If the model concurve symmetry is 1 then the model and runtime image concurves are aligned by cross correlation in θ-S space. If the concurve has rotational symmetry greater than one then the next combination (second highest permutation) is taken and the centers of the concurves aligned. The program is designed to reject transformations that produce poor correlation in θ-S space or unacceptable spacing between concurve centers.

(3) Checking Tentative Transformations: Once a transformation is complete and the image concurves are remapped, there is the final step of verifying the validity of the

transformation. This consists of creating equally spaced perpendicular multisectors on the model concurves. These multisectors are superimposed on the runtime image concurves. A good transformation will result in the runtime image concurves bisecting the multisectors at right angles.

Some disadvantages/limitations of Perkins's Approach are as follows;

(1) The matching of a large amount of straight line segments becomes a complex task,

(2 If the boundary concurve in a runtime image is partially occluded then all the global geometric properties of the concurve will be different from that of the model concurve. Then matching would only be made on the basis of concurves of internal local features. This is a serious limitation as a large amount of information on the object boundary is not used in the matching process.

(3) In the matching process, the system does not take into account the location (structural information) of internal features in relation to the global object boundary.

2.4.2 Local Feature Focus Method:

Bolles and Cain (1982) introduce an approach called the "Local Feature Focus Method" for the recognition of partially occluded objects. The analysis is based upon the recognition of local features such as corners and holes in the object. The technique is based upon forming clusters of local features that lie near each other in the "training" mode. Then in run time mode (i) the most important feature is identified and a cluster developed by adding neighboring features and (ii) a graph is formed in which the nodes represent image-feature to model-feature matching pairs, and edges represent pairwise consistent assignments between nodes. Matching is then performed using maximal clique(s).

The major disadvantage of this approach is that it assumes the presence of sharp local features such as corners. Their approach would not function if the object boundary was in the form of French curves ie. large continuous arcs.

2.4.3 Boundary Matching Using Footprints:

Kalvin et al. (1986) proposed a two-dimensional, model based, approach in which boundary segments are matched using footprints. Their approach can be summarized as follows:

Given a composite scene in which a number of objects overlap, the first task is to find breakpoints along the complex outer boundary that delineate the boundaries of the individual objects. This is accomplished by finding points on the boundary that exhibit sharp concavity.

The delineated object boundary (ie. edge points between two adjacent breakpoints) is plotted in the form of an arclength vs. turning angle graph (in a manner similar to Perkins, 1978). The graph is discretized into a sequence U_j, where j

= 1 to n and where n is the number of equally spaced intervals along the x-axis. For each element, U_j, in the sequence, a footprint is computed. The footprint is obtained by computing the first four Fourier coefficients of a discrete sequence starting at U_j and of length ws (the window size). In order to emphasize sharp features, the total turning angle at U_j is added as a fifth dimension to the footprint at U_j.

In the "training" mode, in order to store model data, a five dimensional space, divided into hypercubes, is defined. Associated with each hypercube, is a list of object models whose footprints pass through the hypercube. Then in runtime mode, given a set of footprints, F, of a partially occluded object, those models that are repeatedly associated with hypercubes that contain sections of F are retrieved. The retrieved models are termed candidate models. The candidate models are then individually matched to the occluded object.

The advantages of this method are;

(1) The generation of candidate models results in an algorithm that degrades in a sub-linear fashion with increase in database size.

(2) There are no constraints (except for concavity, discussed below) on the shape of the object boundary.

(3) The system can accommodate partially occluded objects.

The disadvantages/limitations of this approach are as follows;

(1) In order to accommodate partially occluded objects, the Fourier transform (of discrete points in a window) is computed at every discrete point on the boundary curve. This exhaustive (brute force) method generates a lot of data that has to be stored and later utilized for matching in the "runtime" mode. To achieve this, for a database containing 100 parts (i) a database of 32×10^6 hypercubes is created [Kalvin et al., 1986] to hold the footprints, (ii) only 0.1% of the hypercubes did not hold a model number while 99.9% of the cubes held one or more model numbers. Based upon the above, the size of the database is estimated to be greater than 100 megabytes ([32×10^6 hypercubes] x [average of three bytes per hypercube]).

(2) The method of finding breakpoints by identifying sharp concavity fails in the following situations; (a) if two objects overlap without forming points of sharp concavity at points of intersection and (b) if an object has a sharp concavity in its boundary contour.

(3) If the visible boundary is less than the window size then no candidate models will be selected,

(4) The recognition process does not take into account the location (structural information) of internal features in relation to the global object boundary.

2.5 Summary

Method of Moments and Hough transform are computationally intensive. Stick Transform methods and methods based upon boundary complexity, regularity, and global geometric shape, simplify the object to an extent whereby significant information about the object is lost. As a result reconstruction of the original shape is impossible. Furthermore the probability of improper classification of similar objects increases. Polygonal Approximation is used extensively in the recognition of industrial objects. In this transform, the object boundary is divided into many straight line segments and this enables the technique to recognize partially visible objects. The disadvantage of this technique is that the matching of a large amount of straight line segments becomes a complex task. This disadvantage is also evident in the approach described by Perkins (1978).

Fourier Descriptors are used extensively in the recognition of objects as they can model any shape and only the first 10 - 15 coefficients (depending on shape complexity, [Rohlf, 1984]) of the Fourier Descriptor have to be matched for recognition. The traditional disadvantage of Fourier Descriptors based techniques is that they cannot recognize partially visible objects. The Fourier Descriptors of a partially visible boundary has no structural resemblance to that of the fully visible model boundary. Kalvin et. al. (1986) in their work on recognizing partially visible objects, got around this limitation by computing the Fourier transform (of discrete points in a window) at every discrete point on an arclength vs. turning angle graph. Although this "brute force" method was successful in recognizing partially visible objects, it did so at the cost of making model database extremely large.

In all approaches except the one described by Bolles and Cain (1982), the information on the location of internal features with respect to the boundary is not used in object recognition. In many recognition applications there may not be internal local features such as holes and slots. However, in many industrial applications, for example in aerospace manufacturing, the location of internal features is significant and can be utilized in the recognition process.

Based upon a general survey of literature, it appears that Fourier Descriptors would be very useful for the recognition of complex industrial objects if the recognition of partially occluded objects is accommodated. In the next chapter we describe an approach using extended features which will enable the recognition of partially occluded objects using Fourier Descriptors. The proposed technique is also unique because it makes use of structural information on the location of internal features.

CHAPTER 3

RECOGNITION USING FOURIER DESCRIPTORS OF EXTENDED FEATURES:

The technique for recognizing partially occluded complex industrial objects is described in three sections. The first section describes the proposed Fourier Descriptors and demonstrates some of its properties. The second section describes a technique for recognizing complex industrial objects. In the third section, extension of the basic recognition technique for overlapping and touching objects is discussed.

3.1 Proposed Fourier Descriptors

The radius-vector vs. angle representation of Fourier Descriptors [Rohlf, 1984, Kiryati, 1989] described in section 2.2.3 requires the centroid to be also the origin. This condition is easy to satisfy if there is only one closed curve in the scene. When dealing with complex objects (with multiple internal features), there will be multiple closed curves in the scene. Thus, to use this representation of Fourier Descriptors for the recognition of complex objects, the requirement that the centroid of a connected closed curve is also the origin, should be dropped. The proposed Fourier Descriptors are similar to the aforementioned radius-vector vs. angle representation except for this requirement.

The proposed Fourier Descriptors are described as follows;

> A simple connected closed planar curve, B, shown in figure 3.1a, can be represented parametrically as a function of $r(\theta_n)$. θ_n represents the angular arc length measured from B_o, where $\theta_o = 0$, to some point Bn where $\theta_n = \theta_o + \delta n$. $\delta = 360/k$ where k is the number of terms in the discrete time domain sequence. $r(\theta_n)$ is the absolute distance between the object's focus and a point B_n on the curve C. The focus is defined as (i) the center of geometry when dealing with internal features or (ii) an artificial point (to be described later) when dealing with segments of the global boundary called extended features.

Using this metric, we can compute and plot the characteristic curve of $r(\theta)$ vs. θ for any closed planar curve where the centroid of an internal feature lies within the closed curve. Based upon this metric, $r(\theta)$ would be periodic and will have no discontinuities. The characteristic plot can then be sampled to produce a discrete sequence. The discrete sequence can be transformed to the frequency domain by computing the Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) to produce Fourier Descriptors.

If the boundary curve is highly concave, as shown in figure 3.1b, then the centroid will lie outside the boundary curve.

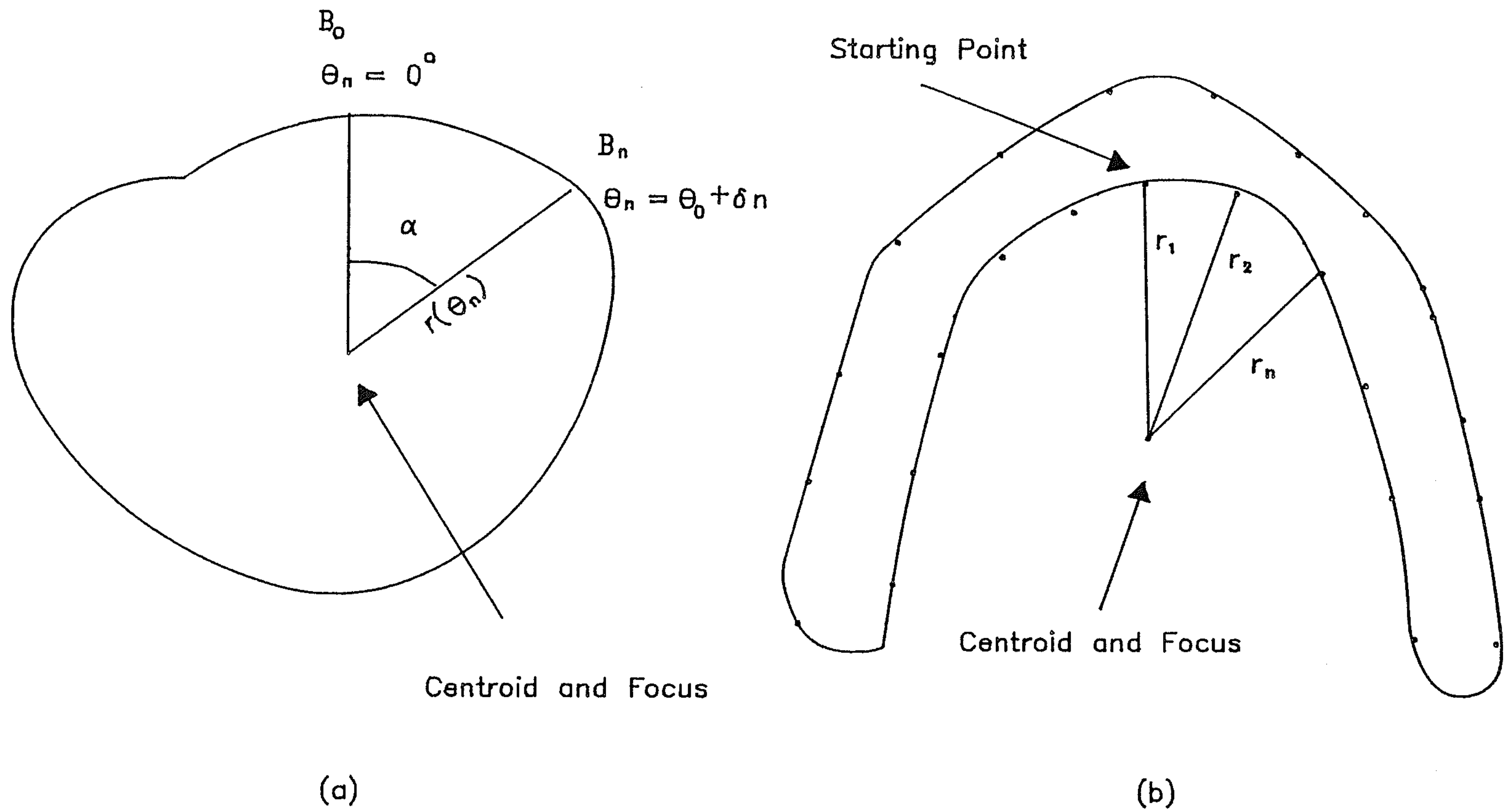

Figure 3.1: (a) Representation of a Closed Curve Where the Centroid Lies Within the Curve (b) Representation of a Highly Concave Curve Where the Centroid Lies Outside the Curve.

In this instance, Fourier Descriptors are generated as follows;

(i) A point on the object boundary, closest to the centroid is defined as the first point on the curve. An additional k-1 point, equally spaced along the object boundary are now generated. The distance from the focus to each one of these points is now computed to produce a discrete time domain sequence with K terms.

It is rare to find objects with internal features that are highly concave. Thus this representation will have limited application. The representation was described for the sake of completeness and will not be discussed further.

As an illustration, we shall compute the FD of a 4" x 8" rectangle shown in figure 3.2. First we divide the global object boundary into 16 segments, each subtending a 22.5 degrees arc at the focus (which in this case is the geometric center). The number of segments has been chosen arbitrably to illustrate the principle. The values for $r(\theta)$ vs. θ for the rectangle are tabulated in table 3.1. An analog plot of $r(\theta)$ vs. θ is shown in figure 3.3. The analog plot is sampled (the 16 points used to draw Fig. 3.3 are used instead) and converted to Fourier domain by an FFT algorithm [Press, 1988] to produce FD shown in table 3.2. A 16 point FFT requires 16 elements in the time domains as input and produces a descriptor consisting of 16 parameters in the frequency domain. Each parameter has a real component

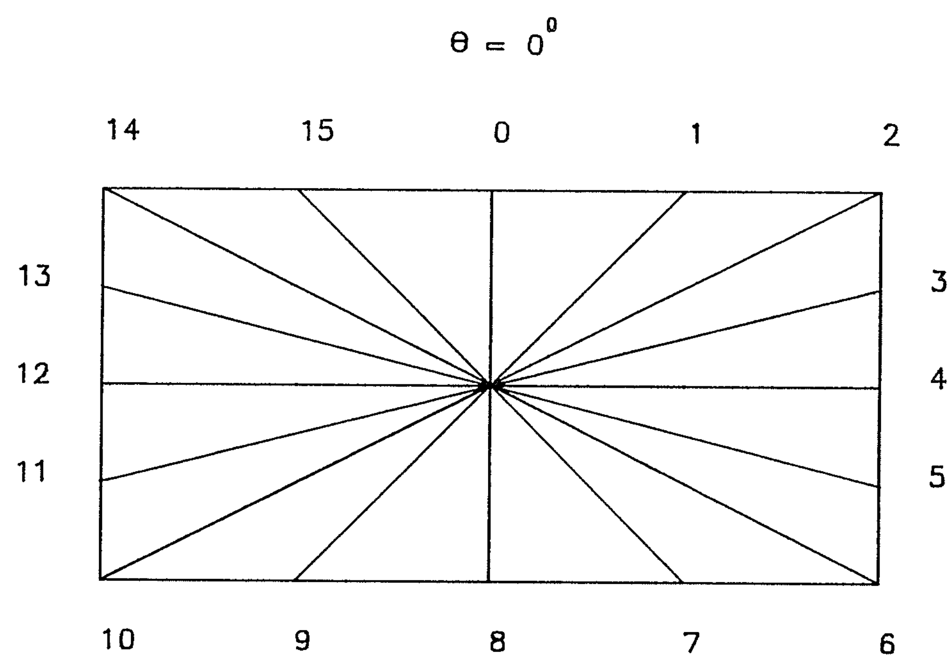

Figure 3.2: 4" x 8" Rectangle

Segment #	θ (degrees)	r(θ) (inches)
0	0.0	2.000
1	22.5	2.165
2	45.0	4.472
3	67.5	4.330
4	90.0	4.000
5	112.5	4.330
6	135.0	4.472
7	157.5	2.165
8	180.0	2.000
9	202.5	2.165
10	225.0	4.472
11	247.5	4.330
12	270.0	4.000
13	292.5	4.330
14	315.0	4.472
15	337.5	2.165
Σ		55.868

Table 3.1: r(θ) vs. θ for 4" x 8" Rectangle in Figure 3.2

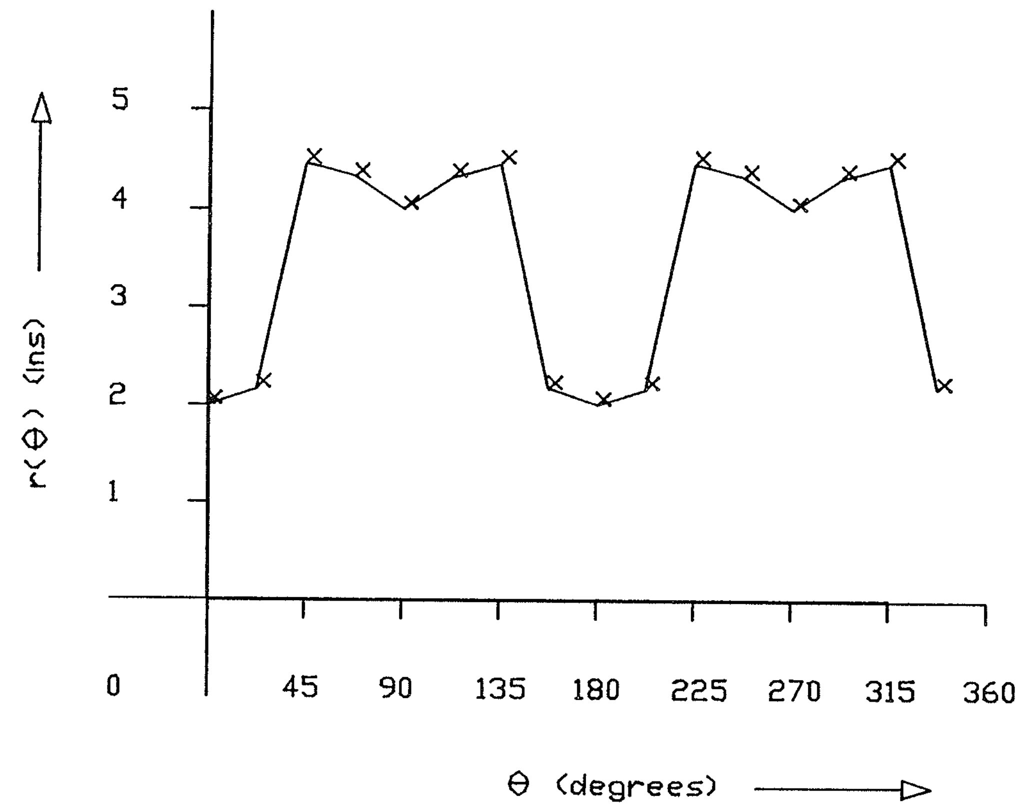

Figure 3.3: A Plot of r(Θ) vs. Θ
for a 4′ × 8′ Rectangle in Figure 3.2

Terms	Real Comp.	Imag. Comp.	Mag.	Phase
DC	55.86800	0.00000	55.86800	0.00
1	-0.00000	-0.00000	0.00000	0.00
2	-10.12354	-0.00000	10.12354	180.00
3	-0.00000	-0.00000	0.00000	0.00
4	-5.88800	0.00000	5.88800	180.00
5	0.00000	-0.00000	0.00000	0.00
6	2.12354	0.00000	2.12354	0.00
7	-0.00000	0.00000	0.00000	0.00

Table 3.2: Fourier Descripitors of 4" x 8" Rectangle in Figure 3.2

and an imaginary component. The magnitude and phase angle of each parameter are also computed and are shown in table 3.2. The first half of the 16 parameters of the Fourier Descriptor are a mirror image of the second half [Ramirez, 1985]. Thus, table 3.2 only shows first eight coefficients of the descriptor. The first term is called the Direct Current (DC) term and normally represents the average value of r(θ). In our case, as a function of the FFT algorithm [Press, 1988] used, all the coefficients of the FD are scaled by a factor of 16 (16, being the number elements in the input time domain sequence). To verify this, note that the magnitude of the first term (DC term) in table 3.2 is 55.868. This is exactly equal to the summation of all r(θ) values in table 3.1.

For a rectangle, it can be seen from table 3.2, that the magnitude of the second, fourth and sixth coefficients are high while the others are zero. This we will see later is characteristic of rectangular shapes. We shall now see the characteristics of some other shapes.

3.1.1 Properties of FD With Respect to Shape

Figure 3.4 shows a 4" x 4" square. The values for r(θ) vs. θ for the square are tabulated in table 3.3. An analog plot of r(θ) vs. θ is shown in figure 3.5. The FD of the square are

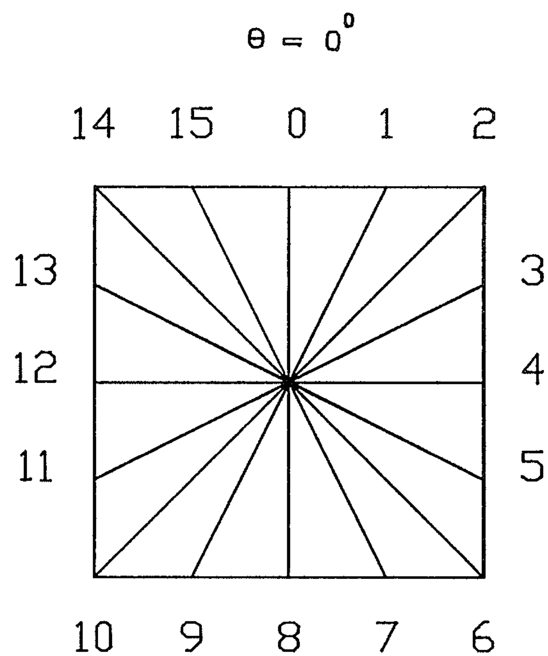

Figure 3.4: 4′ Square

Segment #	θ (degrees)	r(θ) (inches)
0	0.0	2.000
1	22.5	2.165
2	45.0	2.828
3	67.5	2.165
4	90.0	2.000
5	112.5	2.165
6	135.0	2.828
7	157.5	2.165
8	180.0	2.000
9	202.5	2.165
10	225.0	2.828
11	247.5	2.165
12	270.0	2.000
13	292.5	2.165
14	315.0	2.828
15	337.5	2.165

Table 3.3: r(θ) vs. θ for 4" Square in Figure 3.4

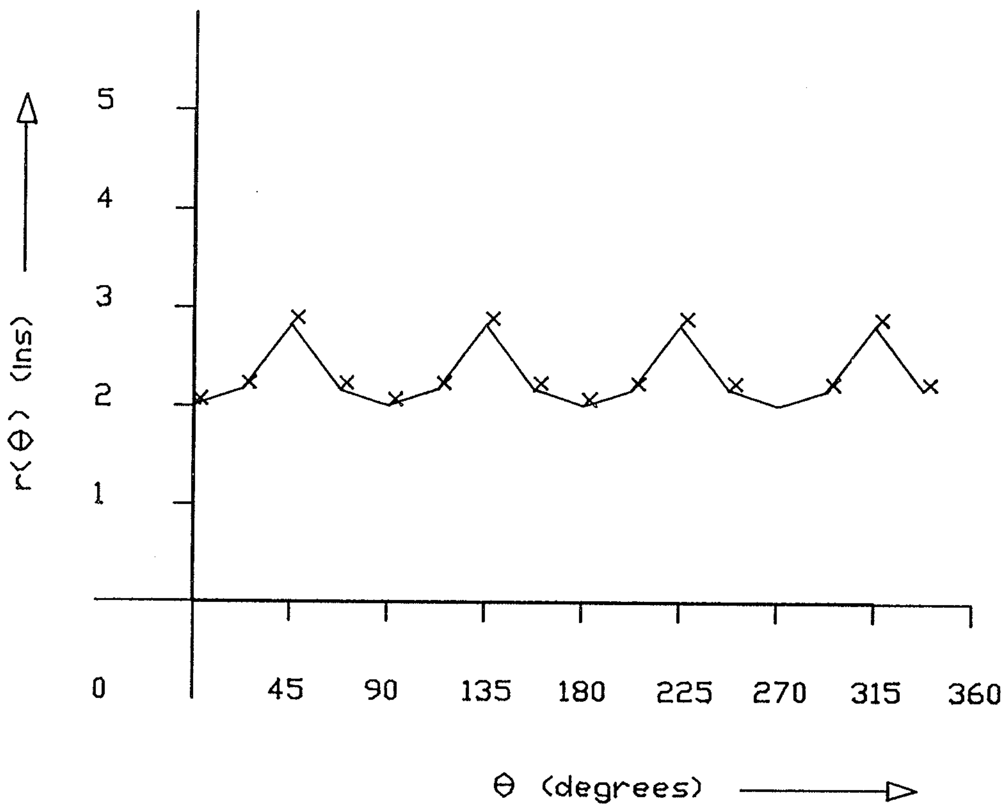

Figure 3.5: A Plot of r(Θ) vs. Θ for a 4" Square in Figure 3.4

shown in table 3.4. It can be seen that besides the DC term, the magnitude of the fourth coefficient is high. This is characteristic of squares.

