IMAGE DATA ACQUISITION, PROCESSING AND OBJECT RECOGNITION FOR USE IN A MACHINE VISION SYSTEM

by

BIZHAN NIKRAVAN

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy

The University of Leeds

Department of Mechanical Engineering

August 1987

This thesis is dedicated to my dear parents for their unfailing support and encouragement.

A B S T R A C T

Since the early pioneering work of Rosenfeld and Sklansky, machine vision has been used successfully in a number of practical applications. In this work the areas of application have been reviewed to determine the extent to which computer vision has been utilised in industry. Its application so far has been restricted principally to the mass production field of car manufacture, in the areas of body panel inspection and to a lesser degree to unit assembly.

The development of robots for picking unsorted components from containers and for inspection depends upon the reliability and the speed of the vision control system employed.

This project investigates a preliminary approach for fast shape recognition of randomly positioned and oriented components. The results obtained indicate that this goal can be achieved by the use of the techniques of contour tracing, Fourier and surface texture analysis and shape from shading methods.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEGEMENTS

The author wishes to thank all those people whose assistance has made this work possible. Special thanks are due to Dr. K.F. Gill, as supervisor, for his guidance and encouragement during the course of this project and his patient and valuable assistance in its presentation. Thanks are extended to Dr. R.M. Baul for a number of enlighting discussions and his valued assistance in various stages of the research.

In addition, I also wish to extend my gratitude to Professor B.N. Cole and Professor D. Dowson, past and present Heads of the Department of Mechanical Engineering, for allowing the work to be carried out in their department.

I am grateful to the systems staff of the departmental VAX 8600 computer for their assistance in overcoming the many programming problems encountered throughout the project.

For the thesis as presented, thanks must go to Miss E.H. Watson for her tireless and patient efforts in typing the manuscript, and to Mrs. J. Lowe for the help so freely given during the preparation of the thesis.

Finally, the author wishes to express his appreciation to his wife, Fariba, for her patience and assistance throughout the completion of this work.

DECLARATION OF ORIGINALITY

The work presented is concerned with the acquisition and processing of data obtained from a single camera image. Previous work on edge detection has been adopted and applied to the enhancement of engineering component boundaries; it has been shown that the edge detection methods available are adequate for this work.

In the application of the Freeman's code, a look-up table has been introduced to accelerate the contour mapping process and successfully applied to engineering component identification.

To eliminate a fixed geometry between camera and component for robotic use, Fourier Descriptors, texture analysis and shading models have been introduced into the identification process.

The adaption of these methods in the field of engineering has made a novel contribution to the literature.

GENERAL LAYOUT AND PRESENTATION

This thesis is presented in accordance with the British Standard recommendations for the Presentation of Thesis; BS4821: 1972, and with the regulations of the University of Leeds.

All symbols and nomenclature, wherever possible, are in accordance with general and consistent usage and they are fully defined at their first appearance. A list of nomenclature is also included for ease of reference.

Equations, figures and tables are designated by the chapter in which they occur and by a secondary number denoting the order in which they appear in the chapter. Figures and table are bound into the thesis near the appropriate text.

The references cited for the present work are listed at the end of the thesis and they are numbered according to the order that they are mentioned in every chapter.

Pages are numbered consecutively through the thesis including figures and tables.

NOMENCLATURE

A list of the principal symbols used within the thesis is given and is intended to supplement the definitions given within the text. When a symbol occurs and is used to represent different variables in different parts of the thesis, all definitions are given and the reader should decide from the context which definition is relevant.

List of Symbols

$X^n(l)$	Parametric representation of curves, arc length 'l', dimensions 'n'.
a_n, b_n	Fourier Coefficients.
L	Total length of a trajectory.
r, θ	Polar coordinates.
δ	Pixel displacement.
i,j	Pixel (2-dimensional) array number.
X,Y,Z	Three-dimensional space.
D1,D2	Rate of change of grey levels in horizontal and vertical directions.
IP(X,Y)	Pixel value in the digitised image, location X (row), and Y (column).
B	Data in local (image) region.
T	Mask data (orthogonal base masks).
θ	Projected angle on the edge subspace.
BP(IX,IY)	Edge magnitude.
L	Sequence of curvature elements.
F	Shifted sequence of curvature elements.
d(j)	Measured similarity at curvature element j.

i	Contour interval.
d	Difference in corresponding curvature elements.
suma	Theoretical sum of the differences (in curvature element values)
RTHRS	Rejection threshold.
dsum	Measured sum of the differences (in curvature element values).
x(m),y(m)	Cartesian coordinates (boundary pixels).
ω_o	Natural frequency.
X(n),Y(n)	Complex Fourier coefficient.
L	Total number of pixels in the boundary.
R(n)	Orientation independant descriptor.
S(n)	Normalised descriptor.
Fr_n	Reference feature vector.
Sr_n	Test feature vector.
K	Number of elements in feature vectors.
E	Similarity measure between two feature vectors.
a_i	Chain codes.
C	Series of chain codes.
k	Number of elements in a chain code series.
t	Perimeter length of a contour.
Δt	Length of code vectors.
$\Delta x, \Delta y$	Changes in x,y coordinates.
x_q, y_q	Projection on the x and y axis of the first q elements in a chain code series.
$\dot{x}(t), \dot{y}(t)$	Derivative of x and y.
a_n, b_n	Fourier coefficients.
$\hat{X}_N(t), \hat{y}_N(t)$	Fourier series truncated after N terms.

ΔE_a	Actual error.
ΔE_p	Predicted error.
$V_o^T(\dot{x}(t))$	Total variation of the time derivatives $\dot{x}(t)$.
$V_o^T(\dot{y}(t)$	Total variation of the time derivatives $\dot{y}(t)$.
t	Time.
T	Time period.
$V\{v_1,..v_m\}$	Feature vector of m feature elements.
PS(i,j)	Power spectrum.
AR(i,j)	Real part of the Fourier transformed image.
AI(i,j)	Imaginary part of the Fourier transformed image.
a(x,y)	Grey level at point (x,y).
K	Number of pixels in a side length for averaging.
$A_K(x,y)$	Segmented image.
D	Measured difference in feature vectors.
D_{max}	Maximum of D's.
V_r	Reference object feature vector.
V_t	Test object feature vector.
mV_r	Arithmetic mean of reference feature vector.
mV_t	Arithmetic mean of test feature vector.
$\underline{i},\underline{j},\underline{k}$,	Unit vectors in the x,y, and z direction.
$\underline{n}$	Normal vector.
f	Function describing a plane surface.
$\underline{l}$	Light source vector.
$\underline{r}$	Reflection vector.
$\underline{V}$	View vector.

I	Intensity.
m	Shininess factor.
ϕ	The angle between the view and reflection vectors.
c	Ambient light term.
a ,b	Intensity function parameters.
α	Angle between the light source vector and the horizontal surface.
β	Angle between the normal vector direction and the horizontal surface.
γ	Angle between the view vector and the horizontal surface.
ψ	Angle between the reflection vector and the horizontal surface.
ρ	Correlation factor.

List of Abbreviations

CPU	Central Processing Unit.
LSI	Large Scale Integration.
AI	Artificial Intelligence.
3D	Three-dimensional.
2D	Two-dimensional.
$2\frac{1}{2}$D	Two and a half dimensional.
NFD	Normalised Fourier Descriptors.
sfs	shape from shading.
GCA	Grid Circle Analyser.
CMU	Carnegie-Mellon University.
ALV	Autonomous Land Vehicle.
DEC	Digital Equipment Corporation.
OMMS	Optional Memory Management System.
RAM	Random Access Memory.

DMA	Direct Memory Access.
ms	mili-seconds.
Hz	Hertz.
ABS	Absolute.
ORD	Ordinal.
sgn	sign.
max	maximum.

C H A P T E R O N E

INTRODUCTION

An Introduction to Image Processing and Computer Vision

is given together with the Research Objectives

1.1 INTRODUCTION

Visual data constitutes practically all the estimated 'one thousand million' bits of information received by human sensory receptors every second [1]. Most of the information humans receive, learn, remember and use is in graphic form. A large area of the brain's cortex is dedicated to vision and this is one explanation for the extraordinary capacity that the human has for processing pictorial information strategically, i.e. according to the specific contextual needs of the observer [2].

Interest in digital image processing methods stems from two principal application areas: improvement of pictorial information for human interpretation, and processing of scene data for autonomous machine perception. Since the early 1960's, the field of image processing has experienced vigorous growth. Digital image processing techniques are used today in a variety of problems which, although often unrelated, share a common need for methods capable of enhancing pictorial information for human interpretation and analysis. In medicine, for instance, physicians are assisted by computer procedures that enhance the contrast or code the intensity levels into colour for easier interpretation of x-rays and other biomedical images. Image enhancement and restoration procedures have been used to process degraded images depicting unrecoverable objects or experimental results too expensive to duplicate [3]. Successful applications

of image processing concepts can be found in astronomy, biology, nuclear medicine, defence and industrial applications.

The second major application area of digital image processing techniques, mentioned earlier, is in problems dealing with machine perception. In this case, interest is focused on procedures for extracting from an image, information in a form suitable for computer processing. Often, this information bears little resemblance to visual features used by humans in interpreting the content of an image. Examples of the type of information used in machine perception are statistical moments, Fourier transform coefficients, and distance measures [4].

Typical problems in machine perception which routinely employ image processing techniques are: industrial robots for product assembly and inspection, automatic character recognition, automatic processing of finger prints and military recognizance.

This chapter provides an overview of the image processing philosophy and reported applications in computer vision. The objectives and the scope of the thesis are also presented.

1.2 MACHINE VISION

The general goal of a machine vision system is the development of a mechanism for effectively interpreting visual image data. Interpreting an image is defined as the process of transforming a video image signal into a symbolic representation within a computer.

Machine vision is based on three related fields:-

i) Image Processing - the input and output signals are both images, with the output image of superior quality to that of the monitored image.

ii) Pattern recognition - the output representation is a description of the input image derived from a prior knowledge of an expected pattern. The processed visual data is correlated with known object silhouettes to identify the input image.

iii) Scene analysis - is concerned with the mapping of simple features into abstract descriptions for objects that cannot be recognised from pattern matching and deals extensively with three-dimensional image classification.

The motivation behind the use of vision in industry is to increase machine versatility and permit more flexible working to achieve lower unit costs. The human factor in the hostile environment is also significantly reduced with the use of machine vision.

The list of applications where it would be desirable to aid or replace human perceptual processing of images by machine processing already spans every aspect of life. Machines that perceive their environment and perform required tasks have an obvious use in many diverse applications. For example machine feeding is a tedious task which does not enrich human life. This work is often done in an unhealthy environment, sometimes exposing hands and arms to physical danger [5]. Industrial assembly and inspection, planetory space exploration , automated medical x-ray screening, the monitoring of the earth's resources by remote sensors and a range of military applications are other examples in which machine vision has a role to play. They are particularly useful when a non-contacting method of inspecting solid objects is desired, for example when inspecting flimsy or soft objects that could be distorted under pressure.

Computer vision had its beginning with the advent of faster digital computers (1950's). It was recognised that visual data could be represented by numbers and, therefore, be of suitable form for computer processing. The problem was how to process these numbers to obtain usable information.

Selfridge [6] proposed "eyes and ears for the computer"; Attneave [7], Langdon [8] and Stevens [9] published quantitative studies of shape and pattern perception, using a psychological approach and humans perceptual models.

The early work of Roberts [10] processed an image composed of children's blocks and Fischer [11] of printed characters and this latter task motivated the paradigm of 'pattern recognition' research. The pioneers in the field were Rosenfeld [12] and Sklansky [13] who have published numerous papers over the years. The first textbook on computer image processing was published as a computer vision Laboratory report of the University of Maryland, USA in 1968.

1.3 THE GENERAL REQUIREMENTS IN VISION SYSTEMS

A computer vision system, to be feasible for industrial application, must be relatively inexpensive and yet must possess the following characteristics:

i) provide a simple input image

ii) do processing in real time

iii) have a low error rate

iv) be flexible, i.e. be able to accommodate changes in products.

Past research and experience have shown these four characteristics to be key requirements for successful industrial computer vision systems [14]. A brief discussion of the above characteristics is presented.

i) Simple input image

A simple but effective idea is to provide the computer with an image it can analyse easily. This can be done by carefully illuminating the object against a simple background to produce a less noisy, high-contrast image. A simple "line of sight" is also important, since most vision systems do not see the object in three dimensions, but rather analyse a plan view of it in two dimensions. Finally, the "structure" of the object must be simple, i.e. it must be viewed in isolation and not as a part of a larger assembly.

ii) Real-time processing

A conventional mini or microcomputer is inefficient for doing iterated feature extraction and pattern matching. These real-time operations, the most popular image processing techniques in industrial vision systems, require specialised hardware, i.e. an image processor. An important strategy for improving real-time efficiency is to ignore irrelevant areas of the input image. Top-down processing methods, guided by models of the objects of interest, make it possible to exclude such areas.

iii) Low error rates

In order to make the vision system more reliable, those features that can be most effectively and reliably detected are selected first. Next, suitable feature extraction methods are chosen and their performance, reliability and processing cost evaluated.

The system should be designed to be insensitive to failures in the feature extraction process. A very low error rate can be obtained, for example, if the system is made to go on to another feature if the one already detected conflicts with the stored model of the object.

iv) Flexibility

In medium and low volume production, the type of object handled often changes. For medium volume applications, a central file of models can be maintained, from which specific models may be drawn. When a new object appears at a work station, the appropriate model is sent to the station's image processor [15].

1.3.1 The Components of Industrial Vision Systems

The three major hardware components used in industrial computer vision systems are the imaging sensor, the image processor, and the central processor. A brief discussion on each of the components is presented.

i) Imaging Sensor

Because they are relatively inexpensive and widely available, standard television cameras using vidicons are the most commonly used imaging sensors. Other less widely used sensors include solid state linear and area sensors, laser scanners, and image dissector cameras.

ii) Image Processor

In most industrial vision systems [16], the image processing is performed in real time, synchronised with the frame rate of television cameras (1/30 of a second in the USA, 1/25 of a second in Britain). To achieve such speeds of processing, most image processors used in industrial systems would have to employ a locally parallel image processor.

iii) Central Processor

Until recently, special-purpose Central Processing Unit (CPU) hardware had to be designed for each application. The rapid progress of Large Scale Integration (LSI) technology, however, has now made it cost-effective to use general-purpose micros and minis as central processors in industrial vision systems. An example of the efficient configuration of imaging sensor, image processor, and central processor into an industrial vision system is the group-controlled system [17] shown in Figure 1.1. Here, a central processor supports several image processors and many imaging sensors.

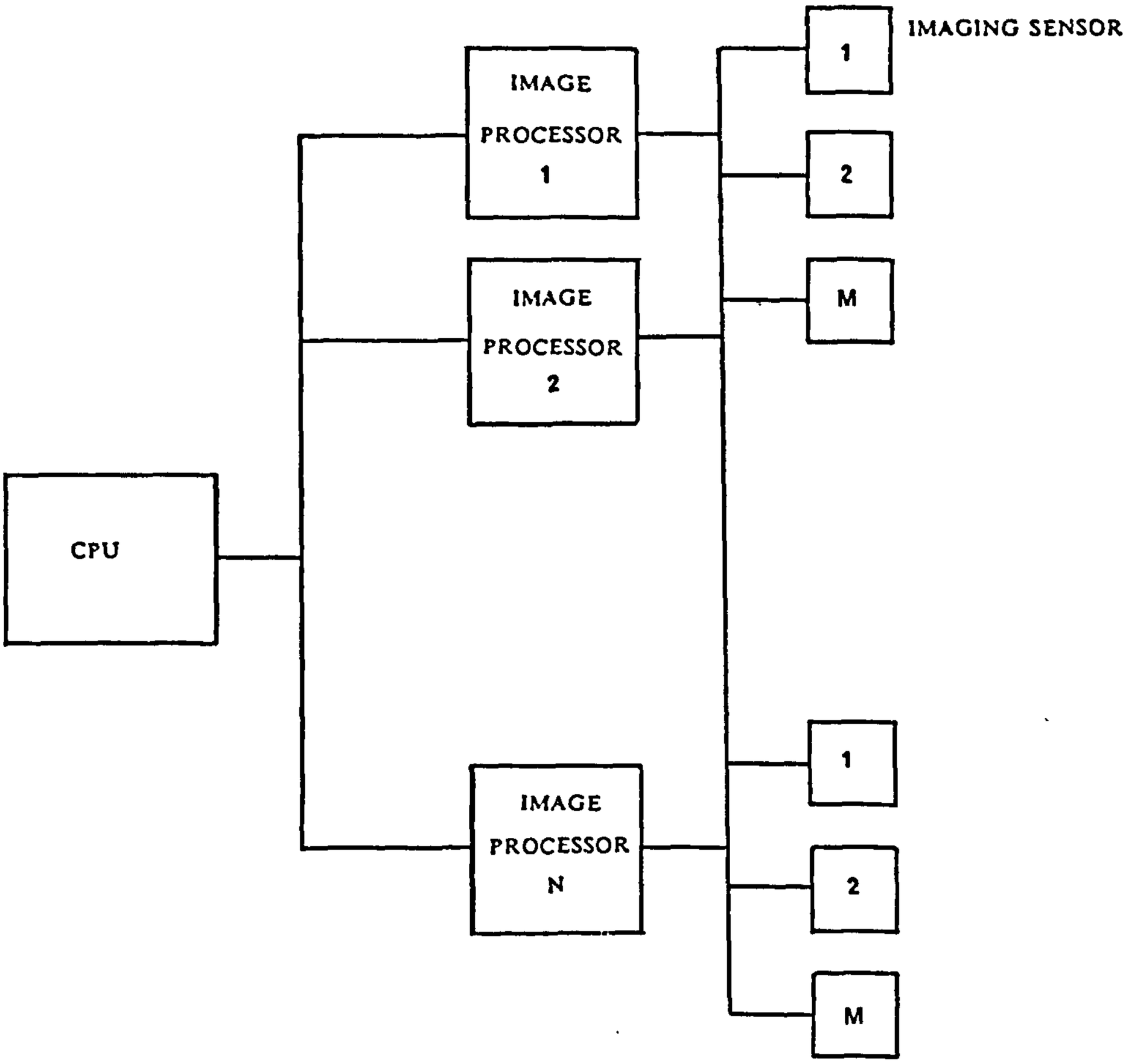


Figure 1.1 A group-controlled system.

1.4 ROBOT TECHNOLOGY

The industrial robot has its origins in both the teleoperator and the numerically controlled machine tool. The former is a remote handling device to permit an operator to perform a task at a distance, while the latter shapes metal automatically, based on digitally encoded cutting data.