Now, lets consider a circle shown in figure 3.6. The values of $r(\theta)$ for various θ are all equal to the radius of the circle. This is evident in figure 3.7 where $r(\theta)$ is plotted against θ. The FD of the circle are shown in table 3.5. From table 3.5 it can be seen that all the parameters (except the DC term) are zero. Hence, a circle in the Fourier domain can be called a shapeless feature.

In summary, the comparison of the first eight coefficients of the Fourier Descriptors' of a rectangle, square and a circle lead to the following observations;

- A circle has a shapeless boundary and as such the magnitude of all coefficients are zero (except the DC term).
- In the case of the rectangle, the second and the fourth coefficients are high.
- In the case of a square, the fourth coefficient is high.

Of the shapes examined so far, the proposed Fourier Descriptors have a unique value for each shape. We will see some more examples of this in chapter 4. The ability of the proposed $r(\theta)$ representation of Fourier Descriptors to

Term	Real Comp.	Imag. Comp.	Mag.	Phase
DC	36.6000	0.0000	36.6000	0.00
1	-0.0000	-0.0000	0.0000	0.00
2	-0.0000	-0.0000	0.0000	0.00
3	-0.0000	-0.0000	0.0000	0.00
4	-3.3000	0.0000	3.3000	180.00
5	0.0000	-0.0000	0.0000	0.00
6	-0.0000	0.0000	0.0000	0.00
7	-0.0000	0.0000	0.0000	0.00

Table 3.4: Fourier Descriptors of 4" Square in Figure 3.4

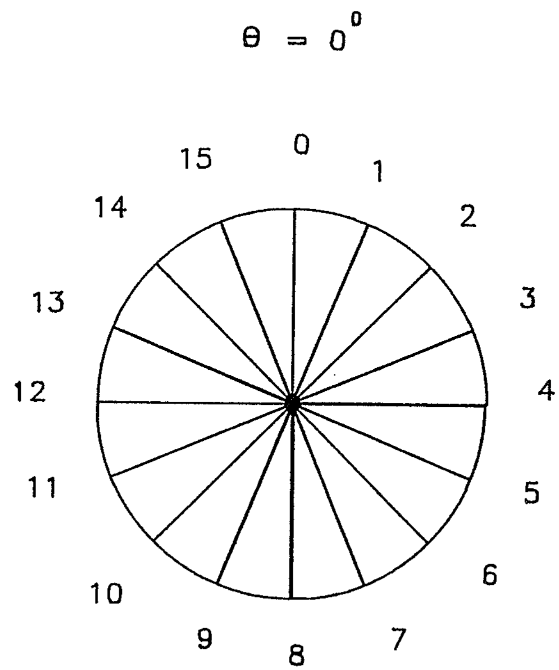

Figure 3.6: 4" Diameter Circle

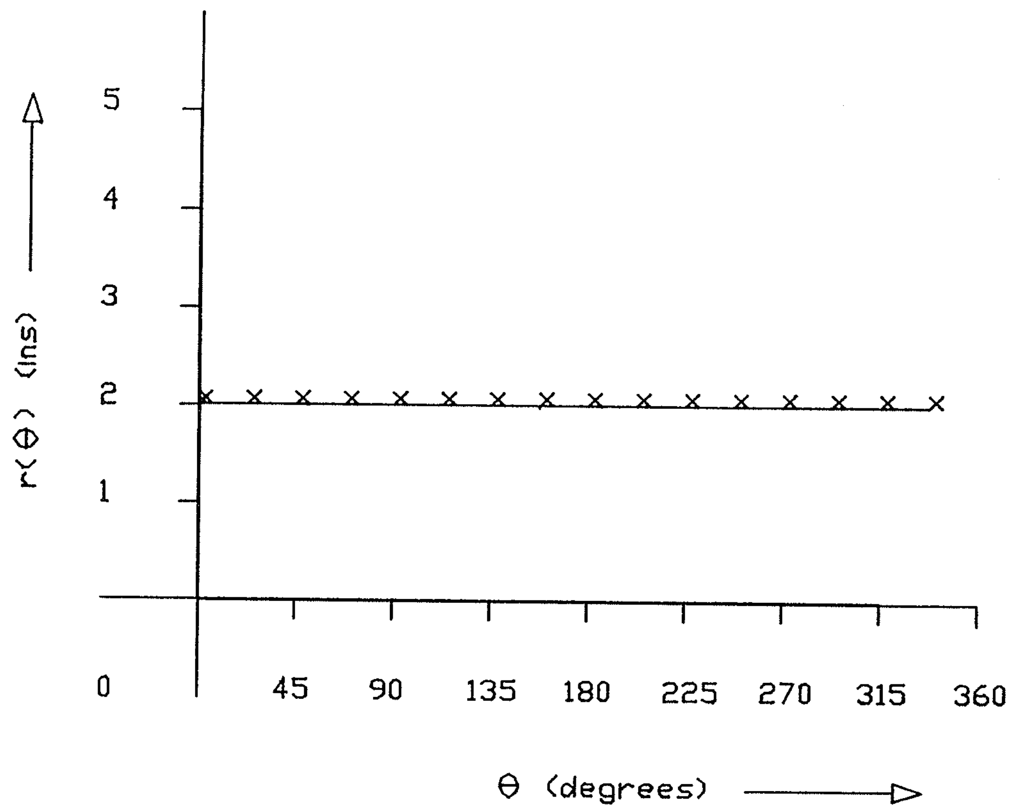

Figure 3.7: A Plot of r(Θ) vs. Θ
for a 4" Diameter Circle in Figure 3.6

Term	Real Comp.	Imag. Comp.	Mag.	Phase
DC	32.0000	0.0000	32.0000	0.00
1	-0.0000	-0.0000	0.0000	0.00
2	-0.0000	-0.0000	0.0000	0.00
3	-0.0000	-0.0000	0.0000	0.00
4	-0.0000	0.0000	0.0000	0.00
5	0.0000	-0.0000	0.0000	0.00
6	-0.0000	0.0000	0.0000	0.00
7	-0.0000	0.0000	0.0000	0.00

Table 3.5: Fourier Descripitors of 4" Circle in Figure 3.6

exhibit a unique descriptor for each shape is similar to that of the radius-vector vs. angle representation described in literature [Rohlf, 1984, Kiryati, 1989].

3.1.2 Properties of FD With Respect to Rotation

The proposed Fourier Descriptors' magnitude is rotation invariant while the phase angle is sensitive to rotation. To verify the sensitivity of the phase angle to rotation, consider the first shifting theorem [Weaver, 1983] which states as follows;

> If the discrete Fourier Transform of a N^{th} order sequence $\{f(k)\}$ is $\{F(j)\}$, then the discrete Fourier Transform of the shifted sequence $\{f(k-n)\}$, $n \in [0,N-1]$, is given by $\{F(j)W_N^{-jn}\}$.

As an example, we will compare the results obtained by (i) taking the unshifted 4" x 8" rectangle in figure 3.2 and mathematically manipulating its Fourier Descriptors to obtain the descriptors of a shifted rectangle, and (ii) physically shifting the rectangle (as shown in figure 3.8) and then computing its Fourier Descriptor.

If the 4" x 8" rectangle in figure 3.2 is rotated by 22.5 degrees then ;

N = 16 {total number of terms}

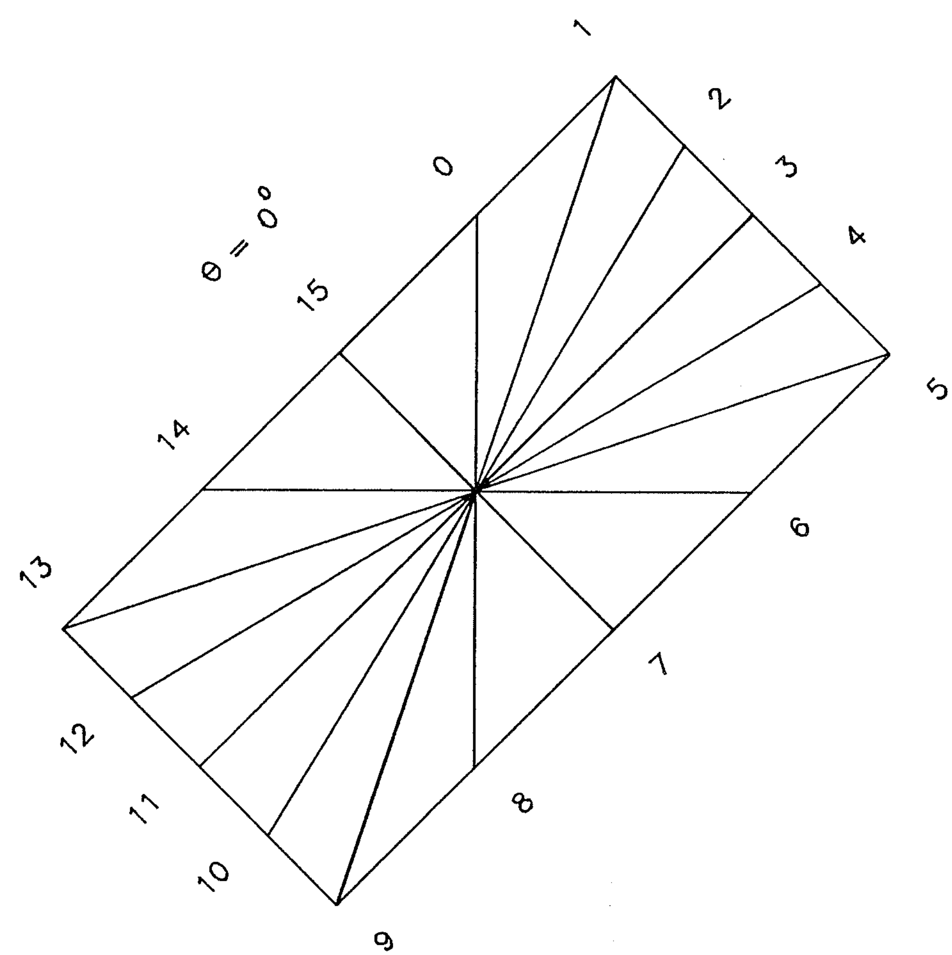

Figure 3.8: 4" x 8" Rectangle of
Figure 3.2 Rotated by 22.5 Degrees

n = 1 {the number of terms by which the sequence is shifted} ; and

$$\tilde{F}[\{f(k-1)\}] = \{F(j)W_N^{-jn}\}$$

$$= \{F(j)W_{16}^{-j}$$

$$= F(j) \cdot e-2\pi ij/16$$

$$= F(j) \cdot e-\pi ij/8$$

$$= F(j) \{ \cos \frac{\pi j}{8} - i \cos \frac{\pi j}{8} \} \text{ --- } 3.1$$

Now from table 3.2,

F(J) = (55.868,0), (0,0), (-10.1,0), (-5.88,0), (0,0) (2.123,0), (0,0) ---------------------------- 3.2

Substituting equation 3.2 in equation 3.1 we get the following;

$\tilde{F}$ [{f(k-1)}] = (55.808,0), (0,0), (-7.071, +i7.071), (0,0), (0, +i5.888), (0,0), (-1.5, -i1.5), (0,0) ------------------ 3.3

To verify the above, we physically shift figure 3.2 to produce figure 3.8. The shifted time domain sequence is shown in table 3.6. The shifted time domain sequence is converted to the Fourier domain and the first 8 coefficients of the descriptor are shown in table 3.7. It can be seen

Segment #	θ (degrees)	r(θ) (inches)
0	0.0	2.165
1	22.5	4.472
2	45.0	4.330
3	67.5	4.000
4	90.0	4.330
5	112.5	4.472
6	135.0	2.165
7	157.5	2.000
8	180.0	2.165
9	202.5	4.472
10	225.0	4.330
11	247.5	4.000
12	270.0	4.330
13	292.5	4.472
14	315.0	2.165
15	337.5	2.000

Table 3.6: r(θ) vs. θ for 4" x 8" Rotated Rectangle in Figure 3.8

Term	Real Comp.	Imag. Comp.	Mag.	Phase
dc	55.9000	0.0000	55.9000	0.00
1	-0.0000	-0.0000	0.0000	0.00
2	-7.2000	7.2000	10.1000	135.00
3	-0.0000	-0.0000	0.0000	0.00
4	-0.0000	5.9000	5.9000	90.00
5	0.0000	-0.0000	0.0000	0.00
6	-1.5000	-1.5000	2.1000	225.00
7	-0.0000	0.0000	0.0000	0.00

Table 3.7: Fourier Descripitors of 4" x 8" Rotated Rectangle in Figure 3.8

that the real and imaginary components of the coefficients in equation 3.3 are approximately equal to the corresponding real and imaginary components in table 3.7.

From the above, the following conclusions can be derived;

(a) The magnitude of the Fourier Descriptors is rotation invariant.

(b) The information pertaining to orientation resides in the phase angle and can be used to determine an objects orientation.

3.1.3 Properties of FD With Respect to Scale

Figure 3.9 shows a 2" x 4" rectangle. The values for $r(\theta)$ vs. θ are the same as in table 3.1 (for 4" x 8" rectangle) except that they are exactly half in magnitude. The analog plot of $r(\theta)$ vs. θ is also similar to figure 3.3 except for the $r(\theta)$ values being half in magnitude. The FD of the 2" x 4" rectangle are shown in table 3.8. Table 3.9 shows a comparison between the magnitude and phase angle of FD of 4" x 8" rectangle (shown in table 3.2) and a 2" x 4" rectangle (shown in table 3.8). It can be seen that the magnitude of the FD of the 4" x 8" rectangle are twice as large as that of the 2" x 4" rectangle. Thus the proposed Fourier Descriptors are not scale invariant although there is a linear relationship between magnitude of sequences in the time and Fourier domains. We will see later that this very property will enable the proposed descriptor to encompass

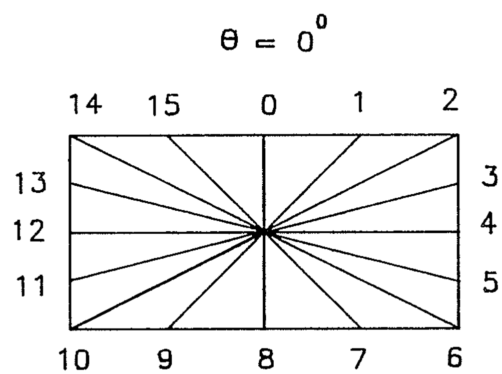

Figure 3.9: 2" x 4" Rectangle

Term	Real Comp.	Imag. Comp.	Mag.	Phase
DC	27.9000	0.0000	27.9000	0.00
1	-0.0000	-0.0000	0.0000	0.00
2	-5.1000	-0.0000	5.1000	180.00
3	-0.0000	-0.0000	0.0000	0.00
4	-2.9000	0.0000	2.9000	180.00
5	0.0000	-0.0000	0.0000	0.00
6	1.1000	0.0000	1.1000	0.00
7	-0.0000	0.0000	0.0000	0.00

Table 3.8: Fourier Descripitors of 2" x 4" Rectangle in Figure 3.9

	4 x 8 Rectangle		2 X 4 Rectangle	
Term	Mag.	Phase	Mag.	Phase
DC	55.868	00.00	27.9000	00.00
1	0.0000	0.00	0.0000	0.00
2	10.1235	180.00	5.1000	180.00
3	0.0000	0.00	0.0000	0.00
4	5.8880	180.00	2.9000	180.00
5	0.0000	0.00	0.0000	0.00
6	2.1235	0.00	1.1000	0.00
7	0.0000	0.00	0.0000	0.00

Table 3.9: Comparison of Fourier Descripitors of 4" x 8" and 2" x 4" Rectangles in Figures 3.2 and 3.9 respectively.

structural information (on the location of internal local features) in the recognition of complex industrial objects.

3.1.4 Properties of FD With Respect to Image Reconstruction.

The proposed Fourier Descriptors have the ability to reconstruct a curve in time domain. This is demonstrated by way of an example. Consider the 4" x 8" rectangle in figure 3.2 for which figure 3.3 shows the plot of $r(\theta)$ vs. θ. Table 3.2 shows the corresponding FD. To reconstruct the $r(\theta)$ vs. θ plot from the FD in table 3.2, we take the FFT of the Fourier coefficients in Table 3.2. The resulting FD are shown in table 3.10. Table 3.11 shows the computation of $r(\theta)$ from the FD in table 3.10. Figure 3.10 shows the reconstructed plot of $r(\theta)$ vs. θ. By comparing the original and the reconstructed curve it is evident that the two curves are identical except for minor amounts of degradation. The degradation is due to noise introduced in the computation

3.1.5 Summary of Proposed Fourier Descriptors

In this section we have proposed a slightly modified representation of an existing Fourier Descriptor. The following has been demonstrated;

(1) Every shape has unique Fourier Descriptors.

Term	Real Comp.	Imag. Comp.	Mag.	Phase
DC	28.09200	0.00000	28.09200	0.00
1	38.54801	-0.00000	38.54801	0.00
2	67.64400	-0.00000	67.64400	0.00
3	73.18799	0.00000	73.18799	0.00
4	60.09200	0.00000	60.09200	0.00
5	73.18799	0.00000	73.18799	0.00
6	67.64400	0.00000	67.64400	0.00
7	38.54801	-0.00000	38.54801	0.00

Table 3.10: Fourier Descriptors of Fourier Descripitors of 4"x 8" Rectangle in Table 3.2

Term	Fourier Coeff. (Real)	r(θ) = Fourier Coeff/N
DC	28.09200	1.756
	38.54801	2.409
	67.64400	4.228
	73.18799	4.574
	60.09200	3.756
	73.18799	4.574
	67.64400	4.228
	38.54801	2.409
Nyquist Freq.	38.54801	2.409
	38.54801	2.409
	67.64400	4.228
	73.18799	4.574
	60.09200	3.756
	73.18799	4.574
	67.64400	4.228
	38.54801	2.409

Table 3.11: Negative and Positive Frequency Terms Reconstructed from Table 3.10

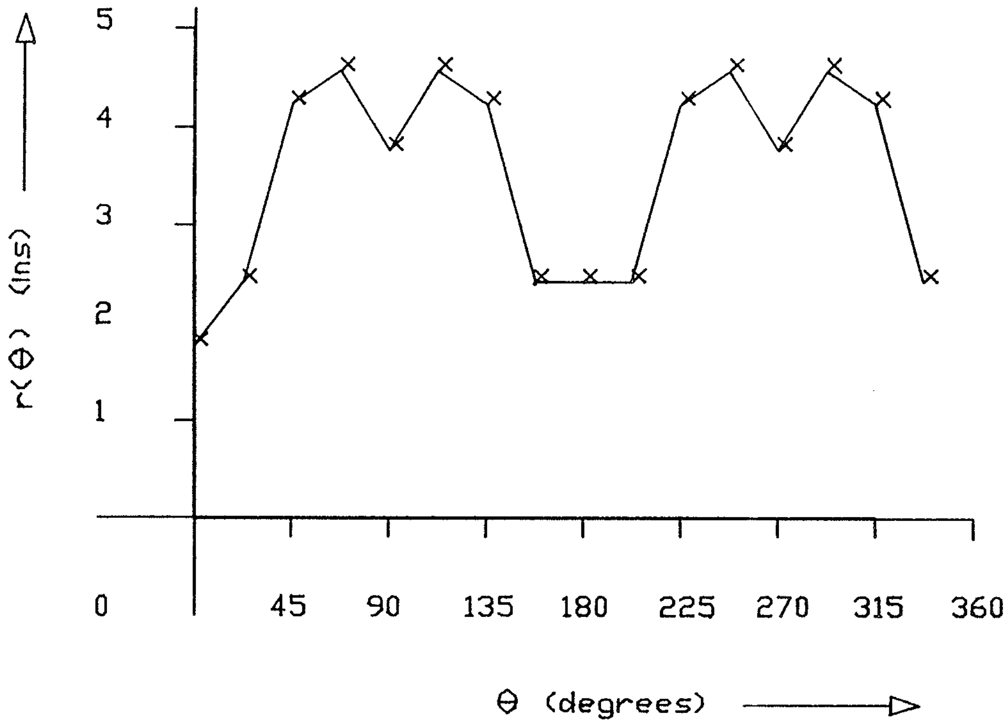

Figure 3.10: Reconstructed (from Fourier Descriptors), a Plot of $r(\Theta)$ vs. Θ for a 4′ × 8′ Rectangle in Figure 3.2

(2) The Fourier Descriptors are not scale invariant, but the magnitude of the coefficients are linearly proportional to the magnitude of the input time domain sequence.

(3) The magnitude of the coefficients of the Fourier Descriptors are rotation invariant. The phase angle of the coefficients holds information about the orientation of the object.

(4) The original time domain image can be reconstructed from the Fourier domain by computing the DFT of the Fourier domain sequence. Being able to reconstruct an object boundary indicates that no information about the boundary is lost in creating the Fourier Descriptors.

3.2 Recognition of Complex Industrial Objects.

In the proposed system, the recognition process consists of two modes. The first one is called "training mode" and the second is called "runtime mode". In the "training mode" models of objects are created and stored in a database file. In "runtime mode", models of objects are created and matched with models in the database. The process of creation of models is identical in the two modes.

A model principally consists of Fourier Descriptors of two types of features. The first of these are internal features such as holes, slots, etc.. The second type are extended features. Extended features are artificially created segments of the object's global boundary.

For purposes of demonstration let us consider a complex industrial object shown in figure 3.11. The object consists of four internal features (circle, square, large rectangle and a small rectangle).

To generate Fourier Descriptors of the internal features, the boundary of each of the four internal features is processed (as described in section 3.1), with the centroid being defined as the focus. This produces four Fourier Descriptors, each of which uniquely corresponds to a shape and size of one of the internal features.

To generate Fourier Descriptors of extended features, the global boundary of an object is divided into a number of segments by a heuristic (to be described later in sections 3.2.1 and 3.2.2). Each segment is called an extended feature. Specifically with reference to figure 3.12, it can be seen that there are four extended features corresponding to boundary segments between points P_1 to P_2, P_2 to P_3, P_3 to P_4 and P_4 to P_1. Associated with each extended feature is an artificial point called a focus. A discrete sequence $r(\theta)$ is computed where $r(\theta)$ is the distance from the focus to the boundary points. The time domain sequence $r(\theta)$ is converted to the Fourier domain using an FFT algorithm. The resulting Fourier Descriptors uniquely define the extended feature's

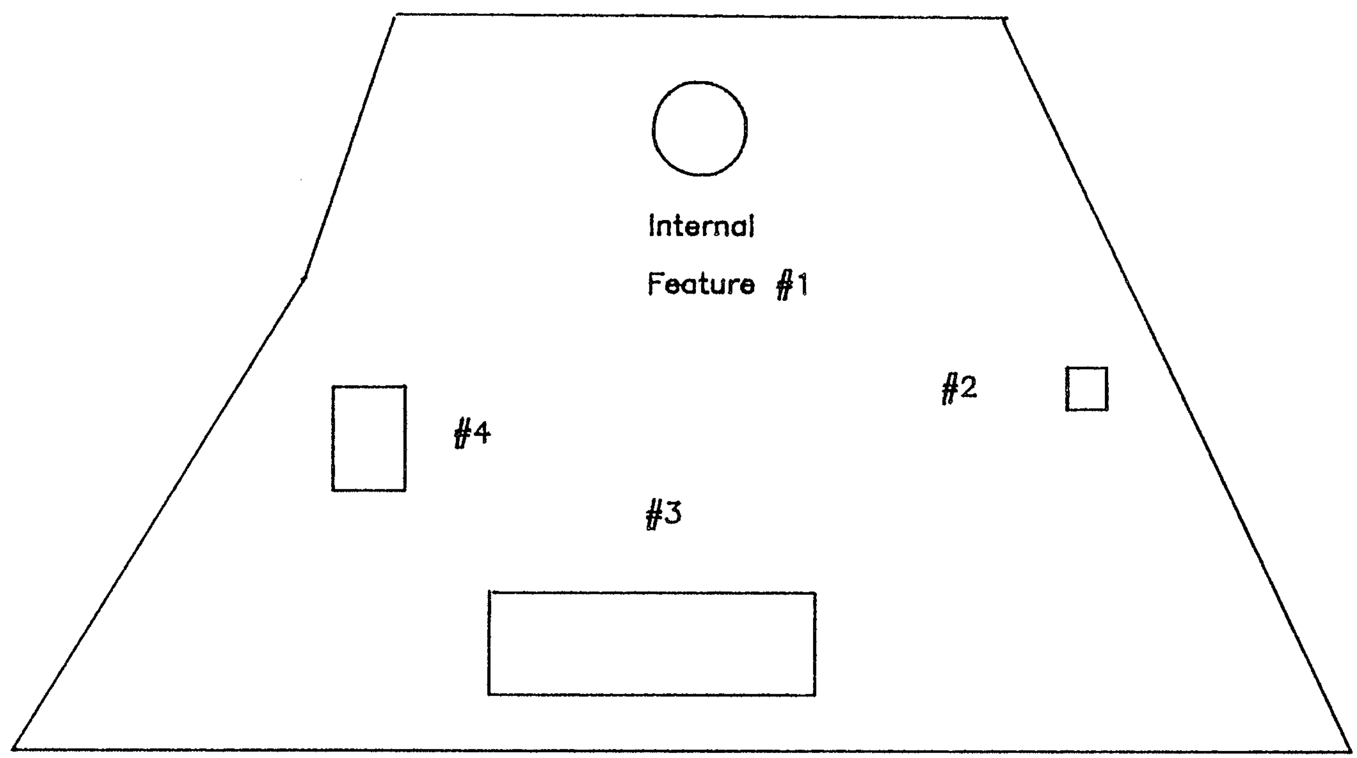

Figure 3.11: Complex Industrial Object With Four Internal Features

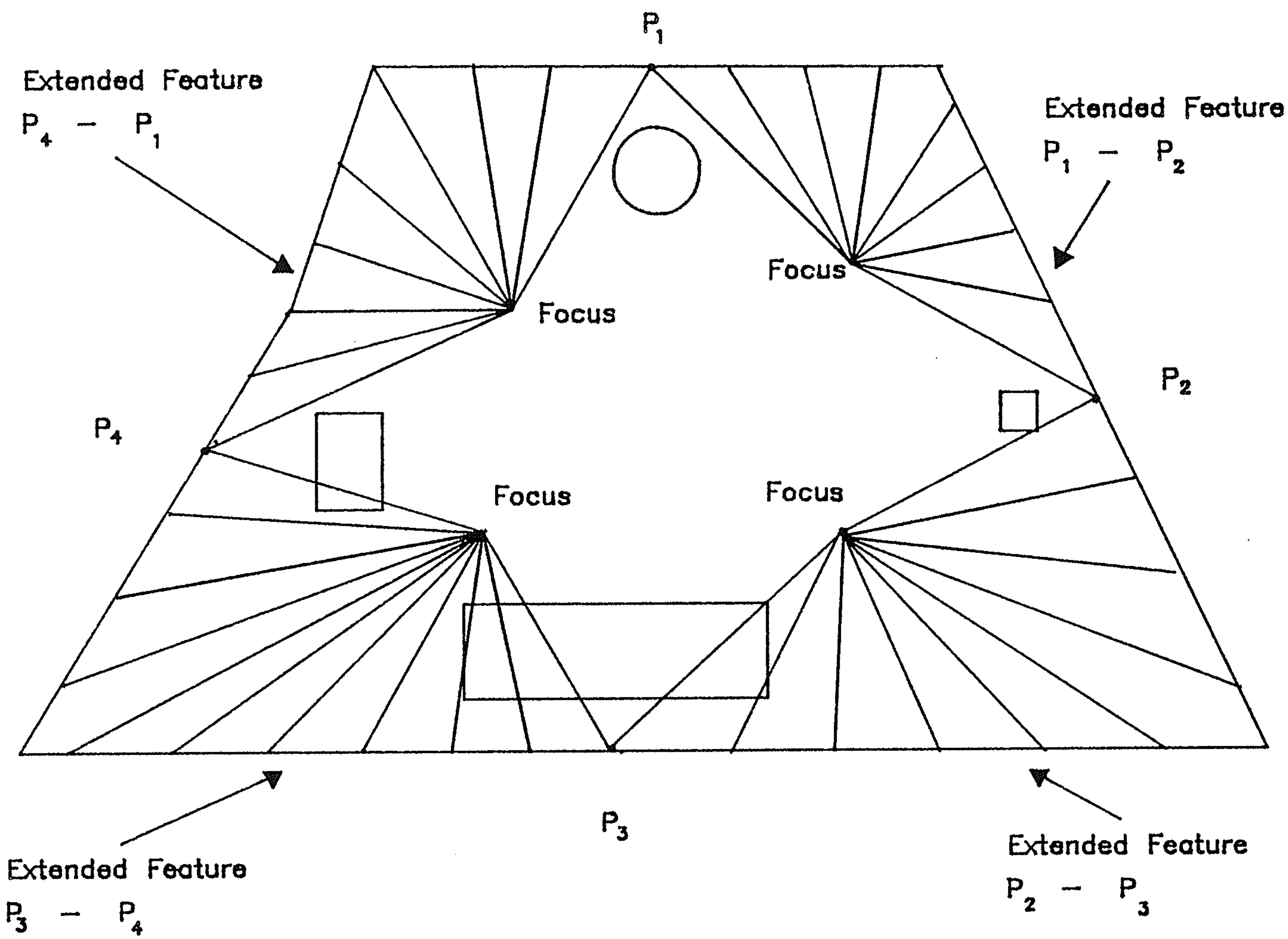

Figure 3.12: Location of the Focus and Extended Features for the Object In Figure 3.11

shape and its location with respect to neighboring internal features.

Referring to figure 3.12, in the "training mode", a total of eight Fourier Descriptors will be generated (four from internal features and four from extended features) and stored in a database under the object's serial number. In "runtime mode", when an object is to be recognized, a model of the "runtime" object is created. If the object is the same as the one shown in figure 3.12 and is not partially occluded then a total of eight Fourier Descriptors will again be generated. If the object is the same but partially occluded then (i) a few of the internal features and extended features may be missing and (ii) a few new extended features would be created. These new extended features will not match with any Fourier Descriptors in the model of the candidate object. The degree of occlusion will determine the number of features that are successfully matched. A system could be configured to declare an object recognition task successfully completed when a pre-determined number of internal and extended features are successfully matched (details of matching are discussed in section 5.2.6).

Two heuristics are proposed for the generation of extended features and their Fourier Descriptors. These are as follows;

(i) One for objects with two or more internal features.

(ii) Another for objects with less than two internal features.

The heuristic for objects with two or more internal features (complex objects) is the primary focus of this thesis. In this heuristic, a minimum of two internal features are required to generate an extended features. For completeness, a companion heuristic is being suggested to accommodate objects with less than two internal features. These heuristics are as follows;

3.2.1 Heuristic #1, For The Generation of Extended Features and Fourier Descriptors For Objects With Two or More Internal Features

For purposes of demonstration we again consider the complex rigid object, with four local features, shown in figure 3.12. The extended features and foci are generated as follows;

a) The geometric center for each internal feature is determined. This is computed as follows;

   Calculate the center of gravity of the internal feature where each pixel inside the internal feature is assumed to have a mass of unity and each pixel outside the feature a mass of zero.

b) Find the shortest cord between the geometric center of each internal feature and the global object boundary. The point, where this shortest cord meets the global boundary, is called a critical point. With reference to figure 3.12, it can be seen that there are four internal features and four critical points, P_1, P_2, P_3 and P_4.

c) Global boundary segments between adjacent critical points (going clockwise around the boundary) are called extended features. Again with reference to the object in figure 3.12, it can be seen that there are four critical points resulting in four extended features as follows;

P_1 to P_2
P_2 to P_3
P_3 to P_4
P_4 to P_1

c) To demonstrate the computation of the coordinates of the focus of extended features we consider extended feature P_1-P_2 (see figure 3.12). The (x,y) coordinates of the focus are computed as follows;

$$x = (xf_1 + xf_2)/2$$

$$y = (yf_1 + yf_2)/2$$

where

(xf_1, yf_1) are the x and y coordinates of the geometric center of the internal feature that determined the first critical point, P_1.

Similarly (xf_2, yf_2) are the x and y coordinates of the center of geometry of the internal feature that determined the second critical point, P_2.

Having described the generation of extended features and the focus, we describe the generation of Fourier Descriptors as follows;

(1) A discrete sequence, $r(\theta)$, the distance from the focus to the extended feature's boundary points is computed. The values of θ are limited to those between θ_1 and θ_2 where (i) θ_1 is the angle formed by the line which joins the focus and the first critical point and the horizontal axis and (ii) θ_2 is angle formed by the line which joins

the focus and the second critical point and the horizontal axis.

(2) An extended feature is an open curve. It is desirable that $r(\theta)$ be periodic and have no discontinuities. To achieve this, an extended segment between two critical points is traced backwards in an anti-clockwise direction. This forms a closed curve. To simulate this, the discrete $r(\theta)$ sequence is also repeated backwards.

(3) The discrete sequence of the closed curve is now converted to the Fourier domain by a FFT algorithm previously used. In the current implementation the time domain sequence has 256 discrete terms.

So far we have described the method for generating Fourier Descriptors of the object shown in figure 3.12. By referring to two additional objects, we now describe some additional features of the heuristic which make the technique potentially widely applicable.

Let us consider the object shown in figure 3.13. This object has three internal features. These internal features are circular and are labeled C_1, C_2 and C_3. Both C_1 and C_3 have a large amount of points on the global boundary that are all equidistant and the shortest distance from the internal features' geometric center. This leads to a large amount of critical points on the global boundary. This scenario is handled as follows;

For a set of critical points, delete all critical points that have neighboring pixels, on both sides, that are critical points. using this method, in figure 3.13, the number of critical points are greatly reduced. All critical points between P_1-P_6 and P_3-P_4 are eliminated.

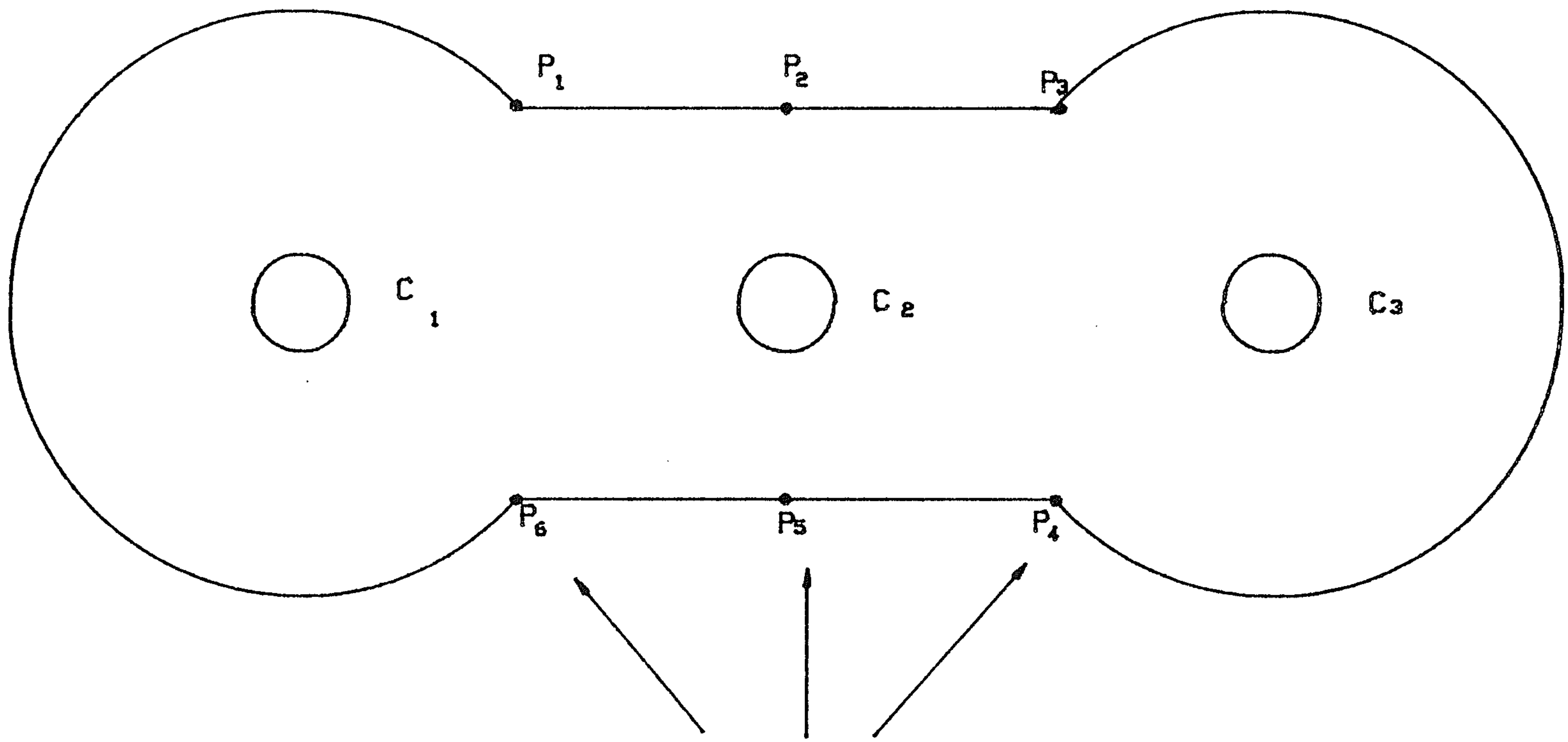

Figure 3.13: Complex Object With Internal Features Having More Than One Critical Point

In figure 3.13, internal feature C_2 will give rise to critical points, P_2 and P_5, if both of these points are equidistant from the geometric center of C_2. Two points are considered equidistant if the difference of their distances is less than a threshold value.

Thus, if the P_2 and P_5 are indeed equidistant from the geometric center of C_2, then Figure 3.13 will have a total of six critical points and thus six extended features. The foci and Fourier Descriptors of the six extended features would be computed in a manner similar that which was previously described in section 3.1.

Let us now consider the object shown in figure 3.14. This object has six relatively large internal features in addition to 24 small holes (pilot holes for riveting). In this case extended features are generated as follows;

(1) Usually pilot holes are in large numbers. It is not desirable to use them to generate extended features for the following reasons;

   (a) Pilot holes are typically repeated at fixed distances along the object's boundary. Their distance from the edge is usually also fixed. This would result in the Fourier Descriptors of the extended features being very similar.

   (b) A large number of extended features would be generated. This would increase the computational complexity of the technique.

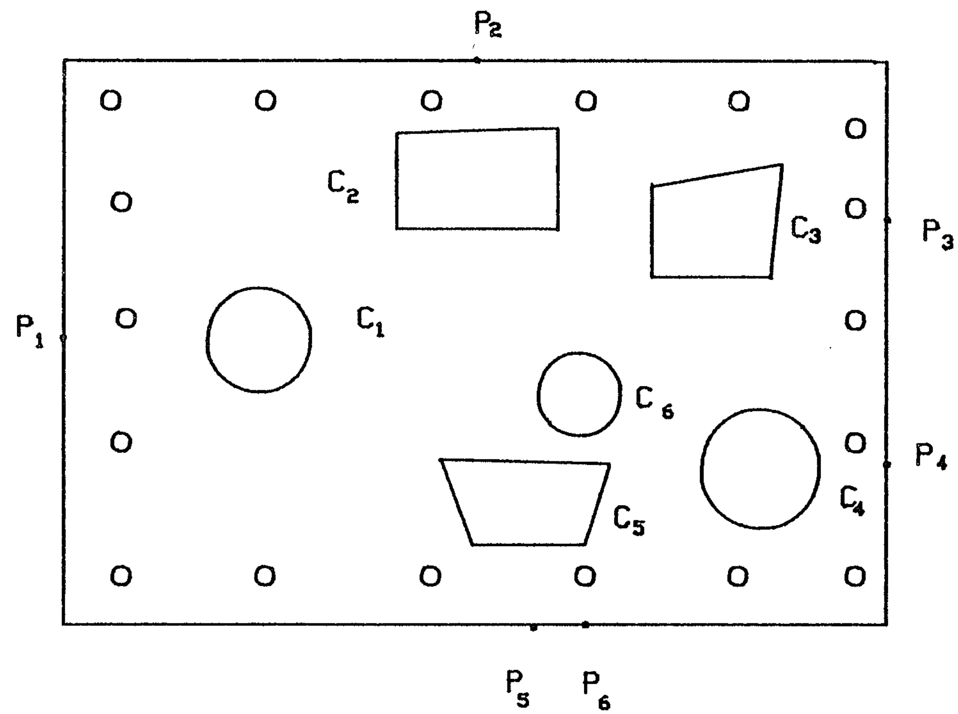

Figure 3.14: A Complex Part With Six Internal Features

The pilot holes are generally of a fixed size (even independent of the size of rivets). Hence they can easily be recognized. In the proposed technique pilot holes are ignored in the generation of extended features.

(2) The elimination of pilot holes leaves six internal features ($C_1, C_2, C_3, C_4, C_5, C_6$) for the generation of extended features. Critical points, p_1 to p_6, corresponding to the six internal features, are shown in figure 3.14. It can be seen that P_5 and P_6 are close together. This is undesirable for the following reasons;

(a) Extended segment generated from critical points P_5 and P_6 would contain information on a very small boundary segment.

(b) Critical points P_5 and P_6 can lie at the same point on the global boundary curve.

Hence it is proposed that, when the shortest cord from an internal features' geometric center to the object's global boundary, passes through another internal feature, then the first feature is ignored in the generation of extended features.

3.2.2 Heuristic #2 For The Generation of Extended Features and Fourier Descriptors For Objects With Less Than Two Internal Features

In this heuristic we consider simple objects, with less than two internal features. In these cases, extended features and foci are generated as follows;

(1) Ignore the presence of any internal features in the generation of extended features.

(2) Compute the curvature and the change of curvature at all points on the global boundary curve.

(3) A maximum of ten points with the highest change in curvature and a change in curvature higher than a threshold value, are selected as critical points. For example, with reference to the object in figure 3.15, points P_1, P_2 and P_3 are chosen as critical points. Similarly, with reference to the object in figure 3.16, points P_1, P_2, P_3, P_4 and P_5 are chosen as critical points.

(4) As defined in the previous heuristic, boundary segments between critical points are defined as extended features.

5) To demonstrate the computation of the coordinates of the focus of extended features we consider extended feature P_4-P_5 in figure 3.16. The (x,y) coordinates of the focus are computed as follows;

$$x = (xP_4 + xP_5)/2$$

$$y = (yP_4 + yP_5)/2$$

where

(xP_4,yP_4) are the x and y coordinates of the first critical point, P_4.

Similarly (xP_5,yP_5) are the x and y coordinates of the second point, P_5.

From this point the computation of the Fourier Descriptors of the extended segments are identical to that described in section 3.2.1.

To summarize the generation of Extended features by the proposed Heuristics, the following observations are made;

(1) Regardless of shape, complex objects with more than two internal features, can be recognized using the extended features defined in the proposed heuristic #1. This heuristic requires a minimum of two internal features to generate extended segments. This is not a serious limitation as complex objects typically have several internal features.