Developments in robotics, machine vision, sensors, manipulators, expert systems and artificial intelligence concepts are being combined to create the integrated manufacturing system. A part of this development is the "smart" robot [18] that will be designed to increase productivity and improve the quality of manufactured products. A robot that can 'see' and 'feel' will be able to perform a greater variety of more complex tasks with the minimum of human intervention [19]. A typical arrangement might be as shown in Figure 1.2.

A robot may be considered to be a mobile machine and this can introduce an extra dimension to machine vision. This mobility allows the location from which the next image is taken to vary and even to be dependent upon the results of analysing the current image. It could be programmed to direct its image processing capability to features that will identify particular components [20] and help minimise confusion between object and shadow.

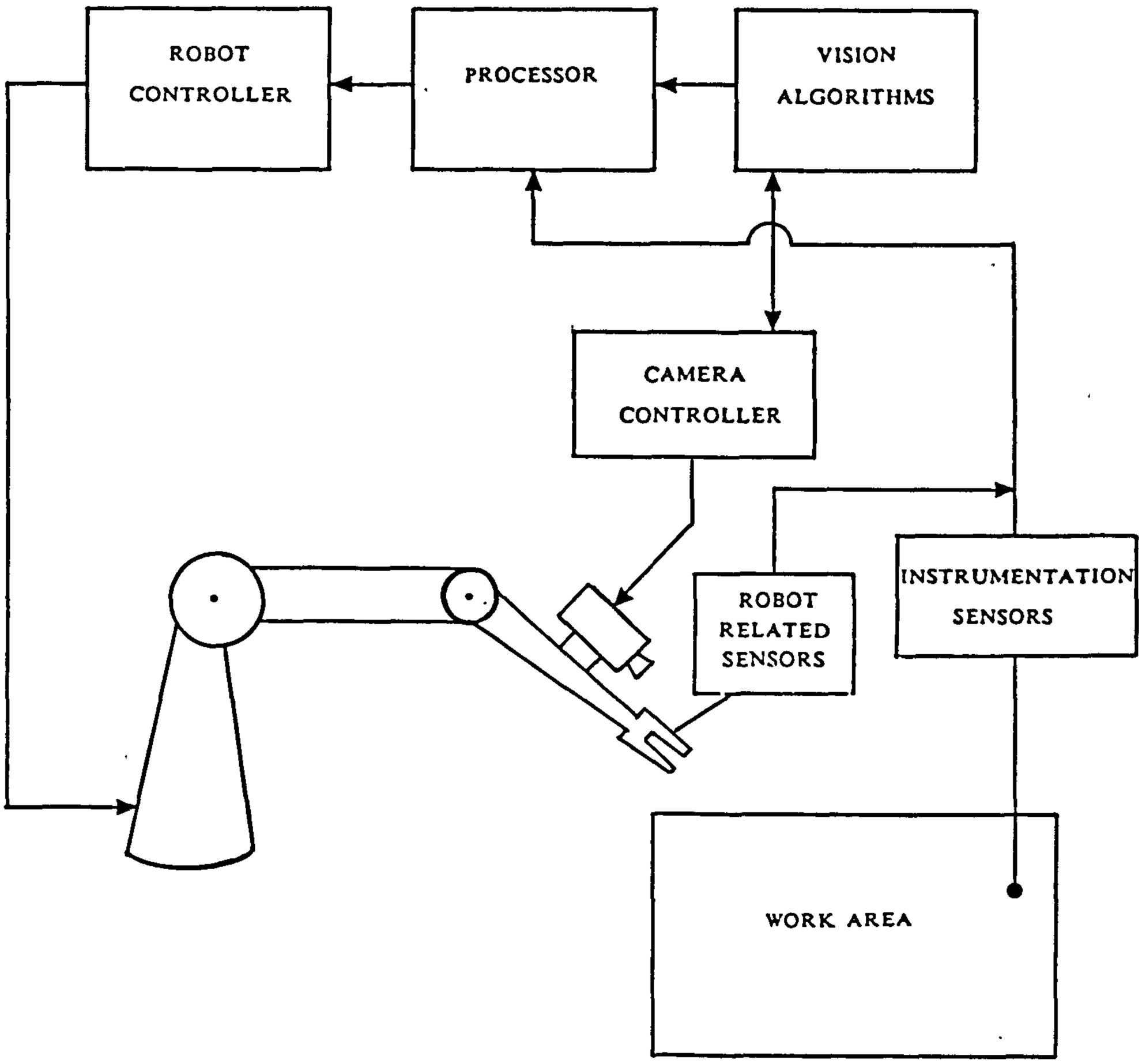


Figure 1.2 The role of computer vision in robot control.

1.5 THE ROLE OF ARTIFICIAL INTELLIGENCE (AI) IN INDUSTRIAL VISION SYSTEMS

Artificial Intelligence (AI) is the study of intelligence using the ideas and methods of computation, i.e. information processing and information representation abilities. These are attributes that form an inherent part of the machine vision systems. Neumann [21] demonstrates that when an industrial vision system is designed without a 'knowledge-base' approach, major shortcomings and limitations in performance will result. It is argued that by the employment of AI concepts, systems with extended application capabilities and predictable performance can be designed and developed [22].

The rule based recognition systems developed, can reduce the regions to be searched and associated pattern matching times [23]. Magee and Nathan [24] have developed a system which represents the topology of two and three dimensional structures using a rule-based approach. By exploiting natural constraints and the manner in which edge types may join at junctions to form higher level objects, a set of rules that describe known models is developed. There is, however, no mention of the difficulty that may be experienced in expanding the approach developed for a curved and more complex surface by the addition of further rules.

Rule-driven machine vision requires a significant knowledge base and a specialised AI computer language, developed primarily for list processing and symbolic manipulation. The AI method is computationally more complex, requires more processing power and as yet is not available at a price that would be acceptable for most industrial applications.

1.6 OBJECTIVES AND SCOPE OF THIS THESIS

The work presented in this thesis is centred on the belief that vision systems utilising relatively simple binary image processing algorithms (image processing/camera-based systems) show the capability of supporting robotic manipulation, essentially for two dimensional non-overlapping objects; and that using range data and modelling techniques, three dimensional object handling and limited scene analysis is possible.

The state of the art in image processing research for computer vision technology, a study of the algorithms for edge and boundary detection and an extension of the existing techniques, are presented in the first three chapters. Developments are made in two-dimensional pattern recognition and orientation estimation and further data reductions are made by use of Fourier analysis and techniques in three-dimensional object recognition. These make up the remaining four chapters.

The thesis organisation is:

Chapter 2 A literature review is presented and industrial systems employing vision techniques at various levels of sophistication for a varied range of applications are discussed.

Chapter 3 An existing processing rig, modified by the author to meet the particular requirements of the vision algorithms developed, is reviewed. Sub-programs are written for edge detection techniques to test the strengths and weaknesses of the available algorithms.

Chapter 4 The Freeman's [25] chain code is employed in two dimensional contour tracking and the data generated used for object identification and orientation measurement in a pattern recognition algorithm.

Chapter 5 An approach is described, based on Fourier analysis, for the fast shape recognition of randomly positioned and oriented components. Two methods are investigated and both methods suggest, for the shapes considered, that the recognition of components can be achieved reliably with a relatively small number of normalised Fourier Coefficients.

Chapter 6 Surface texture analysis techniques are employed to recognise simple objects. Feature vectors, using the power spectra of the images, are developed and the similarities between the corresponding vectors are measured to determine recognition.

Chapter 7 Methods used in image synthesis for developing surface normals are examined in this chapter. In an attempt to recognise the differences in the geometry of simple objects, surface normal vectors are obtained and their effectiveness in recognition procedures are discussed.

Chapter 8 The observations made throughout the project are summarised and discussed.

Chapter 9 The major conclusions that can be drawn from the thesis are presented.

CHAPTER TWO

LITERATURE REVIEW

A survey of work relating to the field of image processing and computer vision is presented, and a prime selection of industrial vision systems incorporating such techniques and principles are discussed.

2.1 INTRODUCTION

The subject of digital image analysis and processing is constantly increasing its spectrum of potential applications, being now an established subject of research and development. New image processing systems are constantly under development [1, 2]. Consequently, new algorithms and new digital system architectures must be developed to handle the enormous flow of data from the new imaging systems with proper precision and adequate throughput.

Computer vision, with digital image processing as its nucleus, has been introduced into manufacturing for component inspection and robot control. Both these uses of computer vision are beginning to change manufacturing methods, with some systems now in production [3].

There is a large number of possible vision tasks, involving essentially two-dimensional image analysis techniques for feature extraction, segmentation, shape analysis and model matching. Few papers exist, however, that describe the problems and techniques involved in three-dimensional scene analysis and object recognition.

This chapter will give a general indication of the state of the technology; discussing briefly, a selected number of existing systems. The chapter begins with an overview of

two-dimensional techniques and applications and continues with a review of the published literature on three-dimensional techniques; addressing problems involved and possible theoretical and engineering approaches for simple three-dimensional object recognition.

2.2 TWO-DIMENSIONAL ANALYSIS

Over the years, two-dimensional image processing techniques for pattern recognition and orientation measurement, as well as improving and enhancing pictorial data have become well established research areas. The early work of Roberts [4], Sobel [5] and Prewitt [6] on edge detection and image segmentation, followed by coding techniques such as Freeman [7], paved the way for developing algorithms dealing with simple object recognition and orientation estimation for applications in robot technology. Further, introducing digital signal processing techniques into the field brought more interesting research and application areas and played an important role in todays state of the technology [8].

One of the most important and widely used signal processing technique is the application of Fourier analysis.

2.2.1 Fourier Analysis Applications

The contour representation of shape by use of Fourier analysis has been attracting the attention of many researchers in recent years.

Surface representation techniques and pattern recognition algorithms have been developed using Fourier descriptions of the image contour by Nikravan et al. [9]. Recognition techniques which are independent of the image size and object orientation, are shown to have potential engineering use.

Arking and Rosenfeld [10] have developed both time and frequency domain techniques for the estimation of cloud motion. Results are presented for both simulated data and data from weather satellites.

Badreldin and Wong [11] have developed shape descriptors for curves and trajectories. They have extended the one-dimensional Fourier analysis to n-dimensional contours by extending the one-dimensional procedure. They do not, however, discuss how the coordinate values are obtained and work from a premise that data measured from an orthogonal coordinate system exist as a parametric representation ($X^{(1)}(1)$, $X^{(2)}(1)$, ... $X^{(n)}(1) = Z(1)$), where 1 is the arc length with reference to a starting point $X_K = (X^{(1)}_K, X^{(2)}_K, ... X^{(n)}_K)$.

Every location $(X^{(1)}, X^{(2)}, ..., X^{(n)})$ of the curve can be expressed as a function of arc length, each of these functions is expressed as a Fourier series:

$$X^{(i)}_K = a_{oi} + \sum_{N=1}^{M} [a^{(i)}_{NX} \cos(\frac{2\pi N}{L} l_K) + b^{(i)}_{NX} \sin(\frac{2\pi N}{L} l_K)]$$

where:

$$l_K = \sum_{K=1}^{m} \sqrt{(X^{(1)}_{K+1} - X^{(1)}_{K})^2 + (X^{(2)}_{K+1} - X^{(2)}_{K})^2 + \ldots\ldots + (X^{(n)}_{K+1} - X^{(n)}_{K})^2}$$

m = total number of points

i = 1, 2, ..., n

M = number of coefficients

N = 1, 2, ..., M

L = total length of the trajectory

Wallace and Wintz [12] have developed a technique, based on their Normalised Fourier Descriptors (NFD), for three-dimensional Aircraft recognition. In their procedure a three-dimensional object is represented by a library of projections, whose NFD's are compared to the NFD of the unknown projection. An interpolation procedure on NFD's enables more accurate determination of the angle at which the unknown object is viewed than can be achieved by methods which simply take the orientation to be that of the nearest library projection. Their work is mainly concerned with the tracking environment.

2.3 3D CONCEPTS: PROBLEMS, DEFINITIONS, METHODS AND SYSTEMS

The basic two-dimensional techniques all break down, at least to some extent, when applied to 3D problems, i.e. recognising the individual elements in a bin of parts. In this quoted example the image cannot be thresholded because the illumination can not be adequately controlled, and the parts cast shadows from one to another [13]. Edge detection would give confusing results, since there is no easy way to distinguish the edges of objects from the edges of shadows. Clearly, when an object is rotated in 3D space, the shape of its silhouette can change drastically, hence, any results from image segmentation cannot be considered reliable because no unique model can exist for comparison. Each change in object orientation would require a different "learnt" model to be available.

Rosenfeld [14] explains that the interpretation of the 3D image becomes much easier if the "topography" of the visible surfaces can be determined. Inevitably, problems can still arise with overlapping objects and this condition is referred to as 2½ dimensional [14]. Recognising this limitation, it can be argued that topographical information should be a sound engineering basis on which to develop a scene analysis processing technique for industrial application.

2.3.1 Deduction of 2½ Dimensional Information for 3D Object Recognition

The most straightforward way for obtaining 2½ dimensional information is by direct measurement. Sensors have been developed that will measure directly the distances to all visible points in the field of view [15, 16]. The output from these sensors [17] is a "range image", that is, an array that represent range values rather than brightness. One approach is to illuminate one point at a time with a coherent beam of light, the phase shift of the reflected light is measured [18] and from this value the range to the point can be computed. A second approach is to project a grid pattern of light onto the scene and the distortion of this pattern is analysed to deduce shape [19].

An alternative range measurement approach is to use two images taken from different view points. If corresponding scene points in the two images can be identified, range data is estimated by "triangulation". The difficulty with stereomapping is that it is not always possible to find pairs of corresponding points. On a smooth, featureless surface, the best that can be done is to try to match up pairs of distinctive points, such as points that lie on edges [14].

With only 2D image data available, 3D object information can be extracted. For example, surfaces generally reflect light diffusely and by employing the change in brightness around a chosen image point, the surface curvature can be estimated point by point [20]. An approach known as "shape from shading" (sfs) because of the inferred 3D surface information. This method could have numerous engineering applications and will be used later in the Thesis. This idea is extended to a technique called "photometric stereo", in which two or more images are taken of a scene, using the same camera position but changing the direction of illumination. The shading in each image imposes constraints on the attitude of the surfaces and from these the orientations are uniquely identified [21].

In other work for the recovery of three-dimensional shapes and surface orientations from single two-dimensional contours, Augusteijn and Dyer [22] take a model-based approach and derive algorithms using contours and point patterns.

If a surface has regular markings, its orientation can be deduced from the way the pattern of markings is distorted by perspective; this idea is called "Shape from texture." Kanade [23] has developed interesting results using algorithms based on this technique and shadow geometry. Surface texture will also be used later in the Thesis.

Finally, if an object is known to have rectangular surfaces, its orientation can be estimated from the way the shapes of these rectangles are distorted in the image; a technique known as "shape from shape", that is , inference of 2½ D surface orientation of shape from the 2D shapes of regions in the image [13].

2.3.2 Processing of Surface Information for 3D Object Recognition

When the 2½ D shapes of an object in the scene are known, shape description or matching techniques may be used to inspect the surfaces, or to identify the objects to which the surfaces belong. For example, bumps or dents in a surface can be detected if they give rise to high surface curvature [24]. Objects can be recognised by comparing properties of the observed shape with the expected property values for the objects. For example, a polyhedral object can be characterised by the shapes of its faces and the angles at which the faces meet.

The image analysis described give rise to a decomposition of the image into regions, or of the scene into objects. A "literal" description of the image or scene can, therefore, be given in the form of a relational structure in which the nodes correspond to features, regions or objects,

labelled by lists of their property values (shape, texture, colour etc.) and the arcs correspond to relations (adjacency, relative position, etc.) [25]. This type of "semantics-free" description is usually not what is wanted, generally object recognition is achieved by comparing descriptions to stored "models", which are generalised features defining object class.

Model formulation is difficult because constraints on the allowable property value and relationships cannot be defined with confidence.

The most extensive work on 3D object recognition from 2D images is the ACRONYM system, developed by Brooks [25]. This system incorporates methods for predicting the two-dimensional appearance (shape, shading, etc.) of an object in an image for a given camera location. Conversely, it provides a means of defining constraints on 3D properties that could give rise to a given image, and for manipulating constraint data. These capabilities are incorporated in a prediction/verification process which uses the image to make predictions about the object and verifies that the image does arise from an object that satisfies the resulting set of constraints [26].

There are, however, many problems associated with the model matching task. The pioneering work on the inference of

relational structure models from examples was done by Winston [26]. For a set of models, it is not clear how to choose combinations for comparison with the object; the selection is known as "indexing" problem. A method suggested by Rosenfeld [14], is to use the models to control the image analysis process and design the process to eliminate most of the possible models at an early stage in the analysis.

2.3.3 Hierarchical Modelling and the Syntactic Approach

In an environment where a priori knowledge of objects is available together with a suitable data base, it is appropriate, to model regions or objects hierarchically [27]. Regions or objects are composed of parts arranged in particular ways, where the parts themselves are arrangements of subparts, and so on. There is an analogy between this type of hierarchical representation and the use of grammer to define a language. In defining languages a sentence is composed of phrases which are in turn composed of clauses, etc. Based on this observation the process of recognising an object as belonging to a given hierarchically defined class of objects is analogous to recognising a well-formed sentence as belonging to a given language. This "syntactic" approach to object or pattern recognition has been extensively studied by Fu [28]. It has been used successfully for recognition of two-dimensional shapes, patterns, and textures; it is not, however, appropriate for 3D object recognition, since no mechanism exists for relating 2D images to 3D objects.

In the following sections shape from (i) Shadows, (ii) Texture, and (iii) Shading is reviewed.

2.4 SHADOWS: BASIC CONSIDERATIONS AND APPLICATIONS

Image shadows provide information which is useful for determining the 3D shapes and orientations of an object in a scene. The interpretation of shadows involves three distinct phases:-

(a) locating the shadow regions

(b) determining which object has cast each shadow region

(c) Geometrically deducing information from the identified object and shadow pairs.

Knowledge gained from (b) and (c) can be used to improve (a) with noise present [29, 30].

Techniques for finding shadow regions, have been proposed by many researchers usually by looking for regions of low intensity with a similar 'hue' to a neighbouring region [29]. A close examination of region contours will reveal that shadows due to the sun will have a slightly bluer hue than illuminated portions of the same surface. Lowe and Binford [31] proposed criteria to identify edges of shadow regions. Witkin [32] investigated shadow edges. Waltz [33] developed a method for labelling lines in line drawings as shadow edges, using local geometric criteria at vertices.

The correspondence problem has been explored primarily by Lowe and Binford [31]. They describe several properties of this correspondence, and include descriptions of special points from which degenerate cases arise. O'Gorman [34] proposed a heuristic method for finding correspondences in the 'blocks world' under orthography.