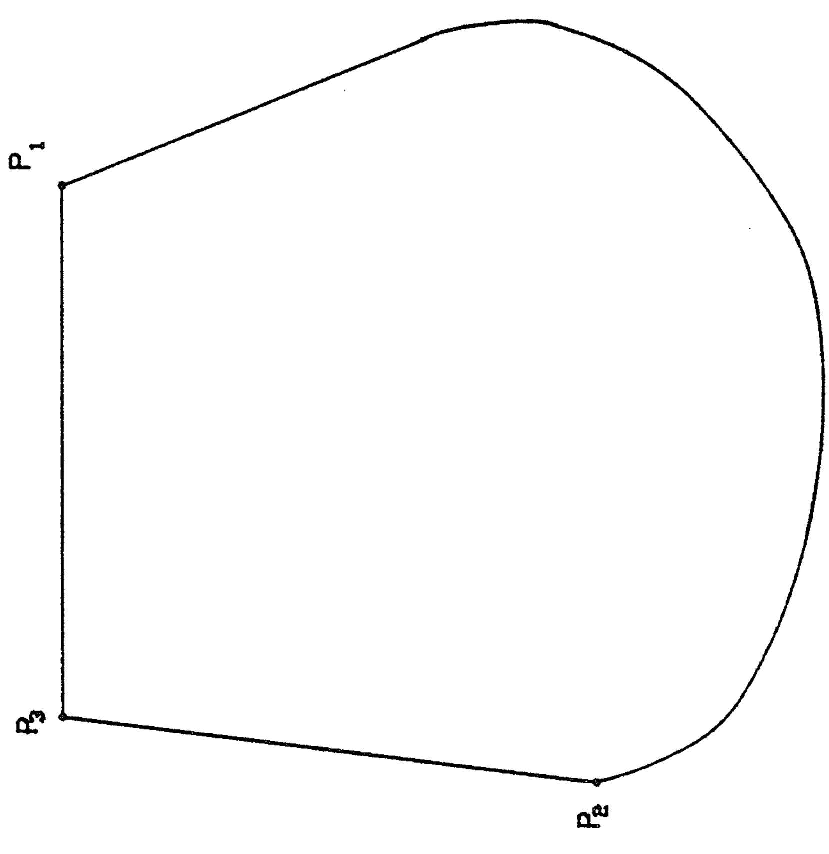

Figure 3.15: Object With No Internal Features

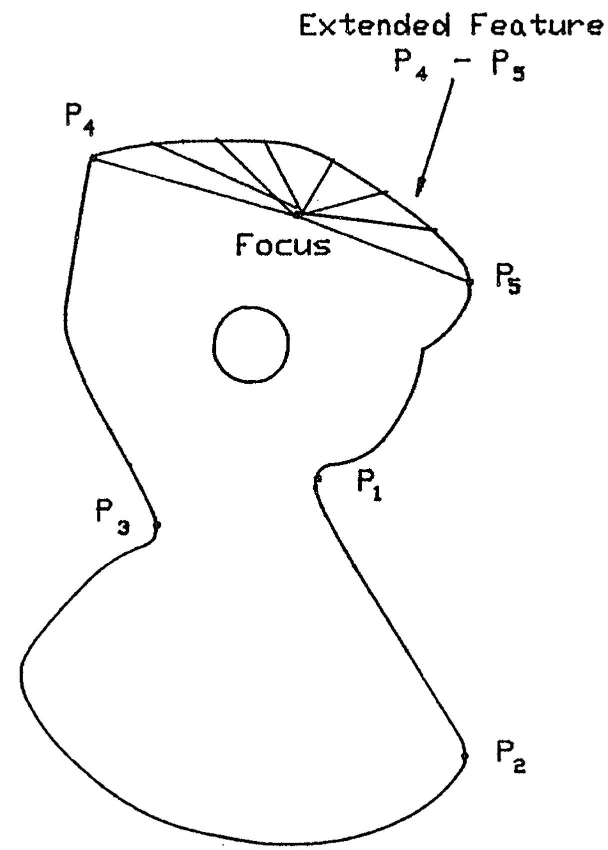

Figure 3.16: Object With One Internal Feature

(2) For completeness, a second heuristic is proposed, for the generation of extended features, to accommodate objects with less than two internal features. The main disadvantages of the second heuristic are ;

   (a) The extended features do not make use of structural information on the location of any internal feature present.

   (b) If an object has no points with sharp changes in curvature then no extended features would be generated. An example of this would be a global boundary in the shape of a circle.

3.3 Recognition of Overlapping and Touching Objects.

In this section we discuss how it is proposed to recognize a collection of partially occluded complex industrial objects. Objects that are partially occluded can be divided into those that are overlapping and those that are touching.

Lets us assume the criteria for recognizing an object in "runtime" mode is the successful matching of (i) two internal feature and one extended feature for objects with two or more internal features in the "training mode" model, and (ii) one extended feature for objects with less than two internal features in the "training mode" model. The method for object recognition varies depending on whether object boundaries can be segmented. If object boundaries can be segmented then recognition is conducted as follows;

(1) Heuristic discussed in section 3.2.1 is used to generate extended features of segmented objects which have more than two internal features in the

"runtime" image. Internal features and generated extended features are then used to tentatively match "runtime" objects with candidate models in the database. Objects with less than two internal features in the "runtime image" are totally ignored in this step.

(2) The edge image of tentative candidate models are superimposed on the edge image of the corresponding segmented "runtime" objects. If an object is matched correctly, then all portions of the edge image in the "runtime" object must coincide with the corresponding portions in the tentative model object. This step essentially confirms the validity of the match between a "runtime" object and the model object.

(3) Heuristic discussed in 3.2.2 is now used to generate extended features for objects that have less than two internal features in the "runtime" image. As discussed before (i) internal features and generated extended features are then used to tentatively match "runtime" objects with candidate models in the database and (ii) validity of the tentative match is confirmed.

In cases where segmentation of object boundaries is not possible, the method for recognizing objects that are overlapping or touching is more complicated. This method is summarized as follows;

(1) Heuristic discussed in section 3.2.1 is used to generate extended features in the "runtime" image. Internal features and generated extended features are then used to match "runtime" features with candidate models in the database. Objects with less than two internal features are included in the exercise, but will not generate any extended features and will thus not form matches with candidate models.

(2) The edge image of matched candidate models are superimposed on the edge images of the corresponding "runtime" features. Portions of the "runtime" edge image, that do not coincide with

the edge points of any matched candidate models are extracted.

(3) Heuristic discussed in 3.2.2 is now used to generate extended features in the extracted edge image from (2) above. Internal features and generated extended features are then used to match "runtime" features with candidate models in the database.

For purposes of demonstration consider six objects shown in figure 3.17. Figure 3.18 shows these objects overlapping and touching. In table 3.12, we see the effect of segmenting the boundary of objects. It can be seen that if the object boundaries are segmented then all objects can potentially be recognized. Otherwise only four objects can be recognized. Thus the benefits of segmenting the boundary of an object from other objects in the field of view are as follows;

(1) Segmenting will mitigate the possibility of boundaries of different objects merging to produce extended features that would not be recognized.

(2) The matching process is simplified as all the extracted features (internal and extended) from an object can be grouped.

Segmentation of object boundaries is considered possible for the following reasons;

(1) The objects may be of different colors.

(2) The object surface may have different reflectance properties,

(3) In the case of touching objects, the boundary may be detected.

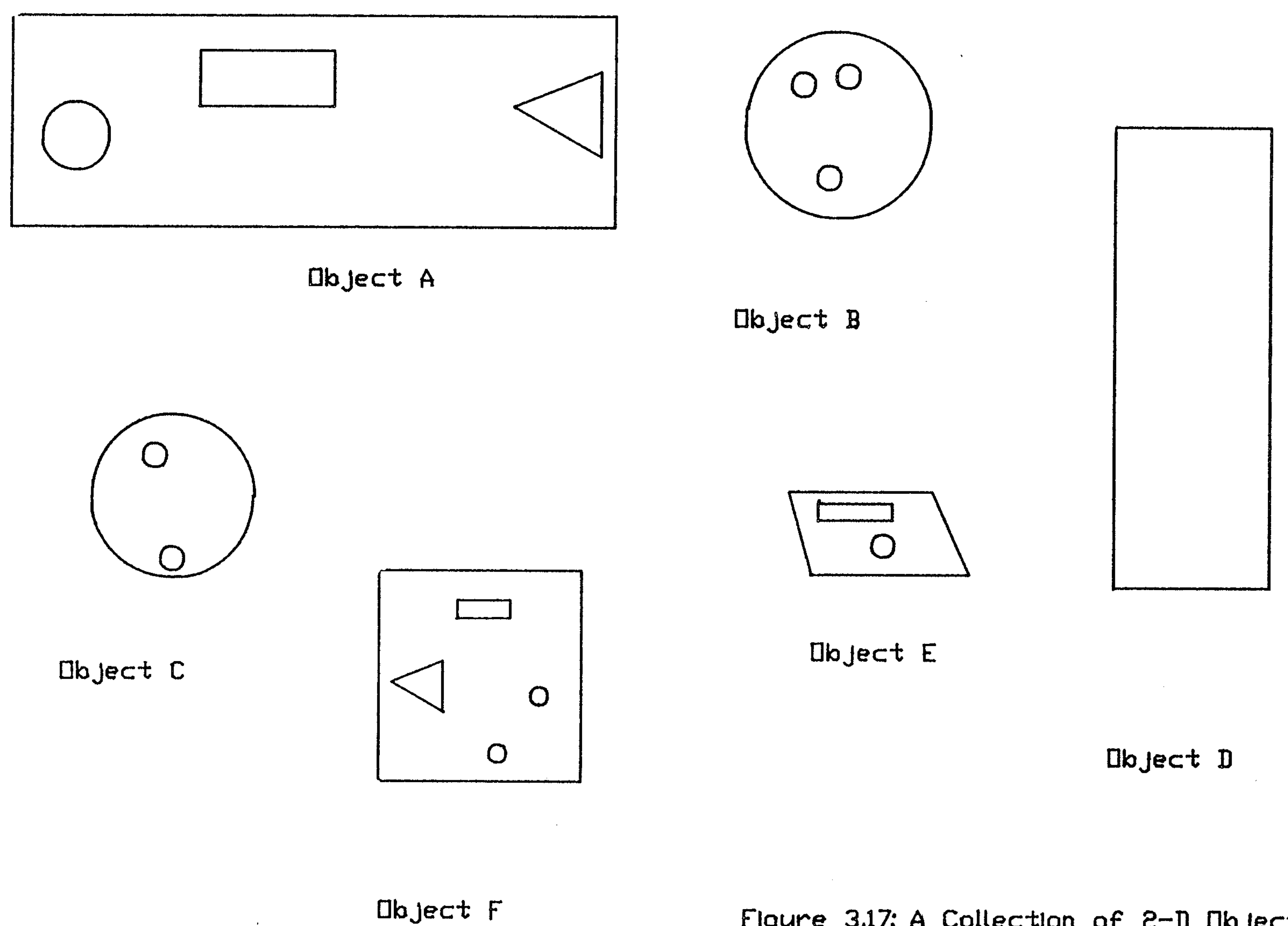

Figure 3.17: A Collection of 2-D Objects

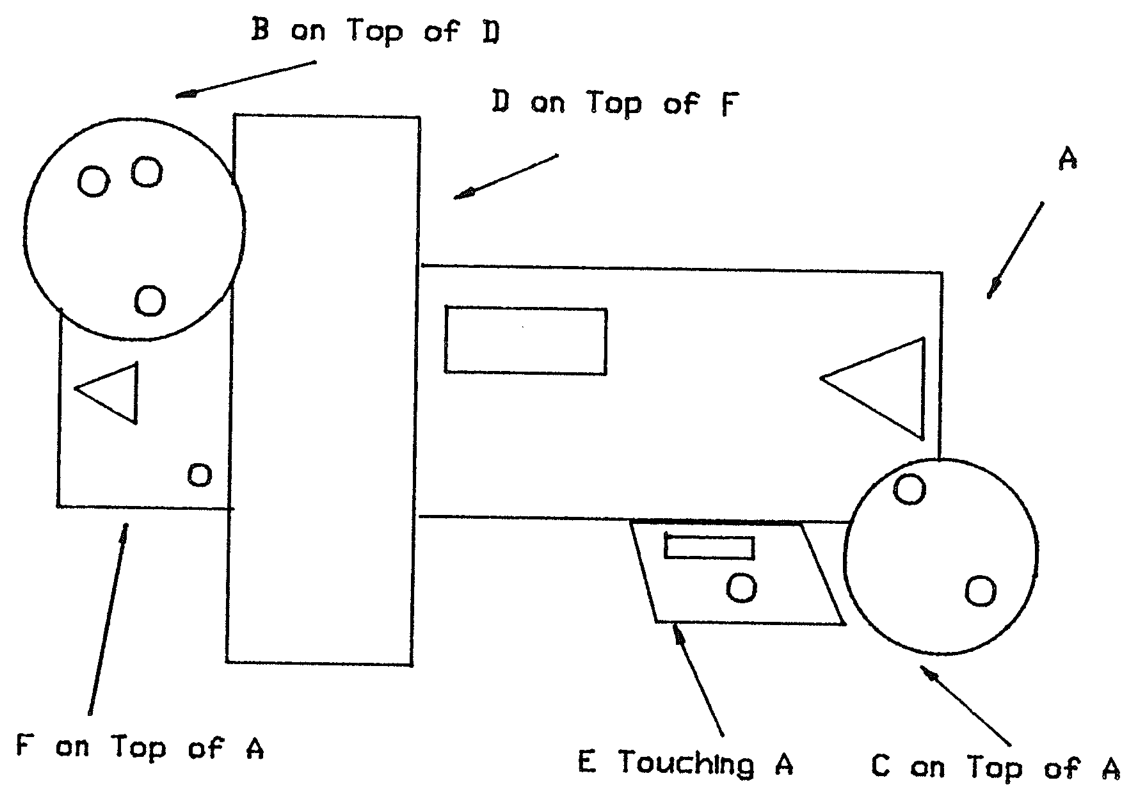

Figure 3.18: Puzzle Consisting of Overlapping and Touching Objects of Figure 3.17

Object	Segmentation: # Local Feature Matched	Segmentation: # Extended Feature Matched	Segmentation: Object matched	No Segmentation: # Local Feature Matched	No Segmentation: # Extended Feature Matched	No Segmentation: Object Matched
A	2	1	Yes	2	1	Yes
B	3	3	Yes	3	1	Yes
C	2	2	Yes	2	0	No
D	0	4	Yes	0	4	Yes
E	2	2	Yes	2	0	No
F	3	1	Yes	3	1	Yes

Table 3.12: The Impact of Segmentation of Object Boundary On Object Recognition

Note; Objects A, B, C, E and F have their extended features generated by the heuristic in 3.2.1 while Object D has its extended features generated by the heuristic in section 3.2.2.

(4) Because objects may be lying partially on top of each other, they will reflect varied degrees of light into the camera.

3.4 Summary: Recognition of Partially Occluded, Complex Industrial Objects.

In this chapter, the basic technique for using extended features for the recognition of complex industrial objects was presented. The advantages of this technique are as follows;

(1) The technique can recognize 2D objects which are partially occluded. Since the technique is based upon Fourier Descriptors, it places no limitation on the shape of the global object boundary or on the shape of the internal features for complex objects (objects with more than two internal features).

(2) Extended features (for objects with two or more internal features) not only carry information about the shape of the object's global boundary but also structural information on the location of internal features. This can improve the efficiency of the matching process.

(3) Extended features can be automatically generated without human intervention.

(4) The proposed method uses information on the structure of the object to generate extended features. As a result a small quantity of extended features, covering the total global boundary, are generated. This is in contrast to the technique presented by Kalvin (1986), where a very large quantity of Fourier Descriptors are calculated (one for each discrete point on the arc vs. turning angle graph).

The main disadvantages of this technique are as follows;

(1) Objects with less than two internal features require sharp changes in curvature in the global object boundary to generate extended features. This dependance on shape limits the application in some instances.

(2) For objects with less than two internal features, the extended features generated, are not based upon structural information on the location of any internal feature which may be present.

(3) If large portions of an object are occluded then recognition will only be based upon the matching of visible internal features as no extended features may be generated.

The first two disadvantages are inherent limitation of the heuristic for objects with less than two internal features. Thus this technique is more applicable to complex parts with multiple internal features and hence the title of this thesis "Recognition of Partially Occluded Complex Industrial Objects". The third disadvantage is also an inherent limitation of this technique but is not limited to the proposed method.

In the next chapter we shall present some experimental results from a preliminary computer vision system for the recognition of partially occluded objects with more than two internal features which have had their boundaries segmented. We will also present statistical analysis of repeatability of dimensional data. Determination of repeatability will enable the recognition of dimensionally defective objects.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter we present results from a computer vision system, developed for the recognition of partially occluded, complex industrial objects whose boundaries have been segmented. This implementation is limited to handling objects that have two or more internal features (as outlined in section 3.2.1). We also present statistical analysis on the repeatability of dimensional results.

In section 4.1 we describe the operator interface, for operating the system in "Training" and "Runtime" modes. In section 4.2 we describe the format of data in object models. Object models are (i) stored in a database file during "training" mode and (ii) are used for matching in the "runtime" mode. In section 4.3 we present examples involving the recognition of partially occluded objects. In section 4.4, we present examples showing the extraction of basic shapes from the Fourier Descriptors. Finally, in section 4.5, we present analysis of repeatability of dimensional data associated with basic shapes.

4.1 Operator Interface

The interaction of the system with the human operator is described in this section. Appendix A1 shows typical interaction with a human operator in the "training" mode while Appendix A2 shows interaction in the "runtime" mode. For purpose of demonstration, in both instances, the same part (figure 4.1, serial #2222 in our experimental database) is used. Upon start up, the system requires the operator to select an operating mode from the following;

CONDUCT OBJECT RECOGNITION
AUTO CALIBRATE THE SYSTEM (NOT IMPLEMENTED)
ADD AN OBJECT TO THE DATABASE
LEAVE CompuVision & EXIT TO DOS

If the operator selects "ADD AN OBJECT TO THE DATABASE" (ie. "training" mode), then the system asks for the object's serial number as follows;

INPUT SERIAL NO. OF OBJ.(INT.<9999) & THEN PRESS ENTER

The selection of 9999 as the highest serial number is arbitrary to some extent. It was felt that serial numbers up to 9,999 were sufficient for our experimental database.

Upon receiving the serial number, the system grabs an image from the video camera and proceeds with the construction of a model. As a diagnostic message, the system prints the number of internal features found, the length of their

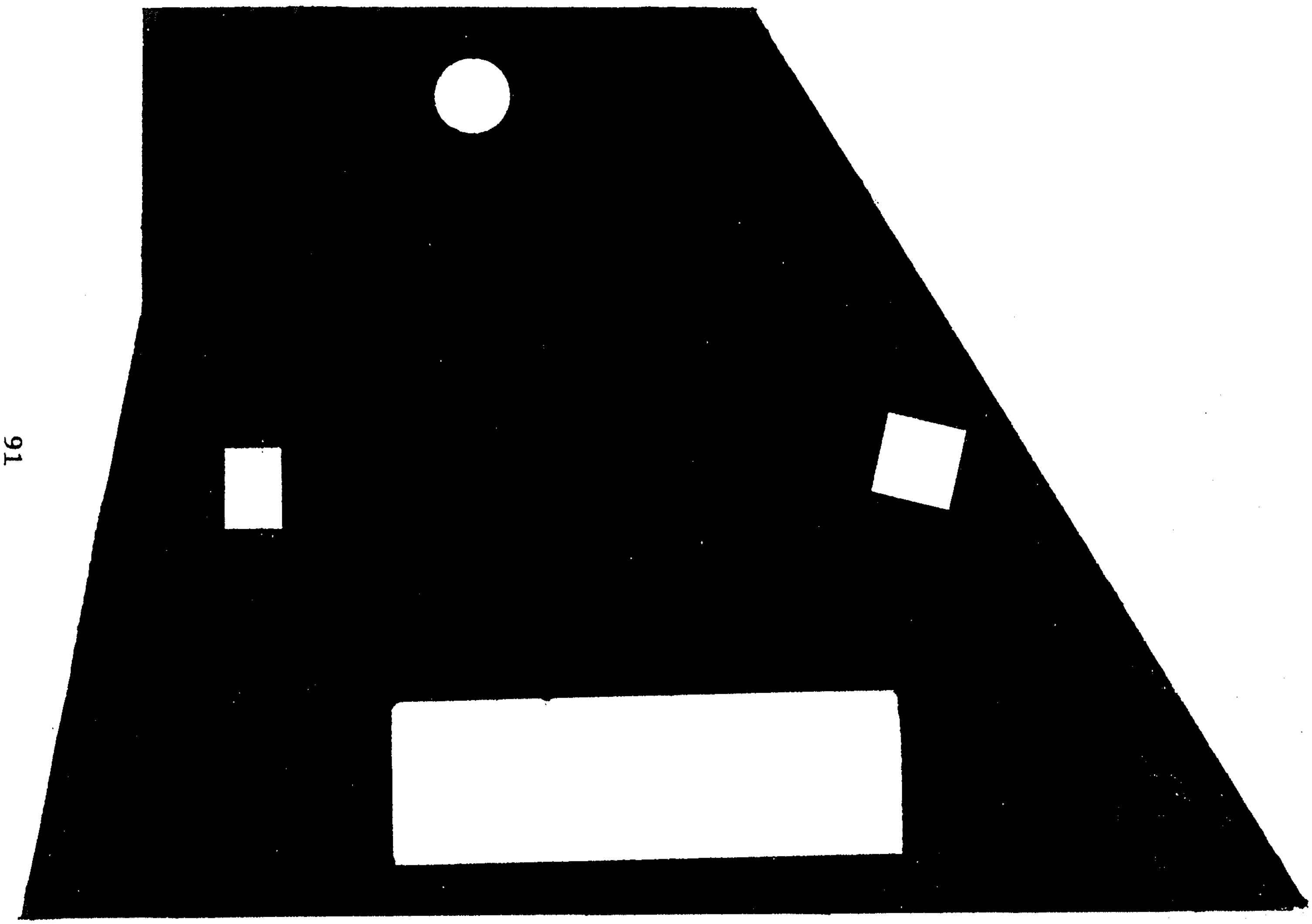

Figure 4.1: Complex Object With Serial #2222
In Database

boundaries and the position of their geometric center as follows;

The following (internal) features have been extracted.

Feature#	# of Edge Points	X - C.of G.	Y - C.of G.
0	593	228.4	252.5
1	211	218.0	211.5
2	40	282.1	257.8
3	47	172.0	260.2
4	35	244.6	318.7

The screen display above indicates that besides the global boundary, four internal features were found. In a future implementation there would be no need for this diagnostic. It is only included in this preliminary system to enable the operator to confirm that all is well in the primitive image processing stage (which include thresholding, edge detection, segmentation of edge image into global boundary and boundaries of internal features).

The system continues with the creation of the model and upon completion prints the following message;

Addition of Serial# 2222 to Dbase Completed.

This indicates that the system has successfully created a model of serial # 2222 and has added it to the database file.

The system now cycles back to the initial menu and enables the operator to select "CONDUCT OBJECT RECOGNITION", (ie. "runtime" mode). The system then enters recognition mode and grabs an image from the camera and continues with its processing. As discussed before, the system prints a diagnostic message as follows;

The following (internal) features have been extracted.

Feature#	# of Edge Points	X - C.of G.	Y - C.of G.
0	592	228.4	252.4
1	213	217.7	211.3
2	41	282.2	257.8
3	47	172.1	260.2
4	35	244.8	318.6

The system continues with the creation and matching of the "runtime" model. Upon completion, it prints the following;

Object Recognition Now Completed

Based Upon Matching
4 Out of 4 Local Features &
4 Out of 4 Global Boundary Segments
The Current Object Has Been Matched
To Serial #2222 From The Database

In regards the "runtime" object, the system now asks if further information on (i) the tentative matching process or (ii) object dimensions and shape, are required. If the response is "yes" then the system prints the following information on the screen;

Details of Matching Current Object to One From the Database
****

Current Object Features		Serial #2222 Features	
Internal	Extended	Internal	Extended
1		1	
2		2	
3		3	
4		4	
	1 - 3		1 - 3
	2 - 1		2 - 1
	3 - 4		3 - 4
	4 - 2		4 - 2

FEATURE#	SHAPE	DIM#1 (mm)	DIM#2 (mm)	POSITION COORDINATES X (mm)	Y (mm)
1	RECTANGLE	29.30	98.15	217.73	211.31
2	RECTANGLE	10.58	17.17	282.20	257.83
3	SQUARE	16.67		172.05	260.17
4	CIRCLE	13.77		244.79	318.62

The screen display above, shows (i) which features (internal and extended) in the "runtime object" are tentatively matched with which features in the candidate model and (ii) gives the shape, size and location of the internal features in the "runtime object".