Geometric interpretation of shadows have been made by Lowe and Binford [31], using shadows from aerial views to determine height. They measure the distance in the image between the outline of an object and the outline of its shadow, and use similar triangles to conclude that this distance is proportional to the height of the object's edge above the ground. Quam [35] has also used shadows to determine depth information. These techniques have also been employed in manual photo-interpretation of aerial photographs [29].

2.5 TEXTURE ANALYSIS AND ITS APPLICATIONS IN SCENE UNDERSTANDING

The concept of texture relates to no rigid description, however, a dictionary definition of texture is 'something composed of closely interwoven elements'. The description of interwoven elements in hinting to the texture resolution, that is the average number of pixels for each distinct texture element. If this number is large, it may be possible to describe the individual elements in some detail [36]. As this

number nears unity, however, it becomes increasingly difficult to characterise these elements individually and they merge into less distinct spatial patterns.

Texture can provide important information for image analysis [37]. Textural features are used to classify images, and differences in texture is used to detect objects in an image. Three-dimensional information such as depth, slope and curved surface measurements can be derived from texture gradients. Therefore, the problems of texture analysis are classified as

(a) Measurement of textural features

(b) Measurement of texture gradient

(c) The segmentation of texture images

2.5.1 Measurement of Textural Features and Texture Gradient

A number of textural features have been proposed by researchers in recent years [37, 38, 39]. Most of them were developed for the purpose of classifying images (or previously extracted portions of an image) on the basis of texture. They can be used in addition for region segmentation or gradient measurement. Texture analysis focuses on models of texture synthesis, such as statistical models [40], structural models [41], time-series (autoregression) models [42] and random mosaic models [36].

2.5.1.1 Analysis in spatial-domain

Texture analysis methods in the spatial domain can be classified, based on the unit considered for the analysis, as follows:

(a) Pixel

Some textural features of an image can be derived from statistics of the grey-level distribution of pixels. Haralick [43] computed a co-occurance matrix, each element (i, j) of which is the probability that a pixel separated by a fixed displacement ∂, $\partial = (r, \theta)$, (r and θ in polar coordinates) from a pixel of grey level i will have grey level j. Weszka [40] has computed a difference statistics (histogram) k-th element of which is the probability that two pixels separated by displacement δ will have a grey-level difference k. Galloway [44] computes a 'run length' matrix each element (i, j) of which is the frequency with which j pixels of grey level i continue in direction θ.

(b) Edge Element

Edge elements are detected by a gradient operator. An edge is characterised by the edge value, the edge direction and the edge size. The average values of edge size (edge density) are correlated with the coarseness of the texture [45]. Rosenfeld [46] has measured the slope of surfaces from the gradient of local average edge values.

(c) Extrema

Mitchell [47] has used local maxima and minima of grey-level (extrema) which are detected by scanning an image in the horizontal and vertical directions. An extrema is characterised by the size (corresponding to the contrast of the texture), and the frequency of extrema of different sizes is computed.

2.5.1.2 Analysis in frequency domain

If a texture is at all spatially periodic or directional, its power spectrum will tend to have peaks for the corresponding spatial frequencies. These peaks [36] can form the basis for a pattern recognition discriminator. Another way is to partition Fourier space into zones. Two kinds of zones are commonly used, namely radial and angular. See Figure 2.1

These spaces, together with the power spectrum are used to define features.

2.5.2 Segmentation of texture images

In images where a variety of textures exist, the regions corresponding to each texture can be detected to assist in the study and the analysis of surface properties as well as eliminating simple shadows in the image.

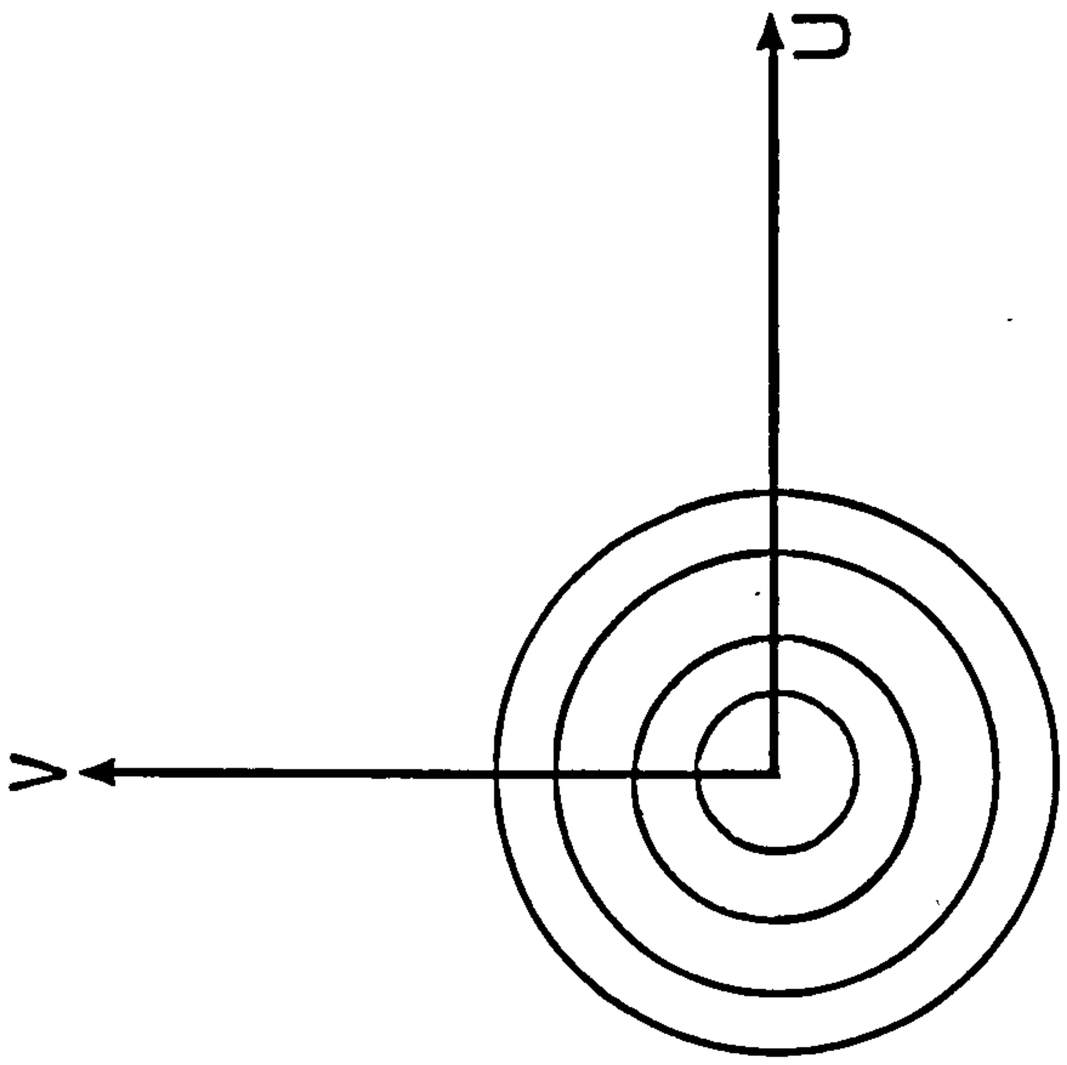


Figure 2.1 The radial and angular zones for the partitioning of Fourier domain in the Fourier transformed images.

Rosenfeld [48] developed a technique based on calculating the difference of average grey level between each pair of neighbourhoods in the image and calculated the best size and direction of the texture edges as well as the corresponding edge values.

Tomita [41] proposed an averaging operator whose neighbourhoods for averaging are variable from point to point. This method preserved the texture edges between regions.

2.6 SHAPE FROM SHADING

Display and analysis of three-dimensional data are important in several different disciplines. They are useful if there is a need or desire to visualise 3D surface. This is the case in many situations of practical importance. Shaded surface displays are a two-dimensional representation of a three-dimensional surface. Shaded surface displays have been successfully employed in the algorithms developed for Robot vision and inspection techniques as well as medical reconstructive surgery [49] and computerised Tomography [50], where the 3D nature of the surface is conveyed with the aid of visual cues such as perspective, shading, texture, shadowing, etc.

However, shaded surface displays are not really suitable for immediate full visualisation of the 3D volume. That is, they require some preprocessing of the 3D data in order

to extract the desired surfaces. In most algorithms, the surfaces are described in terms of patches which join to form the complete surface. Several investigators [51] have employed 'contextual' shading schemes wherein the shading of a displayed face depends on the orientation of its neighbours, its distance from the observer (i.e. range), and the incident angle of the light. Each face is represented by its contour and the 'smoothness' of the contour is controlled by a linear factor controllable by the system user. Details of contour tracing and reconstruction techniques and also, the smoothing considerations, are included in Chapter 5.

2.7 INDUSTRIAL SYSTEMS AND APPLICATIONS

Lothar Rossol [52] has characterised the universe of vision machines as shown in Figure 2.2. Essentially, most vision systems operate in the y = 0 plane, that is they are restricted to shades of grey and cannot deal with colour picture data. Furthermore, commercially available machines are positioned along the Z-axis, that is, they are capable of processing black and white or binary picture data only. Tasks of substantial complexity can be solved with those machines, so long as the visual scene can be thresholded, resulting in a binary picture.

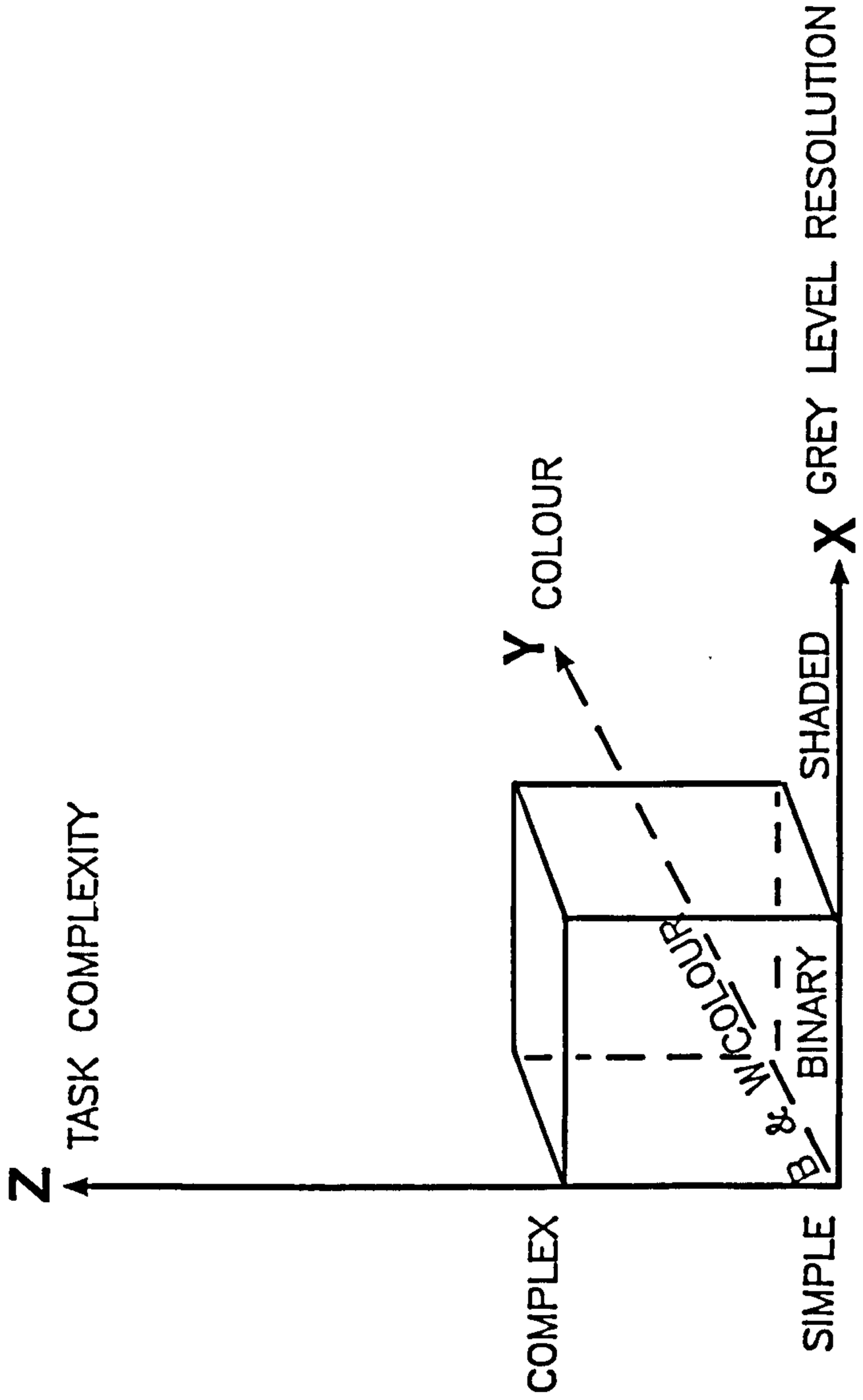


Figure 2.2 Dimensions of computer vision.

The difficulty is, of course, that the majority of industrial tasks do not fall on the z-axis. Significant variations in light reflectance from part to part will usually complicate the task. The major problem in industrial vision tasks, that of object background separation, or segmentation, is thus generally not solvable by simple thresholding. The alternative approach is to use structured lighting.

In the late 1970's and early 1980's the work on two state of the art vision systems were completed at General Motor Research Laboratories, USA. CONSIGHT [53] which is a visually-guided robot system uses structured light, while KEYSIGHT [54] which is for the inspection of valve spring assemblies on engine heads for the presence of valve spring cap keys, is an example of a computer vision system that processes grey level picture data. The structured lighting of CONSIGHT converts a difficult grey level vision problem into an inherently simpler binary problem. CONSIGHT, however, differs from conventional binary systems in that three-dimensional structure can be calculated directly, rather than having to be inferred from light reflect-ivities [53]. It is shown that structured light used in conjunction with stereo vision, results in highly reliable three-dimensional vision systems [36]. KEYSIGHT demonstrates a shortcoming of the grey level inspection systems: they are not easily programmable, especially when compared to binary systems [54].

Two earlier vision systems, namely SIGHT-I [55] and GCA [56] (Grid Circle Analyser), were developed at the same Laboratories in 1976 and 1978 respectively. SIGHT-I, a forerunner of many vision systems used by the electronic industry, automatically aligns and inspects integrated circuit chips, an operation which was performed manually at General Motors factories prior to the installment of the vision system.

The key to a clear understanding of sheet metal formability is the measurement of surface strains after a stamping operation. Steel manufacturers have developed a surface strain measurement method which involves printing a grid of small circles into approximate ellipses. Measurement of these ellipses provides the data necessary to calculate the surface strain. The GCA automates this measurement [56].

Researchers at Carnegie-Mellon University (CMU) have developed an Autonomous Land Vehicle (ALV) whose initial task was to follow roads and sidewalks and on the University campus and avoid obstacles such as trees, humans, and traffic cones [57]. The vehicle, is equiped with three types of sensors: two colour television cameras for stereo vision which give reflectance information to find road edges; a sonar ring, which is a collection of 24 sonars for measuring distances; and a scanning laser rangefinder which has the ability to find both the road boundaries and measure distances, and is used to cross-check the data obtained by the other sensors. The

research vehicle, called the Terregator (short for Terrestrial Navigator) has been designed and built at CMU, the work for which started in 1984 and became operational by the end of 1986. A structured design approach has been adopted to allow development of individual modules [58].

A Waseda University (Japan) project team have built an intelligent robot, Wabot-2 which can play an electric piano, using the fingers and feet, while reading printed music [59]. The software developed uses a hierarchical data structure for printed music and the robot vision system has to distinguish between the black and white keys as well as the relative position of each note. These processing ideas are extended to a conversation mode by using a voice synthesizer. It has been reported [59] that pertinent data can be recognised in approximately 15 seconds, with close to 100% accuracy.

2.8 DISCUSSION AND FINAL REMARKS

The field of computer vision is another example of how changes in technology will make existing industrial practices obsolete. Since 1970 computer vision has been studied extensively at all three levels of: basic research, application oriented research and practical system development [60].

The development of machine vision will open up another approach to industrial mechanisation at a time when manufacturers are looking both for higher levels and greater flexibility in automation since improved product quality is essential in the modern production process [61].

Vision systems are beginning to have an impact in component inspection [2, 27]. These references indicate a great diversity of applications, not only the visual examination for defects, but also measurement of dimensions, counting, checking and part orientation on conveyor belts. Still state of the art and mostly in the future are applications to control robots (guidance), vehicles, sorting and transport [57]. The requirements for robot vision are different in many respects from the need of inspection and quite different types of system will have to be developed.

The requirements for fewer plant operatives together with the need to introduce more flexible manufacturing schemes, will inevitably lead to a major increase in the use of machine vision in the next decade.

C H A P T E R T H R E E

PRE-PROCESSING AND SYSTEM HARDWARE

A number of edge detection techniques are documented and assessed for their strengths and weaknesses. The system hardware and the computational power are also discussed.

3.1 INTRODUCTION

Image edge detection, identified by a sudden change in grey level, is one of the most important properties used in machine vision. It permits an image to be separated from its background, the first step towards the analysis or description of the image.

This chapter documents a number of basic, classical and modern edge detection procedures. They include gradient methods, template matching, and a more recent class of edge detection operator.

To be able to investigate the different published techniques, computer programs have been written and developed for use on a Digital Equipment Corporation VAX 8600 digital computer and schematic diagrams have been produced to help in the detailed description of the hardware used.

3.2 IMAGE PROCESSING: SYSTEM HARDWARE DESCRIPTION

Described is the equipment used for processing and analysis of the image data. The equipment is developed to allow a non-contacting method for gathering data from specialised experiments that are on-line to the department's computing facility.

A schematic representation of the data collection equipment and a photograph showing the main instruments are given in Figures 3.1 and 3.2, respectively, and a brief description of each instrument follows.