4.2 Format of Data in Object Model

In this implementation, object models are created and stored (i) in a database file (dbase1.dat) in "training" mode and (ii) temporarily in a current object file (cobj.dat) for use in the matching process in the "runtime" mode. As an example, we will again consider the part with serial #2222. Figure 4.2 shows the format of data in the model of the object. The four internal features give rise to four extended features (Note: As indicated in section 3.2.1, the number of extended features can be more than the number of internal features. In this preliminary implementation, internal features are strategically located such that they each create only one critical point. This results in there being only one extended feature per internal feature).

In figure 4.3, the data for each internal feature consists of (i) the minimum and maximum distances of edge points from the focus, (ii) the number of edge points in the boundary, (iii) the area of the internal feature, (iv) the x and y coordinate of the geometric center (using camera coordinate system; the top left hand corner of camera's field of view is defined as [0,0]), (v) the first fifteen terms/coefficients of the Fourier Descriptors (each term includes the real and imaginary components and the magnitude).

```
**********************************************************
*  The data is placed in an ASCII file in the following  *
*  order;                                                 *
**********************************************************
* "8888.8" to signify start of object model.             *
*                                                         *
* Object's serial number.                                 *
*                                                         *
* "4" to signify the number of internal features in      *
* object.                                                 *
*                                                         *
* "999999.9" to signify data on internal feat. # 4 to    *
*  follows.                                               *
* data on feature #4 (format shown in figure 4.3).       *
*                                                         *
* "999999.9" to signify data on internal feat. # 3 to    *
*  follows.                                               *
* data on feature #3 (format shown in figure 4.3).       *
*                                                         *
* "999999.9" to signify data on internal feat. # 2 to    *
* follow.                                                 *
* data on feature #2 (format shown in figure 4.3).       *
*                                                         *
* "999999.9" to signify data on internal feat. # 1 to    *
* follow.                                                 *
* data on feature #1 (format shown in figure 4.3).       *
*                                                         *
* "999999.9" followed by an integer "x" signify data on  *
*  extended feat. # (1 - x) to follow.                    *
* data on feature #(1-x) (format shown in figure 4.3).   *
*                                                         *
* "999999.9" followed by an integer "x" signify data on  *
* extended feat. # (2 - x) to follow.                     *
* data on feature #(2-x) (format shown in figure 4.3).   *
*                                                         *
* "999999.9" followed by an integer "x" signify data on  *
* extended feat. # (3 - x) to follow.                     *
* data on feature #(3-x) (format shown in figure 4.3).   *
*                                                         *
* "999999.9" followed by an integer "x" signify data on  *
*  extended feat. # (4 - x) to follow.                    *
* data on feature #(4-x) (format shown in figure 4.3).   *
**********************************************************
```

Figure 4.2 Format of Data in the Model of an Object with Four Internal Features.

```
********************************************************
*  Data is placed in an ASCII file in the             *
*  following order;                                    *
********************************************************
*                                                      *
*                                                      *
* Minimum distance of edge points from the focus to   *
* the geometric center                                 *
*                                                      *
* Maximum distance of edge points from the focus      *
* to the geometric center                              *
*                                                      *
* The number of edge points in the boundary of        *
* the internal feature.                                *
*                                                      *
* The area of the internal feature.                   *
*                                                      *
* The x and y coordinate of the focus.                *
*                                                      *
*  The first fifteen terms/parameters of the Fourier *
*  Descriptors as follows;.                           *
*                                                      *
*  For term 1 - real, imaginary, mag.,                *
*  For term 2 - real, imaginary, mag.,                *
*  For term 3 - real, imaginary, mag.,                *
*  For term 4 - real, imaginary, mag.,                *
*  For term 5 - real, imaginary, mag.,                *
*  For term 6 - real, imaginary, mag.,                *
*  For term 7 - real, imaginary, mag.,                *
*  For term 8 - real, imaginary, mag.,                *
*  For term 9 - real, imaginary, mag.,                *
*  For term 10 - real, imaginary, mag.,               *
*  For term 11 - real, imaginary, mag.,               *
*  For term 12 - real, imaginary, mag.,               *
*  For term 13 - real, imaginary, mag.,               *
*  For term 14 - real, imaginary, mag.,               *
*  For term 15 - real, imaginary, mag.,               *
********************************************************
```

Figure 4.3 Format of Data for Every Feature in an Object Model.

Similarly for extended features, with reference to figure 4.3, the data consists of (i) the minimum and maximum distances of edge points from the focus, (ii) the number of edge points in the boundary of the extended feature, (iii) the area of the extended feature (in the case of the extended feature this is equal to the perimeter or the number of edge points in the boundary), (iv) the x and y coordinate of the geometric center (using camera coordinate system), (v) the first fifteen terms/coefficients of the Fourier Descriptors (as before, each term includes the real and imaginary components, and their magnitude). In the current implementation (i) the number of edge points and the area of extended features, (ii) the location of the focus of an extended feature, are not used for matching and hence do not have any functional value. In future implementations these can be eliminated.

The average size of a object model (with four internal features) is 5 kilobytes. Thus a database of 100 objects will be about 0.5 megabytes in size. By comparison, in Kalvin's (1986) technique, a database for 100 objects is 100 megabytes in size.

4.3 Recognition of Partially Occluded Objects

In this section we present several examples involving the recognition of partially occluded complex objects (with more than two internal features) which have already been segmented from other overlapping and touching objects. These segmented objects can also be construed to be portions of a complex object which has not been fully assembled.

A database of fully visible objects is created in "training" mode and is shown in Appendix B3. It consists of models of the following six objects;

Part With Serial #1111 as figure 4.4
Part with Serial #2222 as figure 4.1
Part with Serial #3333 as figure 4.5
Part with Serial #4444 as figure 4.6
Part With Serial #5555 as figure 4.7
Part with Serial #6666 as figure 4.8

Table 4.1 below, summarizes some recognition exercises conducted using partially occluded objects shown in figures 4.9 to 4.16. The recognition algorithm is programmed to declare a match if (i) two or more internal features and (ii) one or more extended features, are matched with a candidate model. In table 4.1, it can be seen that all partially occluded objects are recognized except for the object in figure 4.15. Looking at figure 4.15 it is evident that only one internal feature is visible. Since two internal features are required to generate an extended feature, no extended features were formed. Hence the object was not matched.

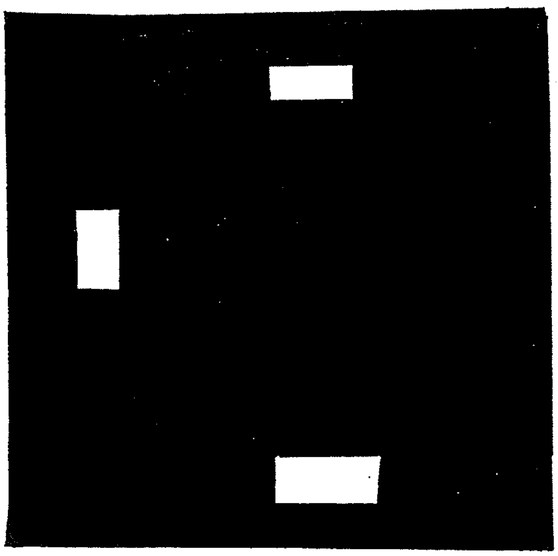

Figure 4.4: Complex Object With Serial #1111
In Database

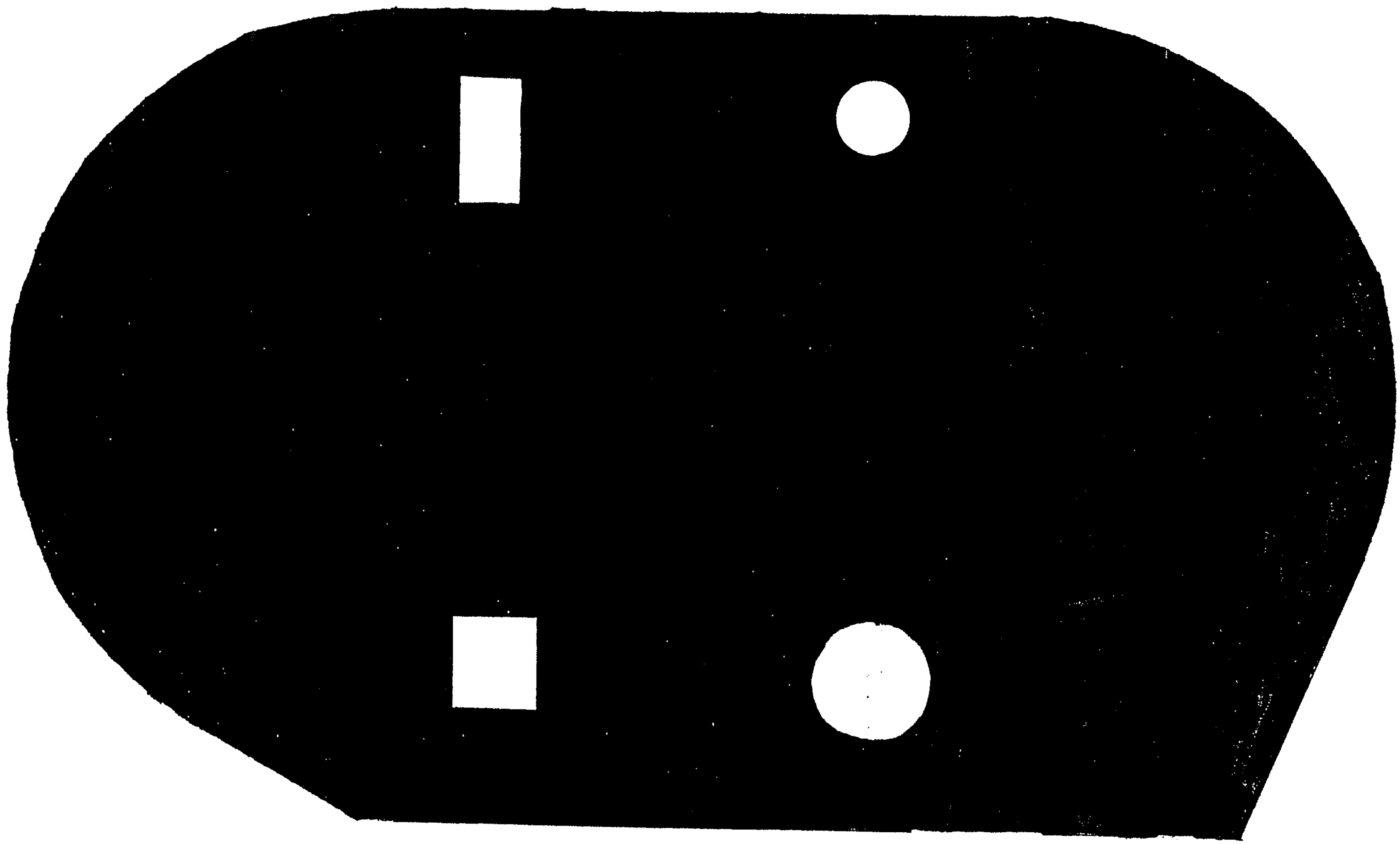

Figure 4.5: Complex Object With Serial #3333
In Database

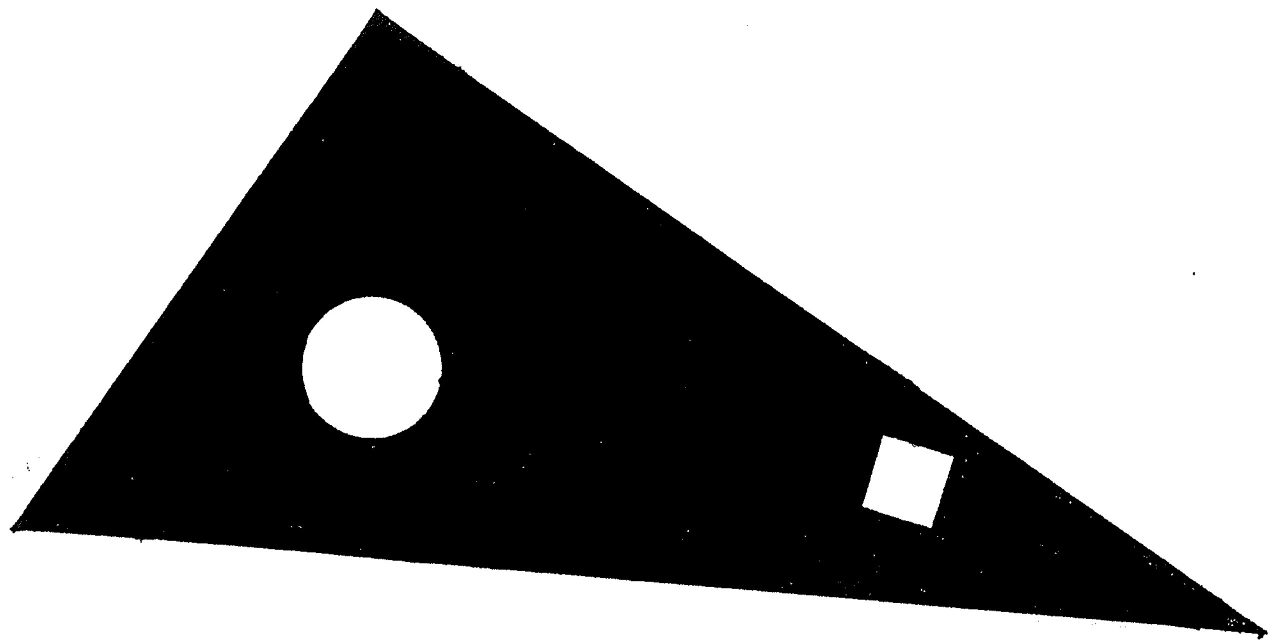

Figure 4.6: Complex Object With Serial #4444
In Database

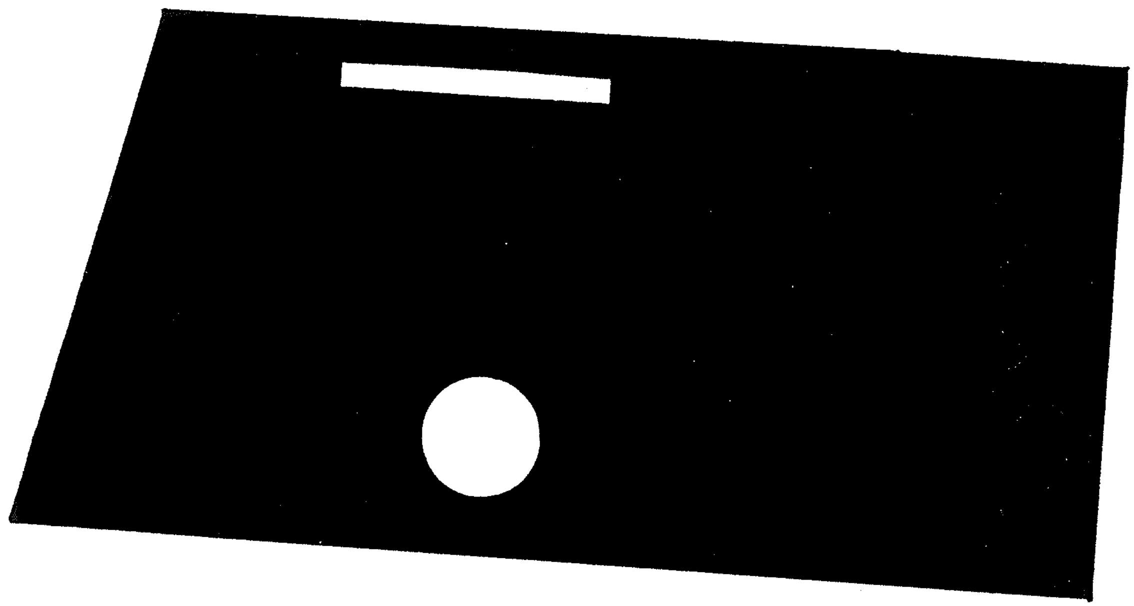

Figure 4.7: Complex Object With Serial #5555 In Database

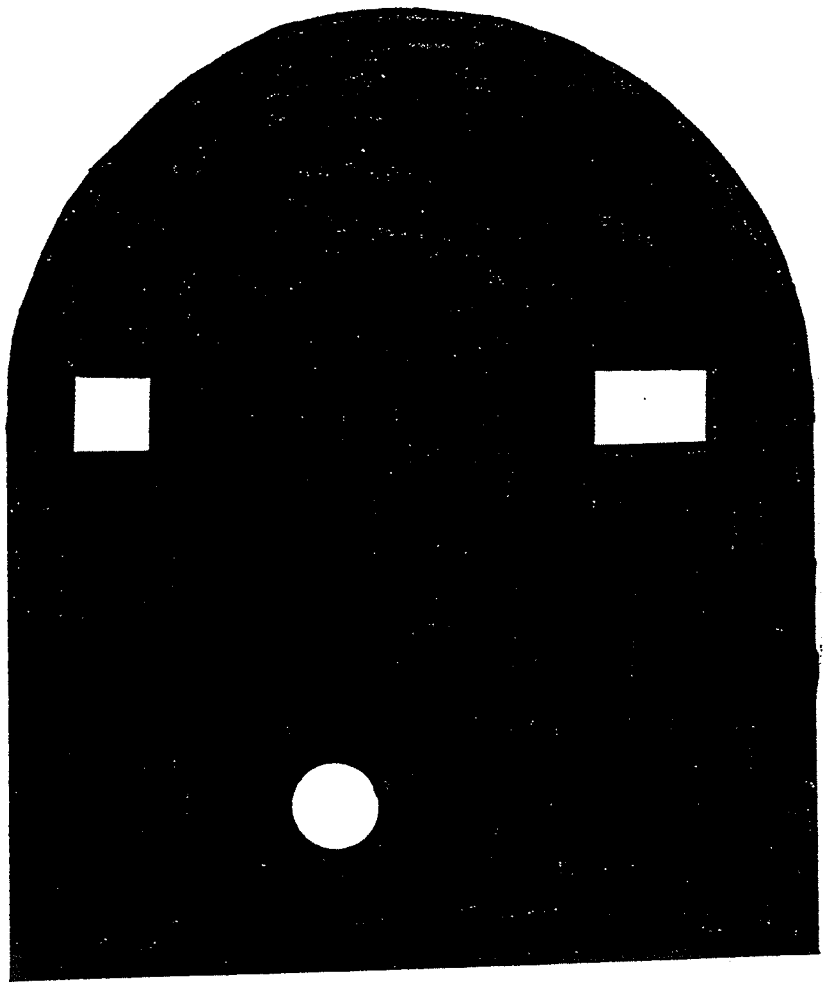

Figure 4.8: Complex Object With Serial #6666
In Database

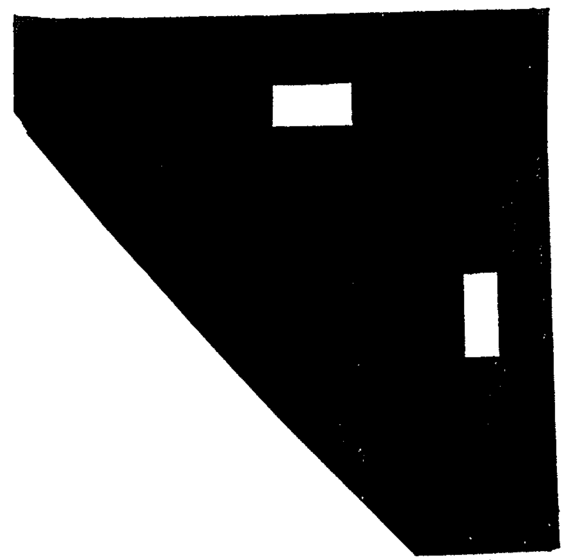

Figure 4.9: Partially Occluded Object

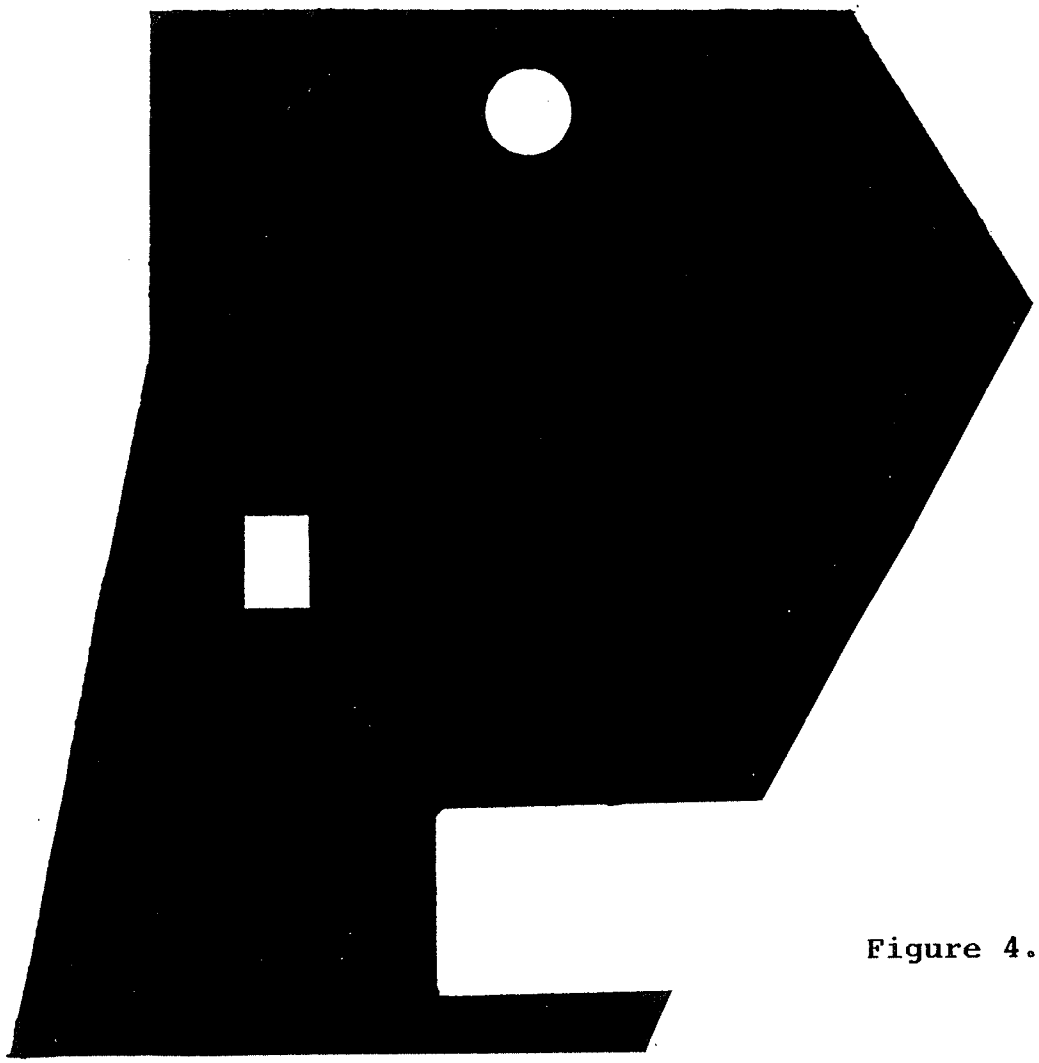

Figure 4.10: Partially Occluded Object

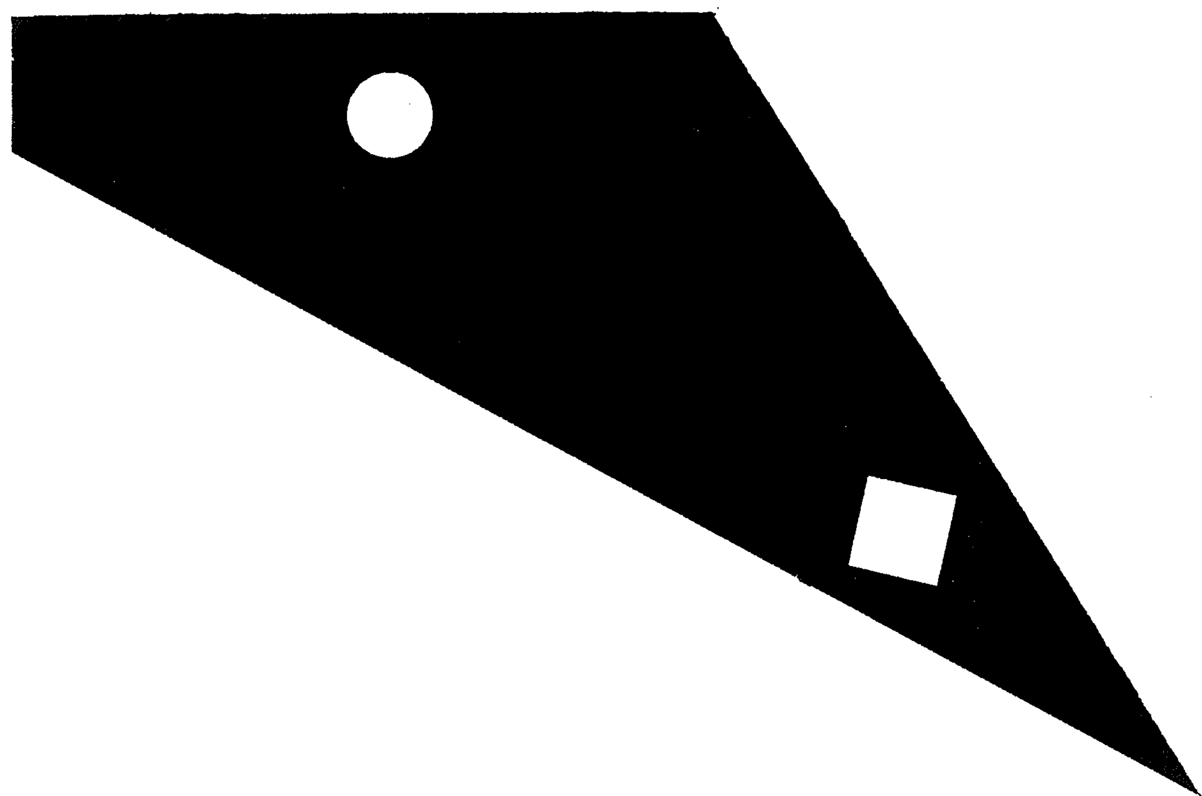

Figure 4.11: Partially Occluded Object

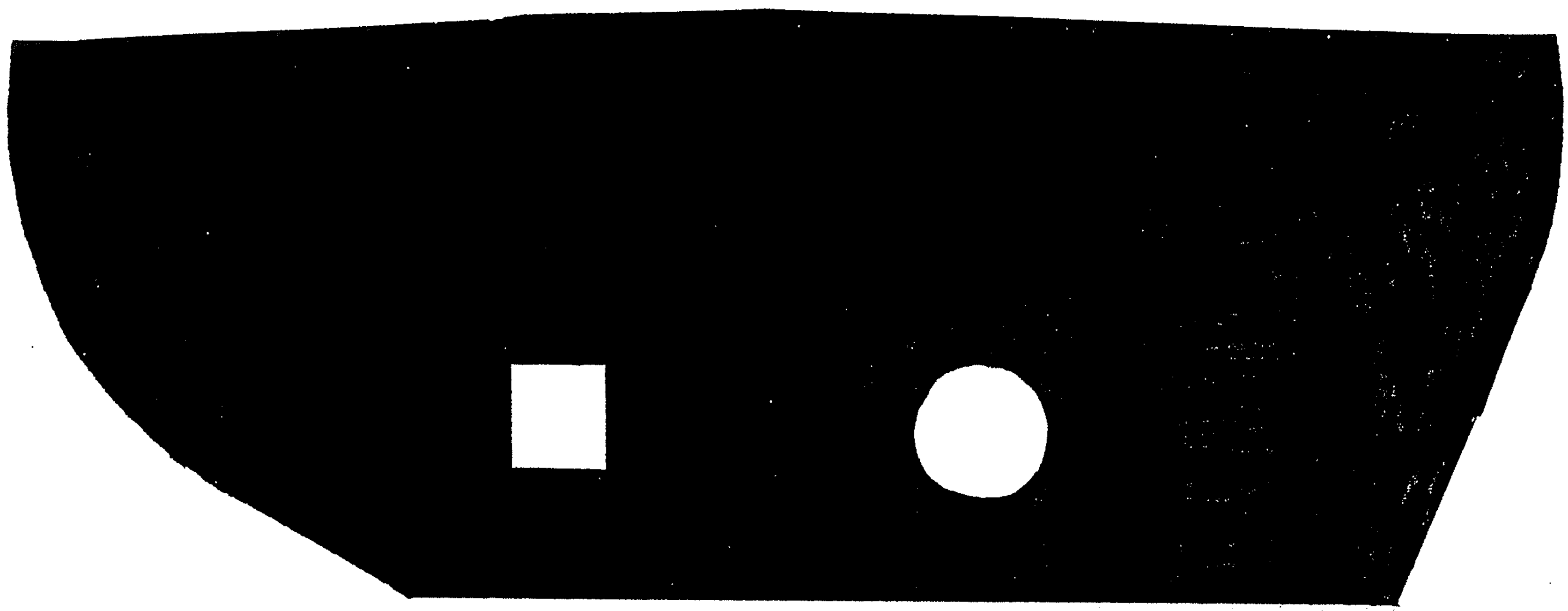

Figure 4.12: Partially Occluded Object

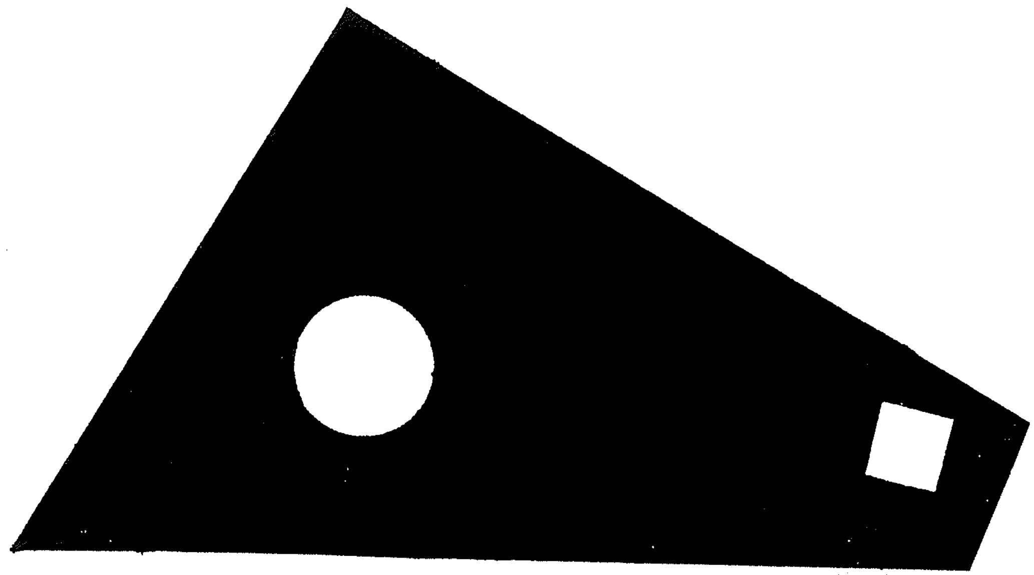

Figure 4.13: Partially Occluded Object

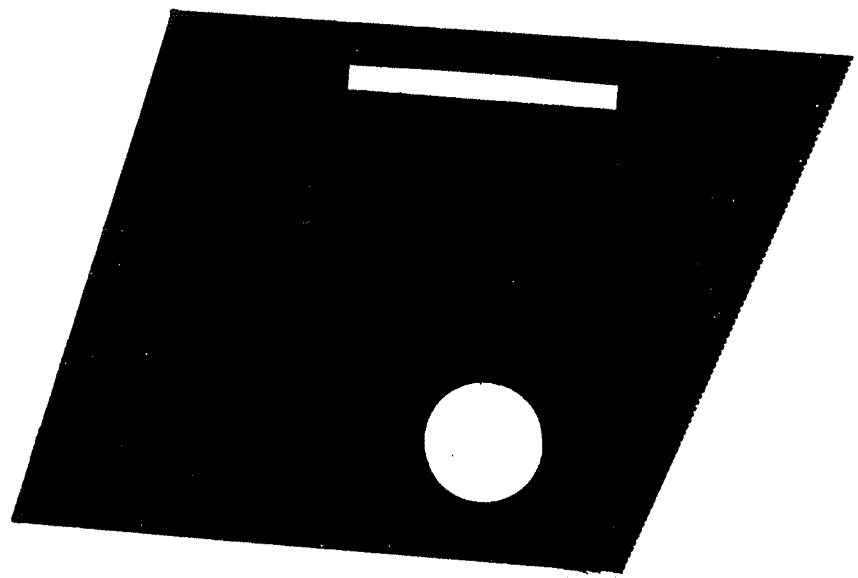

Figure 4.14: Partially Occluded Object

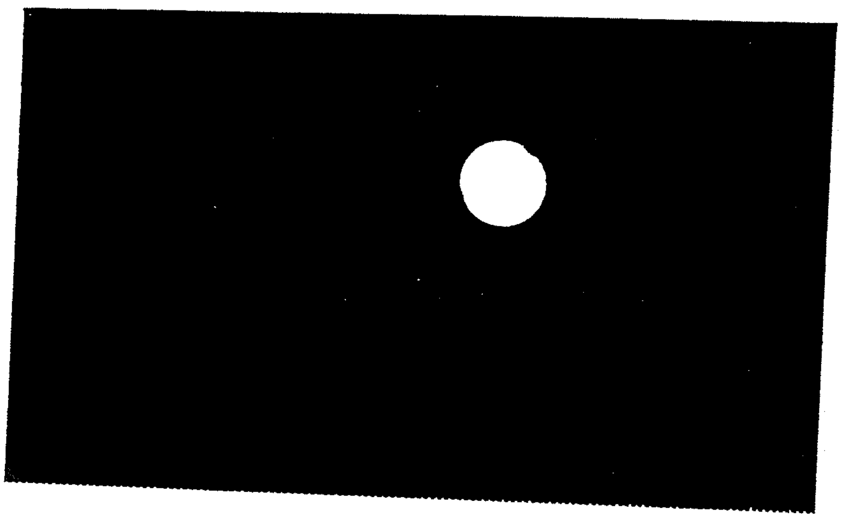

Figure 4.15: Partially Occluded Object

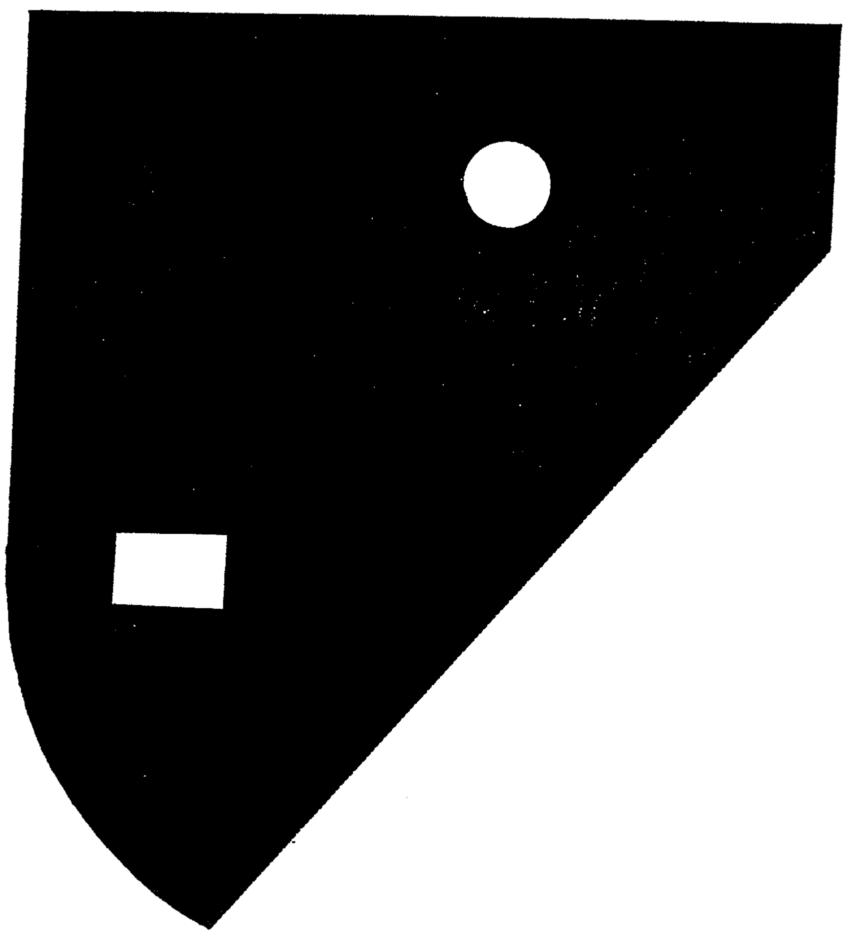

Figure 4.16: Partially Occluded Object

Partially Occluded Object's Figure	Serial # Matched Against	Features in Object Model In Database		Features Successfully Matched	
		Internal	Extended	Internal	Extended
4.9	1111	3	3	2	1
4.10	2222	4	4	2	1
4.11	2222	4	4	2	1
4.12	3333	4	4	2	1
4.13	4444	2	2	2	1
4.14	5555	2	2	2	1
4.15	-	-	-	-	-
4.16	6666	2	2	2	1

Table 4.1 Recognition Exercises Involving Partially Occluded Objects.

4.4 Recognition of Basic Shapes

Now we will experimentally demonstrate (what was previously indicated in section 3.1) the proposed descriptors' ability to recognize basic shapes. To demonstrate this, consider the magnitude of Fourier Descriptors of (i) the internal features of figures 4.1 obtained during "runtime" mode and shown in Appendix A4, and (ii) of internal features of figure 4.5, shown in Database in Appendix A3. Table 4.2 is an extract from these figures. As previously indicated in section 3.1, the following conclusion can be drawn from the data;

(1) A circular object is shapeless. Besides the DC term, all other coefficients are low. The ideal value of coefficients 2 to 15 is zero. The random values obtained are due to noise. Noise could have been introduced in a variety of places including the following; (i) the objects under the camera were made of paper with shapes drawn with pencil, this would have led to shapes that were not ideal, (ii) the process of thresholding, (iii) the process of edge detection and (iv) the computation of $r(\theta)$ vs. θ.

(2) A rectangular object has the third, fifth, seventh coefficients high. The rest are low. In table 4.2, we note that for figure 4.5, the fifth coefficient in the descriptor for the rectangle is not high. This discrepancy is attributed to noise.

(3) A square object has the fifth and ninth coefficients high.

Fourier Coefficients

Figure 4.1			Figure 4.5		
Large Rect.	Square	Circle	Rect.	Square	Large Circle
5636	1735	1452	1920	1945	2362
24	4	1	3	3	1
1717	12	18	360	76	14
20	16	2	7	10	16
607	110	13	12	121	22
35	6	4	11	5	5
114	15	2	61	26	7
26	2	3	11	9	10
103	23	13	38	29	4
7	1	5	9	5	9
160	7	9	8	9	10
7	7	8	4	7	9
127	28	2	7	7	7
8	7	5	1	4	10
64	4	4	1	1	2

Table 4.2 Fourier Descriptors of Basic Shapes, Extracted From Models of Figures 4.1 and 4.5 in the Database in Appendix A3.

4.5 Dimensional Analysis of Repeatability

There are two types of errors in any measurement system. These are systematic and random error. In our application, the systematic errors would include the following;

Scaling
Distortion in Camera Optics

In our application we are not measuring distances to sub-pixel accuracies. Hence distortion in the lens and image sensor are not major factors. This leaves scaling as the other systematic error. Scaling can be addressed by introducing a calibration factor (discussed in section 5.2.5). Thus measurement of system accuracy is not important, as it is largely dependant on the selection of a calibration factor. Random errors are of concern and hence we determine repeatability as a measure of randomness.

Statistical data was collected and an analysis for determining the repeatability of dimensions of internal features was conducted. As a test case, the object in figure 4.1 was selected. Table 4.3 below summarizes the analysis. In table 4.3, the three sigma repeatability varied between 0.291 mm for small dimensions (6 mm), to 1.130 mm for medium size dimensions (14 mm), to 1.171 for large size dimensions (48 mm). Since the calibration factor was 1.1, this translates into a three sigma repeatability of 1.065 (1.17 /

	Internal Features (mm)		
		Large Rectangle	
Sample #	Circle	Length	Width
1	6.349	48.686	14.822
2	6.194	48.099	14.234
3	6.218	48.054	14.116
4	6.392	48.807	14.822
Sigma	0.097	0.390	0.377
Three Sigma	0.291	1.1712	1.130

Table 4.3 Statistical Analysis for Repeatability on the Dimensions of Internal Features of Object in Figure 4.1.

1.1) pixels. This level of repeatability is the best that can be expected as the edge detection algorithm finds edges only to the closest pixel.

Knowing the dimensional repeatability, and the shape of features, the system can categorize parts with dimensions outside a given 3 sigma range as defective.

4.6 Summary of Experimental Results

The summary of experimental results presented in this chapter is as follows;

(1) From a description of the operator interface, it is evident that the method of forming models in "runtime" and "training" modes is completely automated. In "training" mode the operator has only to enter the object's serial number. In the "runtime" mode the operator has only to initiate the recognition task.

(2) The average size of an object model is 5 kilobytes. Thus a database of a 100 objects will be about 0.5 megabytes. This database size is two orders of magnitude smaller in comparison to the database in the technique described by Kalvin, (1986).

(3) The proposed technique can recognize (i) complex objects that are overlapping and touching and (ii) portions of complex industrial objects that are not fully assembled. The current implementation is limited to handling objects segmented objects with two or more local internal features.

(4) The system was able to automatically recognize basic shape and measure basic dimensions. The three sigma repeatability of dimensions of local internal features was found to be approximately 1.06 pixels. The determination of repeatability will enable the recognition of parts that are dimensionally defective. Algorithms for computing

edges to sub-pixel accuracy will enhance this dimensional measurement capability [Lyvers et al., 1989].

CHAPTER 5

DESCRIPTION OF EXPERIMENTAL SYSTEM

5.1 Image Processing Hardware

The image processing system consisted of the following:

- A general purpose, 500 x 500 pixel video camera (Panasonic color, model # WV-3240)
- A Data Translation Frame Grabber (DT-2853-sq-60Hz).
- A IBM PCAT with a 8087 coprocessor operating at 10 MZ in the DOS environment.
- Turbo "C" Compiler.

The DT 2853 board, like most image processing boards had the capability to do primitive image processing functions such as thresholding, edge detection, segmentation etc.. These capabilities could not be utilized as supporting software was not available. The board was used only to capture the image. The pixel data was then down loaded into the host computer where all subsequent processing took place.

5.2 Software Developed.

In this section software developed is described. All source code is written in Turbo 'C' version 2.0. The system can be divided into two modes. The first is the "training" mode in which object models are added to the database. The second mode is termed the "runtime" mode. In this mode the system conducts recognition on unknown objects.

5.2.1 Thresholding

For purposes of concentrating effort on the primary task ie. the use of Fourier Descriptors of extended features, the task of thresholding was deliberately made trivial. This was achieved by having the object painted black (pixel value 0), in sharp contrast to the white background (pixel value 255). This made the histogram bi-modal. Good thresholding (a measure of separating the object pixels from the background pixels) was achieved by using a threshold value of 125. In an actual manufacturing situation, the thresholding of an image would me considerably more complicated. It would require the use of a technique for the automatic generation of an optimum threshold value for each image.

5.2.2 Edge Detection

A simple technique based upon applying four different 3 x 1 pixel filter (see figure 5.1) to each object pixel was utilized. This produced satisfactory results as is evident from the one pixel thick and unbroken edges of figure 5.2.

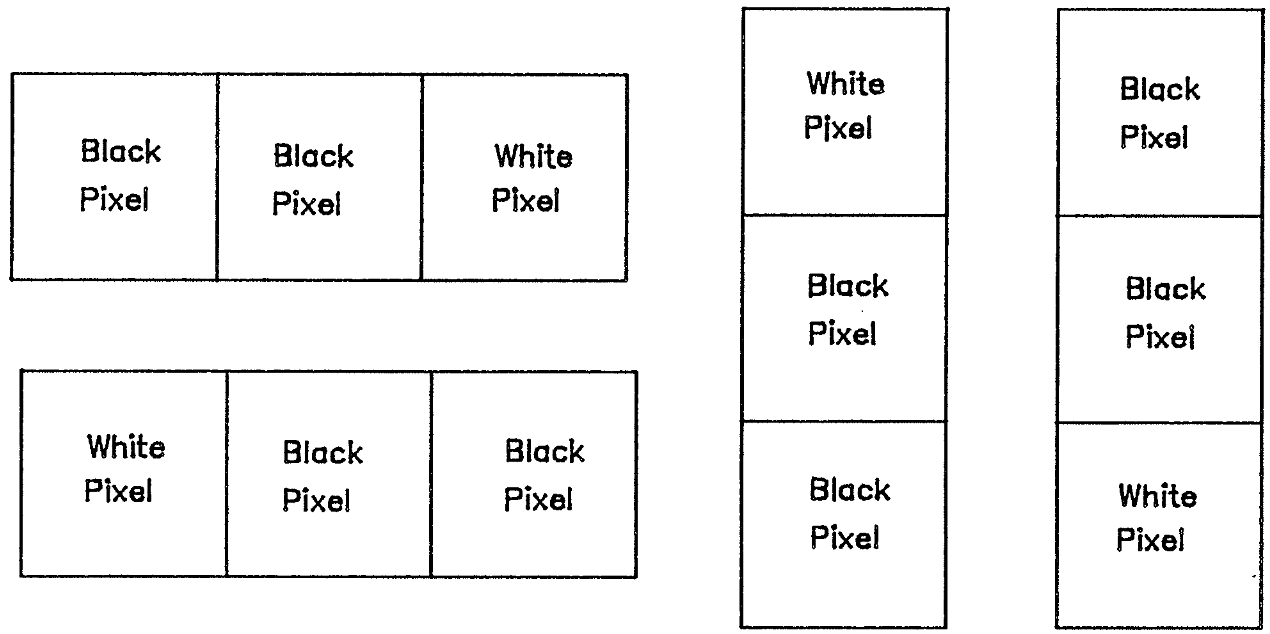

Figure 5.1: 3 x 1 Pixel Filters Used For Edge Detection

```
0000000000000000000000000000000000000000000000000000000000000000
00000ffff3000000000000000000000000000000000000000000000000000000
0000e0000cfffffffffffffffffffffff0000000000000000000000000000000
000020000000000000000000000000000fffffffffffffffffff3cf100000000
0000400000000000000000000000000000000000000000000000c30200000000
0000800000000000000000000000000000000000000000000000000200000000
0000800000000000000000000000000000000000000000000000000100000000
0000010000000000000000000000000000000000000000000000000100000000
0000010000000000000000000000000000000000000000000000000100000000
0000020000000000000000000000000000000000000000000000000100000000
00000200000e0ffffffffffffffffffffffffff7000000000000008000000000
0000040000010000000000000000000000000008000000000000008000000000
0000040000800000000000000000000000000000100000000000008000000000
0000080000800000000000000000000000000000100000000000008000000000
0000001000800000000000000000000000000008000000000000004000000000
0000001000800000000000000000000000000008000000000000004000000000
0000002000800000000000000000000000000008000000000000004000000000
0000004000800000000000000000000000000008000000000000004000000000
0000004000800000000000000000000000000008000000000000002000000000
0000008000800000000000000000000000000008000000000000002000000000
0000000100800000000000000000000000000008000000000000002000000000
0000000100800000000000000000000000000008000000000000002000000000
0000000200800000000000000000000000000008000000000000002000000000
```

Figure 5.2: Hexadecimal Coded Edge Image of a Portion of the Object in Figure 4.1

5.2.3 Detection of Internal Features in Edge Image

The detection of internal features was accomplished by means of a simple algorithm in which the following rules were coded:

(i) All edge pixel on the global boundary of an object are connected.