3.2.1 The LSI-11/23 Mini-Computer

The LSI-11/23 is a 16-BIT mini-computer manufactured by Digital Equipment Corporation (DEC). The LSI-11/23 provides over 400 instructions, 16-BIT (word) or 8-BIT (byte) addressing, 8 high speed general purpose registers, a hardware stack and a 4 level fast vectored interrupt capability [1]. The Optional Memory Management System (OMMS) allows up to 256K bytes of protected multi-user program space. The LSI-11/23 system used for the data collection included an LSI-11/23 micro-processor with floating point option (for faster manipulation of floating point data), 256K bytes of random access memory (RAM), 4 serial input lines, a programmable clock (KWV11-A) and a high speed direct memory access (DMA) interface (DRV11-B) connected to the DR11W on the VAX 8600. The image processing boards, which are connected directly to the LSI-11 Q-BUS, include a Matrox QFG01 frame grabber which is capable of digitising an image of 256 * 256 pixels, each pixel having available a grey level range of 1 to 256 values. Included are two units of QRGB-256 frame store which are capable of storing the digitised information and the system runs under a modified RSX11M multi-user operating system. The data storage is provided by 'virtual disks' which are files

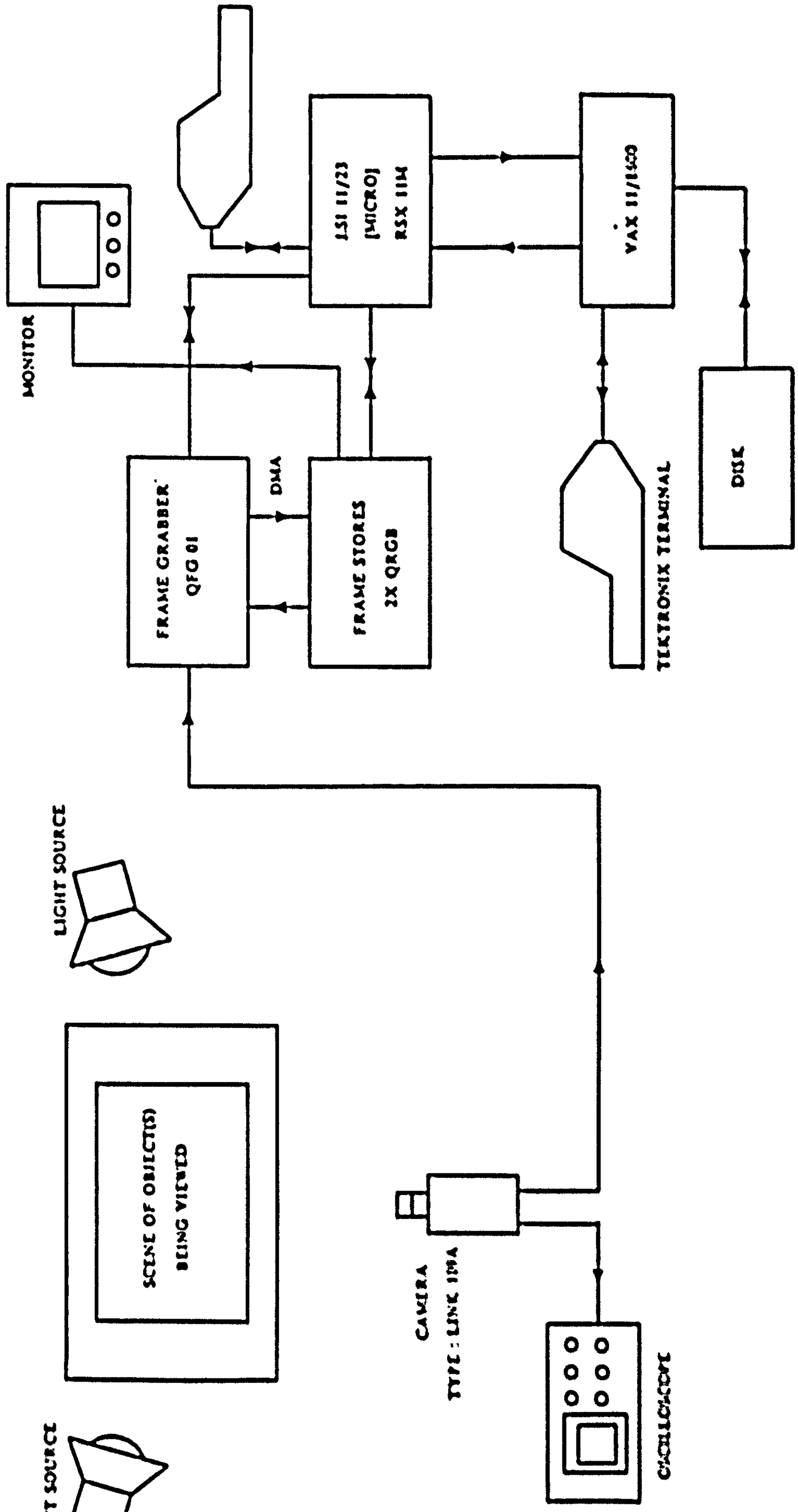


Figure 3.1 Schematic representation of digital image processing system.

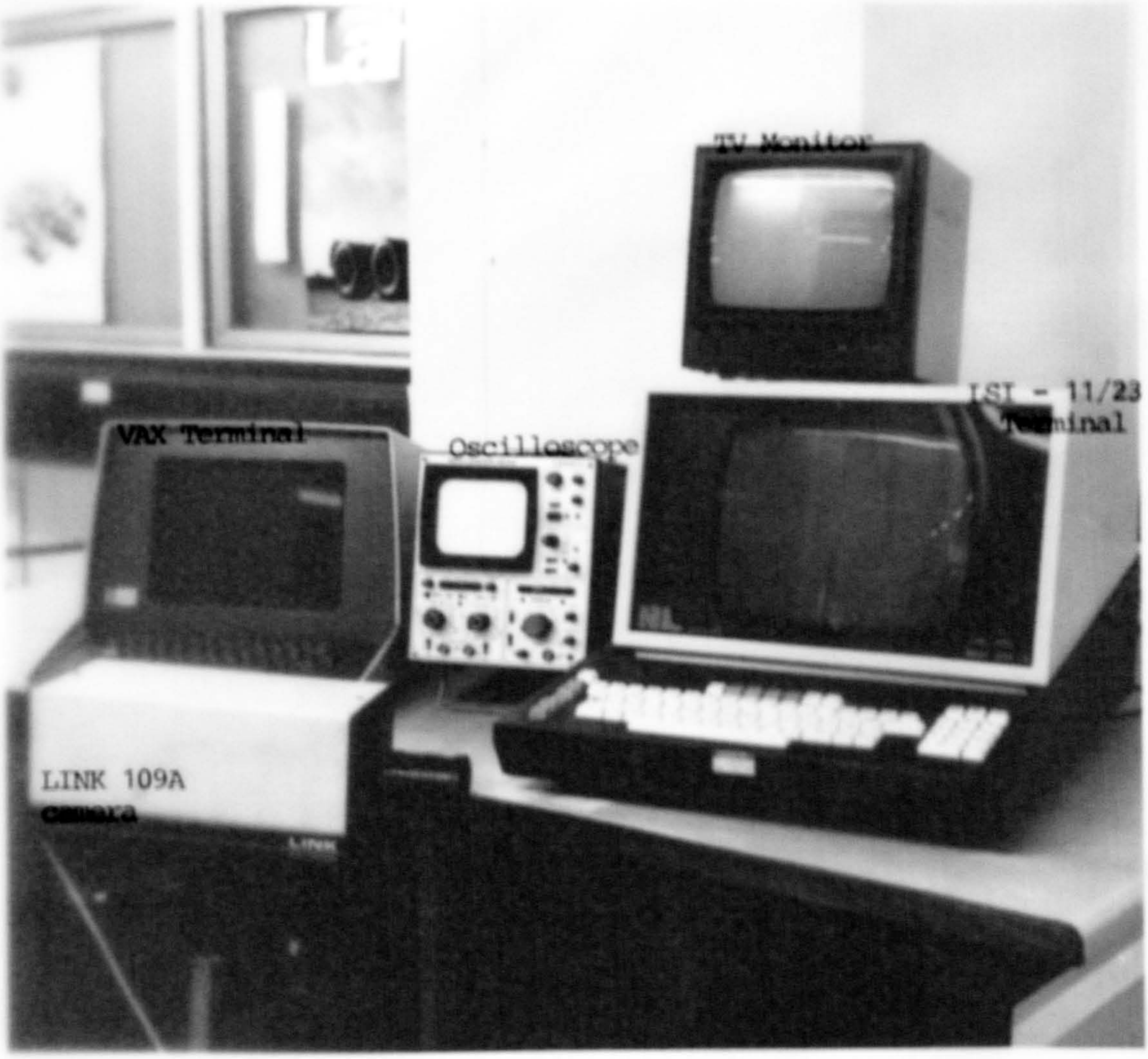


Figure 3.2 Photograph of the main instruments in the image processing system.

located on the high speed disks of the VAX 8600. The DRV11-B to DR11W connection and associated software provided the link between the LSI-11/23 and the 'virtual disks' used for data storage, with the programs written in Fortran and PDP11 assembler code.

3.2.2 The TV Camera and the Attenuator

To send a picture to the LSI-11/23 as a video signal, a standard 625 line monochrome camera, type LINK 109A, is used. The camera has an internal attenuator and has a maximum scanning rate of 50 frames/sec. The attenuator unit of the camera reduces the amplitude of the signal before it goes to LSI-11/23. The LSI-11/23 does not accept a signal with an amplitude in excess of 1 volt.

An oscilloscope unit is used to adjust the camera lens aperture in order to obtain the best dynamic range in the video signal. A monitor redisplays the signal from the LSI-11/23 as a picture on the screen, it is used to examine the contrast of the picture and also to check the positions of the digitised objects in the scenes.

3.3 DATA SIZE AND THE SPEED OF DATA TRANSFER

The camera produces 2 interlaced half images every 40 ms. This means that continuous frame grabbing produces 64K bytes of new data every 40 ms. The data has to be read from the

frame store into the LSI-11/23's memory and then transferred to the VAX for processing. The transfer rate required to do this is greater than 1.6M bytes per second; a rate not obtainable in practice.

The maximum transfer rate of the DMA link between the VAX and LSI-11/23 is 250K words/sec. Allowing for the time the LSI-11/23 requires to access the data from the frame store and for the software overhead for each computer, a practical limit of about 13K bytes per second is obtained.