(ii) All edge pixels of an internal feature are connected.

(iii) If the centers of two edge pixels are less than a specified threshold value then the two edge pixel are considered connected.

(IV) The distance between the edge pixels of any two internal features or the distance between the edge pixels of an internal feature and the object boundary, is large compared to the threshold value in (iii) above. This would typically be the case in industrial parts.

The above was implemented as follows:

(i) Conduct analysis on edge pixels starting at the top left hand corner and proceeding left to right and top to bottom in the edge image.

(ii) The first pixel would belong to the object's global boundary. This would be placed in boundary feature #0. At this time feature #0 would have one pixel while internal features #1, #2, #3, #4, #5 would be empty. Due to memory limitations the system was limited to five internal features.

(iii) The second and all subsequent pixels will be tested to see if they are connected (a distance less then the threshold) to any pixel in any feature. If a current pixel is connected to a feature, it is then added to that feature. If the pixel is not connected to any feature then a new feature is created and the pixel added to it. In some instances a pixel is found to be connected to two or more features. In these instances the two features are merged.

This produced satisfactory results as was evident from the results in chapter four.

5.2.4 Area and Center of Geometry of Internal Features

The area of internal features is calculated. This is equivalent to the total number of pixels in the feature. The center of area (center of geometry) of the feature is calculated as follows:

$$\bar{x} = \frac{\sum_{y=0}^{y=500} \sum_{x=0}^{x=500} x \cdot P(x,y)}{\text{Area of Feature}}$$

$$\bar{y} = \frac{\sum_{y=0}^{y=500} \sum_{x=0}^{x=500} y \cdot P(x,y)}{\text{Area of Feature}}$$

where P(x,y) has a value of one if an edge point exists at (x,y) and a value of zero if no edge point exists at (x,y).

5.2.5 System Calibration

Since the proposed descriptors are not invariant to scale, some system calibration is required. In the current implementation, a calibration factor is manually entered in the system. The calibration factor, for a given camera height above the object, is determined by the following;

$$\text{Calibration Factor:} = \frac{\text{Dimension of a selected internal feature as determined by the system}}{\text{The actual dimension of the same feature}}$$

In future implementation, it would be useful to have an accurate ranging device to calibrate the system.

5.2.6 Matching of Internal and Extended Features in "Training" and "Runtime" Mode;

The matching of a feature in the "runtime" and "training" modes is accomplished as follows;

(i) The likelihood of the Fourier Descriptors (FD) of two features being equivalent is measured on a scale of 0 - 100. The initial score given to the match is zero.

(ii) The value of the highest coefficient in the FD of the "training" feature is determined.

(ii) A fraction (25%) of the value in (ii) above is defined as a threshold value.

(iii) Coefficients in the "training" FD, that are greater then the threshold value (ie. are significant) are further used in the analysis while the others are discarded.

(iv) Each significant coefficient in the "training" FD is compared to its counterpart in the "runtime" FD. The extent of similarity will determine the number of points awarded to the match.

(v) Combination of features with the highest score (and a score larger than a threshold value) are declared as being matched.

5.2.7 Matching of a Partially Occluded Object With Database Objects;

The matching of a partially occluded object with a candidate object models in the database is accomplished as follow;

> The first feature of the partially occluded object is matched with the first feature of the first candidate model in the database. If no match occurs, then the second feature of the first candidate object is matched. This process continues until all features in the candidate object have been tested or a successful match found. This is repeated for all features in the partially occluded object. Upon completion, it is determined if the required number of extended and internal features have been successfully matched. If so, a tentative match between the partially occluded object and the candidate model is declared. If not, then similar analysis is repeated with another model in the database.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

It has been demonstrated that a technique based upon Fourier Descriptors of extended features can be successfully applied in the recognition of complex industrial objects which are;

(i) partially assembled
(2) overlapping and touching.
(3) dimensionally defective.

Additional advantages of the proposed technique include the following;

(i) No limitation is put on the shape of the complex objects.

(2) The extended feature incorporates structural information on the location of internal features in addition to the shape of the global boundary segment. This greatly facilitates the matching process.

(3) Compared to another Fourier Descriptor based technique in literature, the proposed extended feature method requires less computational resources and will have a smaller database in an equivalent application.

In order to concentrate effort on the principal task, several sections of the system software were simplified or not implemented. This has limited the current implementation to a specific family of objects. In section 6.1, limitations of current implementation and related future work is reviewed. In 6.2 we review the inherent limitations of the current technique and related future work.

6.1 Limitations of Current Implementation and Future Work.

(1) Interface a sensor to automatically determine the distance of an object from the camera and thus to calibrate the system. This is currently accomplished manually.

(2) Incorporate subroutines that will accomplish the following:

(a) Implement the heuristic outlined in section 3.2.1 to accommodate instances where there are more than one boundary points which are the minimum distance from the focus of an internal feature.

(b) Implement heuristic in section 3.2.2.. This will enable objects with less than two internal features to generate extended features.

(3) The three sigma dimensional repeatability in the current implementation is limited to 1.06 pixels. This is because the edge is located to the nearest pixel. Téchniques for sub-pixel edge detection can be utilized to obtain improved repeatabilities.

(4) The system can only accept a maximum of five local internal features. This is a limitation set by the size of the Random Access Memory in the microcomputer. This limitation is easily overcome by the use of virtual memory computers.

(5) The current implementation has a fixed threshold value that is used to separate the object pixels from the background pixels. For flexibility, it is desirable to automatically generate an optimum threshold value for each image.

(6) Develop/adopt a technique to segment the boundaries of overlapping and touching objects. This will improve the matching process and allow a greater degree of occlusion while maintaining ability for recognition.

6.2 Future Work to Improve the Basic Technique.

(1) Improve the technique for the recognition of objects with less two internal features such that it can be applied to any global boundary shape. Integrate this method with the proposed method for objects with more than two internal features.

REFERENCES

[1] H. Blum, "Biological Shape and Visual Science (Part I),"Journal of Theoretical Biology, vol. 38, 1972, pp. 205-286.

[2] H. Blum, "Shape Description Using Weighted Symmetric Axis Features," Pattern Recognition, vol. 10, no. 3, 1978, pp. 167-180.

[3] Bolles, R. C., Cain, R. A., "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," The International Journal of Robotics Research," Vol. 1, No. 3, Fall, 1982.

[5] Duda, R.O., Hart, P.E.,"Use of Hough Transform to Detect Lines and Curves in Pictures," Communications to ACM, vol. 15, no. 1, January, 1972, pp. 11-15.

[6] Faugereras O.D., Fundamentals of Computer Vision, Cambridge University Press, Cambridge, London 1983, pp. 262- 325.

[7] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," IEEE Trans. Electron. Comput., vol. EC-10, June 1961a, pp. 260-268.

[8] H. Freeman, "On the Classification and Recognition of Geometric Patterns," Proceedings of the 3d International Congress on Cybernetics, Namur, Belgium, 1961b, pp. 334-369.

[9] H. Freeman and L.S. Davis, "A Corner-Finding Algorithm for Chain-Coded Curves," IEEE Transactions on Computers, vol C-26, no. 3, March 1977a, pp. 297-303.

[10] H. Freeman, "Shape Description Using Critical Points," Proceedings of IEEE Computer Society Conference on Pattern Recognition and image processing, PRIP-77,Renssalaer Polytechnic Institute, Troy, N.Y. June 6-8, 1977b, pp. 168-174.

[11] H. Freeman, "Lines, Curves and the Characterization of Shape," Report no. IPL-TR-80-004, Image Processing Laboratory, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, N.Y.. March, 1980.

[12] G. H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition," IEEE Trans. Comput., vol. C-21, Feb. 1972, pp. 195-201.

[13] Hough, P.V.C., "Method and Means for Recognizing Complex patterns", U.S. Patent 3,069,654, Dec. 18, 1962.

[14] Hu, M. "Visual Pattern Recognition By Moment Invariants",IRE T. on Inf. Theory, IT-8, pp. 179-187.

[15] Kurozumi, Y., "Polygonal Approximation by the Minimax Method",Computer Graphics and Image Processing, vol. 19, no. 3, July 1982, pp. 284-264.

[16] Kalvin, A., Schonberg, E., Schwartz, J. T., Sharir, M., "Two Dimensional Model-Based Boundary Matching Using Footprints", The International Journal of Robotics Research, vol. 5, no. 4, Winter 1986.

[17] Kiryati, N., Maydan, D., "Calculating Geometric Properties From Fourier Representation", Pattern Recognition, Vol. 22, No. 5, 1989, pp. 469-475.

[18] Levine D. Martin ,Vision in Man And Machine, New York, N.Y., McGraw-Hill Book Company, 1985, pp 480-545.

[19] Lyvers, Edward, P., et al., "Subpixel Measurement Using a Moment-Based Edge Operator," IEEE Transactions on Pattern Analysis and Machine Intelligence," Vol. 11, No. 12, December, 1989.

[20] T. Pavlidis., "Polygonal Approximation by Newton's Method," IEEE Trans. on Computers, vol. C-26, no. 8, August, 1977, pp. 800-807.

[21] W. A. Perkins, "A Model-Based Vision System for Industrial Parts," IEEE Trans. Comput. C-27, 1978, pp. 126-143.

[22] E. Person and K.S. Fu, "Shape discrimation using Fourier Dcriptors of the Bundary Crve," IEEE Trans. Sys. Man Cybern. 7, 1977, pp. 170-179.

[23] Press, W.H., et al., "Numerical Recipies in C", Cambridge University Press, First Edition, New York, 1988, pp. 399-437.

[24] Ramer, U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves," Computer Graphics and Image Processing, vol. 1, no. 3, November 1972, pp. 244-256.

[25] Rohlf, F. J., Archie, J. W., " A Comparison of Fourier Methods for The Description of Wing Shape in Mosquitoes (Diptera: Culicidae)," Sys. Zool., 33(3):, 1984, 302-317.

[26] Weaver, H. J., "Applications of Discrete and Continuous Fourier Analysis", New York, N. Y., Wiley, 1985, pp. 89-110.

[27] C. T. Zahn and R. Z. Roskies, " Fourier descriptors for plane closed curves," IEEE Trans. Comput., vol. C-21, Mar. 1972, pp. 269-281.

Appendix A1: Listing of Interaction with System Software for Adding an Object (serial #2222) to the Database.

```
                    CompuVision

     Automated Inspection System for Industrial Parts
                    Version 1.0



  c         TO CONTINUE WITH CompuVision
  e         TO EXIT CompuVision



     MAKE A SELECTION AND THEN PRESS ENTER

------------------ Screen Cleared ------------------------

                    ******************
                    *  CompuVision   *
                    ******************


  1         CONDUCT OBJECT RECOGNITION
  2         AUTO CALIBRATE THE SYSTEM (NOT IMPLEMENTED)
  3         ADD AN OBJECT TO THE DATABASE
  9         LEAVE CompuVision & EXIT TO DOS

MAKE A SELECTION FROM 0 TO 9 AND THEN PRESS ENTER


------------------- Screen Cleared ------------------------
                 *****************
                 *  CompuVision  *
                 *****************

               System in Learn Mode
               ---------------------


INPUT SERIAL NO. OF OBJ.(INT.<9999) & THEN PRESS ENTER


The following features have been extracted.
```

```
Feature#      # of Edge Points    X - C.of G.  Y - C.of G.
--------      ----------------    -----------  -----------

   0                593              228.4        252.5
   1                211              218.0        211.5
   2                 40              282.1        257.8
   3                 47              172.0        260.2
   4                 35              244.6        318.7

------------------- Screen Cleared -----------------------


              ******************
              *  CompuVision   *
              ******************


Addition of Serial# 2222 to Dbase Completed
----------------------------------------------------


     TYPE 'c' AND THEN ENTER TO CONTINUE


------------------- Screen Cleared -------------------------

                           CompuVision

        Automated Inspection System for Industrial Parts
                          Version 1.0


   c         TO CONTINUE WITH CompuVision
   e         TO EXIT CompuVision

      MAKE A SELECTION AND THEN PRESS ENTER
```

Appendix A2: Listing of Interaction with System Software for the Recognition of an Object (serial #2222) in the Database.

```
                    CompuVision

   Automated Inspection System for Industrial Parts
                    Version 1.0


c         TO CONTINUE WITH CompuVision
e         TO EXIT CompuVision

   MAKE A SELECTION AND THEN PRESS ENTER


------------------- Screen Cleared ------------------------

                    ******************
                    *  CompuVision   *
                    ******************


1         CONDUCT OBJECT RECOGNITION
2         AUTO CALIBRATE THE SYSTEM (NOT IMPLEMENTED)
3         ADD AN OBJECT TO THE DATABASE
9         LEAVE CompuVision & EXIT TO DOS

MAKE A SELECTION FROM 0 TO 9 AND THEN PRESS ENTER    ?


------------------- Screen Cleared ------------------------

              *****************
              *  CompuVision  *
              *****************

         System in Object Recognition Mode
         ---------------------------------
```

The following features have been extracted.

Feature#	# of Edge Points	X - C.of G.	Y - C.of G.
0	592	228.4	252.4
1	213	217.7	211.3
2	41	282.2	257.8
3	47	172.1	260.2
4	35	244.8	318.6

------------------- Screen Cleared ------------------------

```
******************
*  CompuVision   *
******************
```

Object Recognition Now Completed

Based Upon Matching
4 Out of 4 Local Features &
4 Out of 4 Global Boundary Segments
The Current Object Has Been Matched
To Serial #2222 From The Database

TYPE 'm' FOR MORE DETAILS ON MATCHING
TYPE 'c' TO CONTINUE WITHOUT ANY FURTHER DETAILS

------------------- Screen Cleared -----------------------

Details of Matching Current Object to One From the Database
****

Current Object Features		Serial #2222 Features	
Local	Boundary	Local	Boundary
1		1	
2		2	
3		3	
4		4	
	1 - 3		1 - 3
	2 - 1		2 - 1
	3 - 4		3 - 4
	4 - 2		4 - 2

TYPE 'm' FOR SHAPE & DIMENSIONAL ANALYSIS ON CURRENT OBJECT
TYPE 'c' TO CONTINUE WITHOUT FURTHER DETAILS

------------------- Screen Cleared -----------------------

FEATURE#	SHAPE	DIM#1 (mm)	DIM#2 (mm)	POSITION COORDINATES X (mm)	Y (mm)
1	RECTANGLE	29.30	98.15	217.73	211.31
2	RECTANGLE	10.58	17.17	282.20	257.83
3	SQUARE	16.67		172.05	260.17
4	CIRCLE	13.77		244.79	318.62

TYPE 'c' TO CONTINUE

------------------- Screen Cleared ------------------------

CompuVision

Automated Inspection System for Industrial Parts
Version 1.0

c TO CONTINUE WITH CompuVision
e TO EXIT CompuVision

MAKE A SELECTION AND THEN PRESS ENTER

Appendix A3: Database Containing Models of Six (Serial #1111, #2222, #3333, #4444, #5555, #6666) Objects.

```
8888.8                   /* Serial #1111 STARTS HERE  */
1111
3.000000
99999.9
 3.168202 6.979819
33.000001
107.690005
190.921005
346.708008
   1 1095.352  0.000 1095.352
   2 10.313 -0.566 10.328
   3 172.508 30.675 175.214
   4 -29.442 -1.638 29.488
   5  2.543 -1.899  3.174
   6  7.135 11.014 13.123
   7 -24.876  7.100 25.870
   8 12.271  3.570 12.779
   9 -2.156  2.224  3.097
  10 11.388  6.552 13.138
  11  1.906  2.717  3.319
  12  0.806 -0.485  0.940
  13  3.692 -0.395  3.713
  14 -2.477 -0.144  2.481
  15 -1.102 -2.335  2.582
99999.9
 2.342566 7.596000
31.900001
89.540004
165.878006
318.148987
   1 952.906  0.000 952.906
   2 -13.938 18.316 23.016
   3 -229.826 -11.615 230.119
   4 -37.541 27.273 46.402
   5 45.510 -2.348 45.571
   6 -10.114 -22.500 24.668
   7  5.605 19.277 20.075
   8 25.747 -1.609 25.797
   9  4.045 -17.661 18.119
  10 -6.762 -1.874  7.017
  11 -7.895  3.907  8.809
  12 -2.582  0.383  2.610
  13 -0.958  6.000  6.076
  14  1.189  6.348  6.458
  15  3.944  0.178  3.948
```

```
99999.9
 3.821396 8.770762
42.900001
147.620006
223.451004
314.614990
   1 1270.140  0.000 1270.140
   2  5.540  0.554  5.568
   3 -267.172  8.081 267.294
   4  3.594 -2.984  4.671
   5 30.120  3.964 30.380
   6 -6.450  5.873  8.723
   7 29.563 -9.061 30.920
   8  3.306 -6.325  7.137
   9 -25.449  5.151 25.965
  10  1.699  3.584  3.967
  11  8.140  1.657  8.307
  12 -4.435  0.439  4.457
  13  1.964 -5.171  5.531
  14  3.504 -3.024  4.629
  15 -2.871  3.717  4.696
99999.9
 2 34.522414 55.151835
150.700003
165.770007
194.664505
316.381989
   1 9072.965  0.000 9072.965
   2 -10.718  0.132 10.719
   3 593.705 -14.575 593.884
   4 35.227 -1.297 35.251
   5 -553.129 27.174 553.796
   6 -29.353  1.803 29.408
   7 -254.024 18.738 254.714
   8 -28.887  2.488 28.994
   9 122.510 -12.066 123.103
  10 -1.284  0.142  1.292
  11 111.393 -13.739 112.237
  12 32.660 -4.436 32.960
  13 -34.217  5.076 34.592
  14 11.326 -1.822 11.472
  15 -59.009 10.239 59.891
99999.9
 3 23.734845 38.306889
88.000002
96.800004
178.399506
332.428497
   1 6183.684  0.000 6183.684
   2 -365.420  4.485 365.447
   3 -172.277  4.229 172.329
```

```
   4 169.251 -6.234 169.366
   5 299.489 -14.713 299.850
   6 -132.458  8.138 132.708
   7 -20.299  1.497 20.354
   8 66.932 -5.764 67.179
   9 47.451 -4.674 47.681
  10 -42.079  4.666 42.337
  11 23.486 -2.897 23.664
  12 35.346 -4.801 35.671
  13 15.363 -2.279 15.531
  14 -23.492  3.780 23.794
  15  5.794 -1.005  5.881
99999.9
 1 28.398515 41.354859
89.100002
98.010004
207.186005
330.661499
   1 6832.933  0.000 6832.933
   2 76.087 -0.934 76.093
   3 -313.513  7.696 313.607
   4 -135.189  4.979 135.281
   5 311.074 -15.282 311.449
   6 43.564 -2.676 43.646
   7 -27.641  2.039 27.716
   8 -63.122  5.436 63.356
   9 65.818 -6.482 66.136
  10 43.914 -4.870 44.184
  11 -7.352  0.907  7.408
  12 -34.574  4.696 34.892
  13 26.152 -3.879 26.439
  14 18.178 -2.925 18.412
  15  7.668 -1.331  7.783

8888.8                     /* Serial #2222 STARTS HERE  */
2222
4.000000
99999.9
 6.196518 7.197237
37.400001
162.140007
252.403000
325.910004
   1 1418.991  0.000 1418.991
   2 -0.689  0.529  0.868
   3  2.695  0.269  2.708
   4 -10.565 -2.606 10.882
   5  3.325  0.352  3.344
   6 -8.179 -5.458  9.833
   7  3.820 -0.946  3.935
   8 -2.129 10.471 10.685
```

```
   9  4.467  0.517  4.497
  10 -2.663 -7.770  8.214
  11 11.336 -3.535 11.874
  12  2.511 -2.472  3.523
  13  6.321 -3.293  7.128
  14  5.515  2.490  6.051
  15  7.485 -2.517  7.897
99999.9
 6.834794 9.824038
51.700001
238.370010
178.106995
269.136993
   1 1720.519  0.000 1720.519
   2 -0.984  3.704  3.832
   3 -1.098 12.171 12.220
   4 -6.332 -0.155  6.334
   5 -59.029 90.480 108.032
   6 11.919  3.197 12.341
   7 -10.400  1.054 10.453
   8 -1.453 -8.053  8.183
   9  4.316 -23.684 24.074
  10 -12.955  1.954 13.102
  11  7.486  0.807  7.529
  12  3.782 -9.573 10.293
  13 12.929 -20.560 24.287
  14 -1.088 -6.760  6.847
  15  3.874  3.364  5.131
99999.9
 5.247662 8.617145
45.100001
182.710008
288.231995
263.894012
   1 1481.262  0.000 1481.262
   2 -6.655  1.669  6.861
   3 -138.112  9.739 138.455
   4 -24.867 28.974 38.182
   5 -44.818 -15.511 47.426
   6 -2.774 -9.792 10.177
   7 29.994  5.917 30.572
   8  2.177 -21.861 21.969
   9  7.202 10.844 13.018
  10  7.692 10.558 13.062
  11 -11.487 -5.582 12.772
  12  2.548  3.679  4.475
  13 -3.795  1.439  4.058
  14 -8.018  0.234  8.021
  15  1.451 -2.403  2.807
```

```
99999.9
 14.542249 48.675521
231.000005
2831.400123
222.658997
219.270004
   1 5613.725  0.000 5613.725
   2 -9.251 -8.923 12.853
   3 1739.470 27.776 1739.692
   4 22.768  1.559 22.821
   5 632.248 22.568 632.651
   6 -12.349  0.990 12.389
   7 109.329 17.820 110.771
   8 -5.372 22.677 23.304
   9 -104.668 -8.035 104.976
  10 12.188  9.128 15.228
  11 -165.378 -19.108 166.479
  12 16.031  5.564 16.969
  13 -119.094 -14.048 119.920
  14  7.355 -0.319  7.362
  15 -56.761 -14.832 58.667
99999.9
 3 49.727316 106.035581
200.200004
220.220010
200.382996
244.203491
   1 13287.570  0.000 13287.570
   2 -563.863  6.920 563.906
   3 -1618.278 39.726 1618.765
   4 848.537 -31.253 849.113
   5 713.371 -35.046 714.231
   6 -475.633 29.221 476.529
   7 -113.035  8.338 113.342
   8 392.709 -33.818 394.162
   9 20.655 -2.034 20.755
  10 -201.710 22.369 202.946
  11 112.814 -13.914 113.668
  12 142.739 -19.386 144.049
  13 -88.911 13.189 89.884
  14 -42.454  6.831 43.000
  15 117.633 -20.411 119.391
99999.9
 1 47.917555 92.387489
194.700004
214.170009
255.445496
241.582001
   1 13260.648  0.000 13260.648
   2 1342.179 -16.472 1342.280
   3 -327.830  8.048 327.928
```

```
   4 -1027.808 37.856 1028.505
   5 120.974 -5.943 121.120
   6 317.174 -19.486 317.772
   7 310.185 -22.881 311.028
   8 -105.788  9.110 106.180
   9 -118.643 11.685 119.217
  10 -74.107  8.218 74.561
  11 115.975 -14.304 116.854
  12 84.095 -11.422 84.868
  13 21.585 -3.202 21.821
  14 -91.732 14.760 92.912
  15 -29.366  5.095 29.805
99999.9
 4 34.227556 60.953250
121.000003
133.100006
215.255005
297.523499
   1 9240.300  0.000 9240.300
   2 -503.843  6.183 503.881
   3 782.447 -19.208 782.682
   4 362.136 -13.338 362.382
   5 273.572 -13.440 273.902
   6 -7.729  0.475  7.744
   7 276.350 -20.385 277.101
   8 10.740 -0.925 10.779
   9 86.221 -8.492 86.638
  10 42.855 -4.753 43.118
  11 78.406 -9.670 79.000
  12 -9.272  1.259  9.357
  13 74.004 -10.977 74.813
  14 20.876 -3.359 21.144
  15 17.647 -3.062 17.911
99999.9
 2 40.351799 62.363402
136.400003
150.040007
270.317505
294.902008
   1 10286.330  0.000 10286.330
   2 441.249 -5.415 441.282
   3 -232.066  5.697 232.136
   4 -488.379 17.988 488.710
   5 395.869 -19.448 396.346
   6 140.392 -8.625 140.657
   7 80.628 -5.948 80.847
   8 -155.270 13.371 155.845
   9 26.224 -2.583 26.351
  10 55.142 -6.115 55.480
  11 73.918 -9.117 74.478
  12 -40.694  5.527 41.068
```

```
  13 -16.373  2.429 16.552
  14 -1.710  0.275  1.732
  15 51.442 -8.926 52.210
8888.8                      /* Serial #3333 STARTS HERE  */
3333
4.000000
99999.9
 6.455822 7.480531
38.500001
174.240008
214.604004
269.639008
   1 1475.304  0.000 1475.304
   2 -0.410  0.274  0.493
   3  5.857 16.361 17.378
   4 -13.238 -3.587 13.715
   5 18.504 -6.866 19.737
   6 -1.299  6.568  6.695
   7 -0.677 -3.030  3.105
   8  1.604  4.309  4.598
   9 10.401 -6.259 12.138
  10  5.024  1.095  5.142
  11 -2.807  4.586  5.377
  12 -3.130 -4.346  5.356
  13 -2.164  0.672  2.266
  14  1.239  1.329  1.817
  15 -2.521  1.197  2.791
99999.9
 6.027118 12.890066
62.700001
313.390014
280.528992
264.212006
   1 1920.879  0.000 1920.879
   2 -2.210 -1.547  2.698
   3 -358.996 21.954 359.666
   4 -4.763  4.508  6.558
   5  8.390 -8.361 11.845
   6  7.933 -7.933 11.219
   7 61.482 -0.269 61.483
   8 -7.871  8.809 11.813
   9 -38.926 -0.332 38.928
  10  5.893 -6.270  8.604
  11  7.406  3.847  8.346
  12 -3.510  2.200  4.142
  13  4.728 -4.098  6.257
  14  1.320  0.568  1.437
  15 -1.315  0.451  1.390
```

```
99999.9
 7.684454 11.059299
60.500001
302.500013
276.980011
178.916000
   1 1945.407   0.000 1945.407
   2  1.967 -2.188  2.942
   3 -75.308 13.154 76.448
   4  4.236  9.458 10.364
   5 -120.428 11.839 121.008
   6 -3.186 -3.824  4.977
   7 22.930 -11.235 25.535
   8 -6.141 -5.922  8.531
   9 28.097 -5.659 28.661
  10  4.463  2.578  5.154
  11 -6.561  6.699  9.377
  12  6.659  2.443  7.093
  13 -7.271  2.463  7.677
  14 -4.231 -0.211  4.236
  15 -0.032 -1.834  1.834
99999.9
 10.492783 11.907589
63.800001
427.130019
211.968994
177.699997
   1 2361.985   0.000 2361.985
   2  0.015 -1.001  1.001
   3 12.650  7.020 14.467
   4 -7.831 13.908 15.962
   5 -8.592 20.690 22.403
   6  3.392 -3.890  5.161
   7 -5.688 -4.128  7.028
   8  9.277  2.263  9.549
   9 -3.176 -2.147  3.833
  10  5.321 -6.822  8.652
  11 -9.569 -2.090  9.795
  12 -0.988  8.766  8.821
  13 -6.329 -3.927  7.448
  14  2.723 -9.561  9.941
  15 -0.649  1.743  1.859
99999.9
 4 68.566033 104.236055
280.500006
308.550013
213.286499
223.669495
   1 18778.424  0.000 18778.424
   2 683.448 -8.388 683.500
   3 -1356.996 33.312 1357.405
```

```
    4 -95.311   3.511 95.376
    5 -373.162 18.332 373.612
    6 -263.073 16.162 263.569
    7 -112.821  8.322 113.127
    8 76.700 -6.605 76.984
    9 129.890 -12.793 130.518
   10 108.294 -12.010 108.958
   11 50.684 -6.251 51.068
   12 -25.350   3.443 25.582
   13 -48.106   7.136 48.632
   14 -64.136 10.319 64.960
   15 -18.271   3.170 18.544
99999.9
 1 27.843956 44.620862
71.500002
78.650003
244.474503
178.307999
    1 6930.048   0.000 6930.048
    2 -33.775   0.414 33.777
    3 671.311 -16.480 671.513
    4   5.618 -0.207   5.622
    5 216.871 -10.654 217.133
    6 12.307 -0.756 12.330
    7 98.533 -7.268 98.801
    8 11.995 -1.033 12.040
    9 54.266 -5.345 54.529
   10   9.202 -1.020   9.258
   11 34.968 -4.313 35.233
   12   5.551 -0.754   5.602
   13 25.996 -3.856 26.280
   14   1.906 -0.307   1.931
   15 21.624 -3.752 21.948
99999.9
 2 69.906914 93.063768
236.500005
260.150011
278.754517
221.563995
    1 17556.408   0.000 17556.408
    2 -61.396   0.753 61.401
    3 -984.233 24.162 984.530
    4 -205.864   7.582 206.003
    5 -11.835   0.581 11.849
    6 -44.491   2.733 44.575
    7 -36.652   2.704 36.752
    8   3.132 -0.270   3.143
    9 -43.144   4.249 43.352
   10 52.801 -5.856 53.125
   11 -41.714   5.145 42.030
   12 14.357 -1.950 14.489
```

```
  13  2.027 -0.301  2.049
  14 -5.229  0.841  5.297
  15  9.576 -1.662  9.719
99999.9
 3 20.987770 41.807038
64.900001
71.390003
247.566498
266.925507
   1 5574.777  0.000 5574.777
   2 -53.565  0.657 53.569
   3 765.201 -18.785 765.431
   4 -16.509  0.608 16.520
   5 285.963 -14.048 286.308
   6  6.098 -0.375  6.110
   7 125.379 -9.248 125.719
   8 20.269 -1.745 20.344
   9 72.700 -7.160 73.051
  10  1.414 -0.157  1.422
  11 63.423 -7.822 63.903
  12 -6.932  0.941  6.996
  13 41.850 -6.208 42.308
  14  2.738 -0.440  2.773
  15 28.060 -4.869 28.479
8888.8                   /* Serial #4444 STARTS HERE  */
4444
2.000000
99999.9
 10.389254 11.487190
61.600001
416.240018
269.993988
291.063995
   1 2333.179  0.000 2333.179
   2  0.165 -0.363  0.398
   3  7.776 -1.523  7.924
   4 -8.785  0.012  8.785
   5  4.500 -3.548  5.731
   6  5.296 -1.150  5.420
   7 -6.320 -6.053  8.751
   8 -0.381 -7.997  8.006
   9  1.862  2.805  3.367
  10  1.403  1.321  1.927
  11 -5.408 -1.613  5.644
  12  5.305 -0.950  5.390
  13  2.786 -0.954  2.945
  14  1.680  1.454  2.222
  15  1.580  8.308  8.456
```

```
99999.9
 5.416258 8.106484
40.700001
164.560007
191.095993
281.705994
   1 1416.541  0.000 1416.541
   2 -3.525  2.561  4.357
   3 -7.590 30.973 31.889
   4 -13.846  9.872 17.005
   5 -44.680 61.163 75.744
   6 11.409  6.595 13.178
   7 -1.924  4.648  5.030
   8 -19.175 -2.699 19.364
   9 -12.012  9.145 15.097
  10 -4.469 -1.547  4.730
  11  5.026  1.030  5.130
  12 -0.041 -1.191  1.192
  13  6.757 -1.842  7.004
  14 -0.043  1.124  1.124
  15  0.813 -2.648  2.770
99999.9
 2 28.475945 101.987514
213.400005
234.740010
230.544983
286.385010
   1 10756.105  0.000 10756.105
   2 -2580.015 31.663 2580.210
   3 501.528 -12.312 501.679
   4 1092.859 -40.252 1093.600
   5 -274.831 13.502 275.163
   6 197.452 -12.131 197.824
   7 -54.592  4.027 54.740
   8 533.617 -45.952 535.592
   9 -505.797 49.817 508.244
  10 463.653 -51.418 466.496
  11 -170.353 21.011 171.644
  12 182.008 -24.720 183.679
  13 -157.334 23.338 159.055
  14 221.353 -35.616 224.200
  15 -72.260 12.538 73.340
99999.9
 1 23.529218 104.310084
216.700005
238.370010
230.544983
286.385010
   1 8312.960  0.000 8312.960
   2 -1658.153 20.350 1658.278
   3 1869.238 -45.887 1869.801
```

```
    4 -64.216  2.365 64.259
    5 135.893 -6.676 136.057
    6 619.444 -38.056 620.612
    7 -439.371 32.410 440.565
    8 708.622 -61.023 711.244
    9 -440.962 43.431 443.095
   10 481.127 -53.356 484.076
   11 -205.610 25.360 207.168
   12 176.650 -23.992 178.272
   13 37.895 -5.621 38.310
   14 -46.045  7.409 46.638
   15 169.451 -29.402 171.983
8888.8                    /* Serial #5555 STARTS HERE  */
5555
2.000000
99999.9
 2.689368 22.308792
89.100002
249.260011
224.649994
337.583008
    1 1292.658  0.000 1292.658
    2 -30.673  3.730 30.899
    3 558.894 -72.080 563.523
    4 -9.915 -16.975 19.658
    5 348.338 -101.550 362.838
    6 13.665 -17.510 22.211
    7 239.418 -101.468 260.032
    8 24.770 -5.438 25.360
    9 164.981 -87.900 186.936
   10 22.365  5.433 23.016
   11 109.951 -73.574 132.297
   12 14.955  9.222 17.569
   13 72.359 -60.920 94.589
   14  8.277  8.714 12.019
   15 48.169 -47.177 67.423
99999.9
 8.928636 9.991565
53.900001
309.760013
222.483994
289.222992
    1 1996.426  0.000 1996.426
    2 -1.454 -0.015  1.454
    3  6.006  7.051  9.262
    4 -5.439  4.944  7.350
    5  1.007 -2.749  2.927
    6  1.228 -12.695 12.754
    7 -1.493 -10.301 10.409
    8 12.983 12.007 17.684
    9 -1.755 -1.203  2.127
```

```
  10  0.378 -9.992  9.999
  11 -5.887  6.860  9.040
  12 -5.920  1.712  6.163
  13 -8.405  6.250 10.474
  14 -0.785  1.949  2.101
  15 -1.283  3.439  3.671
99999.9
 2 34.761245 108.740321
279.400006
307.340013
223.566986
313.403015
   1 13934.974  0.000 13934.974
   2 1190.537 -14.611 1190.627
   3 -3517.225 86.343 3518.285
   4 -827.558 30.481 828.119
   5 748.687 -36.781 749.590
   6 54.982 -3.378 55.086
   7 133.025 -9.813 133.387
   8 310.617 -26.749 311.767
   9 -207.897 20.476 208.903
  10 -386.341 42.844 388.709
  11 91.504 -11.286 92.197
  12 242.859 -32.984 245.089
  13 -4.780  0.709  4.833
  14 -29.176  4.694 29.551
  15 12.936 -2.245 13.130
99999.9
 1 34.844585 84.396216
201.300004
221.430010
223.566986
313.403015
   1 11492.409  0.000 11492.409
   2 -543.154  6.666 543.195
   3 -1597.776 39.223 1598.257
   4 285.765 -10.525 285.958
   5 -252.203 12.390 252.508
   6 -376.033 23.102 376.742
   7 431.634 -31.839 432.806
   8 156.777 -13.501 157.357
   9 -129.555 12.760 130.181
  10 61.700 -6.842 62.078
  11 -26.172  3.228 26.370
  12 -171.882 23.344 173.460
  13 57.667 -8.554 58.298
  14 66.863 -10.758 67.723
  15 -42.052  7.297 42.681
```

```
8888.8                    /* Serial #6666 STARTS HERE  */
6666
3.000000
99999.9
 5.374023 7.799167
40.700001
158.510007
282.794006
274.450012
   1 1385.759  0.000 1385.759
   2 -0.257 -3.635  3.644
   3 -14.493 26.628 30.317
   4 -1.759 -2.342  2.929
   5 -72.254 24.083 76.162
   6 -7.619 -11.206 13.551
   7  0.358 -4.521  4.535
   8 -6.040 -4.230  7.374
   9  5.333 -11.302 12.497
  10  4.588  7.561  8.844
  11  0.049 -7.757  7.757
  12  7.555  6.406  9.905
  13  4.740 -6.955  8.416
  14  1.983  3.893  4.369
  15  0.571 -1.541  1.644
99999.9
 6.035183 10.338518
52.800001
232.320010
204.744995
272.479004
   1 1677.817  0.000 1677.817
   2 -3.315 -0.771  3.403
   3 166.705 14.304 167.318
   4 -5.983  6.635  8.934
   5 -72.796 32.173 79.589
   6 -6.213 10.718 12.389
   7 -53.283 17.369 56.043
   8 -1.359  9.108  9.209
   9 -6.565 -3.990  7.682
  10  3.222  3.773  4.961
  11  6.223 -8.488 10.525
  12  1.822  0.236  1.837
  13  1.020 -0.984  1.417
  14 -3.860  1.069  4.006
  15 -1.105  2.858  3.064
```

```
99999.9
 6.445482 7.478973
38.500001
173.030008
252.593994
218.684998
   1 1469.466  0.000 1469.466
   2 -0.695 -0.020  0.695
   3 10.926 13.408 17.296
   4 -9.393  0.874  9.434
   5 15.470 -2.348 15.647
   6 -6.268  5.188  8.137
   7 -0.692 -8.317  8.346
   8  6.765  3.520  7.626
   9 12.476 -1.587 12.576
  10  1.565  4.356  4.629
  11 -6.314  2.184  6.681
  12 -1.928 -8.567  8.782
  13  1.911  0.895  2.111
  14  1.813  4.832  5.161
  15 -5.387  2.575  5.971
99999.9
 2 51.357359 73.353348
160.600003
176.660008
228.669495
245.582001
   1 12228.570  0.000 12228.570
   2 182.778 -2.243 182.792
   3 -470.819 11.558 470.961
   4 -214.140  7.887 214.286
   5 529.049 -25.990 529.687
   6 102.816 -6.317 103.010
   7 -63.911  4.714 64.085
   8 -85.835  7.392 86.153
   9 112.469 -11.077 113.013
  10 71.747 -7.957 72.187
  11 -4.134  0.510  4.165
  12 -57.041  7.747 57.564
  13 38.029 -5.641 38.445
  14 41.250 -6.637 41.780
  15 16.789 -2.913 17.040
99999.9
 3 58.704885 69.570617
178.200004
196.020008
243.769501
273.464508
   1 13256.193  0.000 13256.193
   2 495.172 -6.077 495.210
   3 189.188 -4.644 189.245
```

```
    4 21.445 -0.790 21.460
    5 19.763 -0.971 19.787
    6 -21.049  1.293 21.089
    7 -10.784  0.795 10.813
    8  9.966 -0.858 10.003
    9 -4.510  0.444  4.532
   10 -9.357  1.038  9.414
   11  5.126 -0.632  5.164
   12 -1.334  0.181  1.347
   13  0.702 -0.104  0.710
   14 -0.589  0.095  0.596
   15 -4.593  0.797  4.662
99999.9
 1 34.440360 65.759200
138.600003
152.460007
267.694000
246.567505
    1 9846.094  0.000 9846.094
    2 -1352.846 16.603 1352.948
    3 182.602 -4.483 182.657
    4 628.575 -23.152 629.001
    5 -85.531  4.202 85.634
    6 -89.293  5.486 89.461
    7 267.337 -19.720 268.063
    8 -53.642  4.619 53.841
    9 -21.279  2.096 21.382
   10 105.958 -11.750 106.607
   11 12.443 -1.535 12.537
   12 -25.649  3.484 25.885
   13 66.548 -9.871 67.276
   14 -2.104  0.339  2.131
   15 -3.471  0.602  3.523
```

Appendix A4: Object Model for Serial #2222 Obtained During "Runtime" Mode.

```
8888.8
4.000000
99999.9
 6.339738 7.426057
38.500001
169.400007
244.785995
318.621002
   1 1451.920   0.000 1451.920
   2   0.403   0.979   1.059
   3 -12.764 13.413  18.516
   4   0.873  -2.112   2.285
   5  11.141  -6.914  13.112
   6   0.656   4.599   4.646
   7  -0.058   1.725   1.726
   8   0.941  -3.269   3.401
   9  13.044   1.298  13.109
  10   1.282  -4.674   4.847
  11  -2.126   8.578   8.837
  12  -4.962  -6.685   8.325
  13   1.471   0.957   1.755
  14  -3.569   4.001   5.362
  15   2.139  -3.577   4.168
99999.9
 6.816049 9.853131
51.700001
242.000010
172.054993
260.170013
   1 1735.390   0.000 1735.390
   2  -1.705   3.130   3.564
   3  -0.471  12.338  12.347
   4 -15.504  -4.372  16.109
   5 -46.352  99.600 109.858
   6   4.725   3.911   6.134
   7  -8.357 -12.241  14.821
   8   1.649   1.007   1.932
   9  -9.227 -21.592  23.481
  10   0.520   0.577   0.776
  11   7.079  -1.023   7.152
  12   2.703  -6.782   7.301
  13   5.702 -27.525  28.109
  14   6.361   3.886   7.454
  15   3.580  -1.360   3.830
99999.9
 5.288546 8.584725
45.100001
182.710008
282.199005
257.828003
```

```
   1 1465.833  0.000 1465.833
   2 -4.748 14.637 15.388
   3 -122.938 23.259 125.119
   4 -18.862 -6.885 20.080
   5 -56.845  0.029 56.845
   6 -10.427  7.714 12.970
   7 37.902 -22.118 43.883
   8  6.349 -1.205  6.462
   9 -0.762  7.577  7.615
  10  9.887 -4.677 10.937
  11 -2.493 12.012 12.268
  12 -5.335 -8.473 10.012
  13 -9.762  3.983 10.544
  14  2.192 12.437 12.629
  15  2.283 -13.081 13.279
99999.9
 14.651460 49.076243
234.300005
2835.030123
217.725998
211.313995
   1 5635.816  0.000 5635.816
   2 -17.887 -16.435 24.292
   3 1716.657 -53.972 1717.505
   4 11.870 15.588 19.592
   5 606.462 -29.185 607.164
   6 33.650 10.510 35.253
   7 113.765 -9.892 114.195
   8 25.328 -5.217 25.860
   9 -102.663  9.409 103.093
  10  3.837 -6.247  7.331
  11 -158.135 28.693 160.717
  12 -7.158  1.630  7.342
  13 -123.377 32.202 127.510
  14 -7.975  2.754  8.437
  15 -61.403 16.780 63.655
99999.9
 3 50.321728 105.956509
205.700004
226.270010
194.890503
235.742004
   1 13273.181  0.000 13273.181
   2 -461.043  5.658 461.077
   3 -1628.945 39.988 1629.436
   4 713.116 -26.266 713.600
   5 812.181 -39.900 813.161
   6 -433.932 26.659 434.750
   7 -217.012 16.008 217.601
   8 358.871 -30.904 360.199
   9 126.556 -12.465 127.169
```

```
  10 -238.710 26.472 240.174
  11 21.352 -2.633 21.513
  12 184.211 -25.019 185.902
  13 -29.497  4.375 29.819
  14 -111.386 17.922 112.819
  15 71.954 -12.485 73.029
99999.9
 1 47.902902 92.597633
191.400004
210.540009
249.962494
234.570999
   1 13258.482  0.000 13258.482
   2 1349.872 -16.566 1349.973
   3 -298.574  7.330 298.664
   4 -1052.294 38.758 1053.008
   5 73.689 -3.620 73.778
   6 280.424 -17.228 280.953
   7 354.870 -26.177 355.834
   8 -88.737  7.642 89.065
   9 -115.892 11.414 116.453
  10 -101.533 11.260 102.156
  11 108.374 -13.367 109.195
  12 79.911 -10.853 80.645
  13 40.568 -6.018 41.012
  14 -71.359 11.482 72.277
  15 -44.492  7.720 45.157
99999.9
 4 34.451545 61.768721
122.100003
134.310006
208.420502
289.395508
   1 9234.241  0.000 9234.241
   2 -579.836  7.116 579.880
   3 747.071 -18.340 747.296
   4 301.191 -11.094 301.395
   5 304.202 -14.944 304.569
   6 -38.184  2.346 38.256
   7 240.626 -17.750 241.280
   8 35.872 -3.089 36.005
   9 56.818 -5.596 57.093
  10 40.281 -4.467 40.528
  11 94.266 -11.627 94.980
  12 -20.003  2.717 20.187
  13 63.670 -9.445 64.366
  14 24.606 -3.959 24.922
  15 16.932 -2.938 17.185
```

```
99999.9
 2 40.162224 62.143753
136.400003
150.040007
263.492493
288.224487
   1 10281.327  0.000 10281.327
   2 433.576 -5.321 433.609
   3 -191.541  4.702 191.599
   4 -539.656 19.877 540.022
   5 352.398 -17.312 352.823
   6 130.091 -7.992 130.336
   7 122.226 -9.016 122.558
   8 -140.418 12.092 140.938
   9  8.900 -0.877  8.943
  10 53.800 -5.966 54.130
  11 93.052 -11.477 93.757
  12 -30.856  4.191 31.139
  13 -22.960  3.406 23.212
  14 -19.815  3.188 20.069
  15 46.830 -8.126 47.530
```