The two alternatives, therefore, are to send complete frames at a less frequent period or to send only part images every 40 mS. It should be noted that a 25 Hz sampling rate results in a Nyquist frequency of 12.5 Hz so that with real time data capture and transfer, the frequencies that can be monitored without aliasing are limited.

3.4 EDGE DETECTION: DEFINITIONS AND TECHNIQUES

Discontinuities of intensity in an image are called edges, and are a basic part of image information. Experiments with the human visual system have shown that image boundaries are extremely important; often an object can be recognised from only a crude outline [2]. This fact provides the principal motivation for representing objects by their boundaries, the boundary representation being incorporated into the object recognition algorithm.

The algorithms employed for edge detection vary in complexity and the suitability of some only becomes evident when applied to a scene.

Numerous researchers have developed methods for detecting boundaries and edges and a review of the classical approaches may be found in [3]. This chapter proceeds by dividing the techniques into the following categories and examining each category separately:-

A) Differential Operators:

   i) first order operators, eg. Roberts, Sobel, Prewitt.

   ii) Second order operators, eg. Laplacian operator.

B) Template matching, eg. Prewitt, Kirsch, Robinson.

C) New class of operators, eg. Frei and Chen approach.

Examples of the results to be obtained from programs written to solve these algorithms are presented for completeness.

3.5 DIFFERENTIAL OPERATORS FOR EDGE DETECTION

These operators are divided into the two types, first and second order. Amongst the first order types, Roberts [4], Sobel [5] and Prewitt [6] are the most commonly used. If the rate of change of grey levels, D1 and D2 in two perpendicular directions are known, then the maximum rate of change (the gradient) in any direction is given by $(D1^2 + D2^2)^{\frac{1}{2}}$ and the direction of this maximum is $\tan^{-1}(D2/D1)$. For computational simplicity, one may use $|D1| + |D2|$ or $\max(|D1|, |D2|)$, although the first definition, that is $(D1^2 + D2^2)^{\frac{1}{2}}$ has been used throughout this work.

Let IP (x, y) represent a pixel in the digitised image,

Robert's operator uses: D1(x, y) = IP(x, y) - IP(x+1,y+1), and

D2(x, y) = IP(x+1, y) - IP(x, y+1).

Sobel operator uses: D1(x, y) = [IP(x-1, y+1) + 2IP(x-1, y) +

IP(x-1, y-1)] - [IP(x+1, y+1) + 2IP(x+1, y) + IP(x+1, y-1)]

and D2(x, y) = [IP(x-1, y+1) + 2IP(x, y+1) + IP(x+1, y+1)] -

[IP(x-1, y-1) + 2IP(x, y-1) + IP(x-1, y-1)].

Prewitt operator (differential type) uses:

D1(x, y) = [IP(x-1, y+1) + IP(x-1, y) + IP(x-1, y-1)] -

[IP(x+1, y+1) + IP(x+1, y) + IP(x+1, y-1)]

and

D2(x,y) = [IP(x-1, y+1) + IP(x, y+1) + IP(x+1, y+1)] -

[IP(x-1, y-1) + IP(x, y-1) + IP(x-1, y-1)].

The Laplacian [7] operator is classed as a second order

operator. Three different types of convolution masks are used to implement the operator, and these are shown in Figure 3.3.

The manner in which these convolution masks, for both this operator and for the template matching operators of the next section, are applied to the image is shown in Figure 3.4. The effect of signal differentiation is shown in Figure 3.5, and results obtained using the Robert's and the Laplacian operators to detect an image edge are shown in Figures 3.6 and 3.7 for a human face and engineering components.

3.6 TEMPLATE MATCHING OPERATORS

These methods, known also as spatial convolution operations, are used to calculate the variation in pixel brightness around the point of processing. The movement across the image is pixel by pixel, and the output image data relative locations match that of the input image exactly. The calculation is different, however, with the output pixel brightness being dependent on the group of pixels surrounding the one being processed. As is illustrated in Figure 3.4, for every pixel in the input image a value for the output image pixel is estimated by calculating a weighted average of it and its surrounding neighbours.

The most accepted and widely used template matching operators are the Prewitt [6], Kirsch [8] and Robinson [9]

TYPE 1

0	-1	0
-1	4	-1
0	-1	0

TYPE 2

-1	-1	-1
-1	8	-1
-1	-1	-1

TYPE 3

1	-2	1
-2	4	-2
1	-2	1

Figure 3.3 Three different types of Laplacian convolution masks.

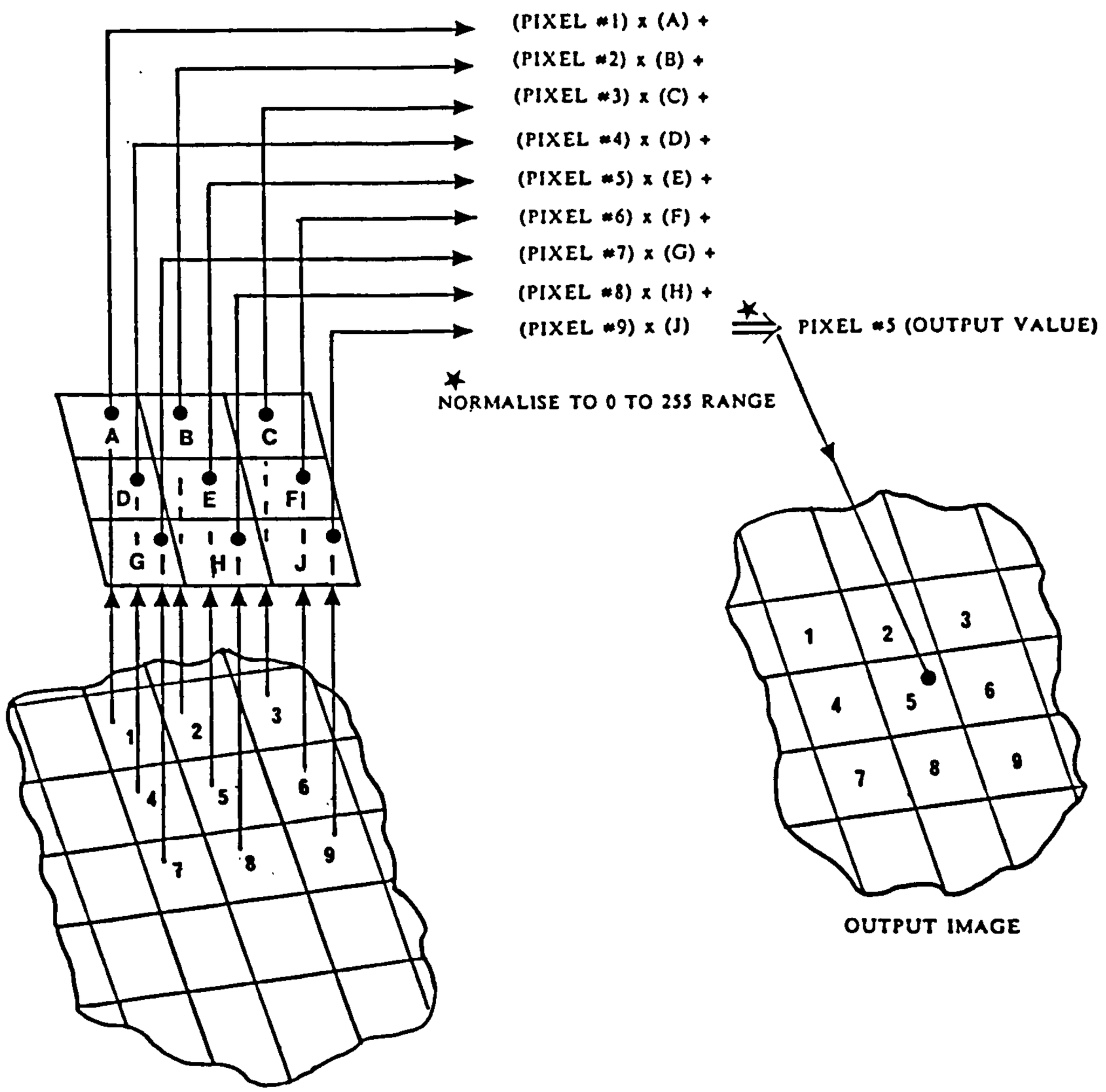


Figure 3.4 Convolution mask calculation flow representation.

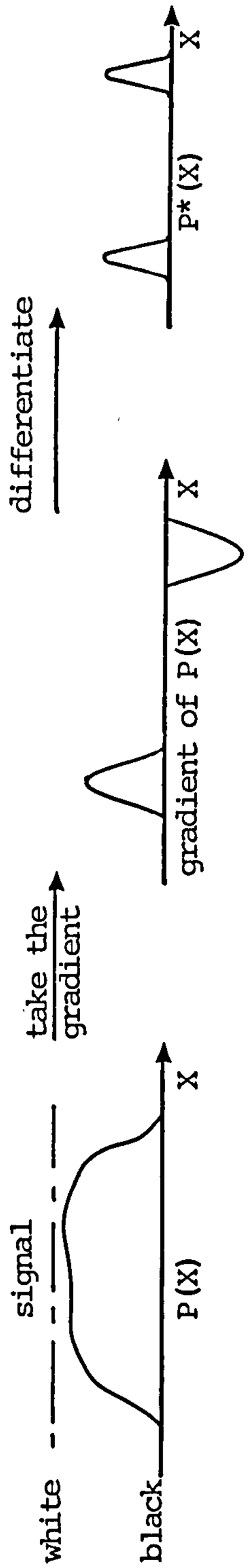


Figure 3.5 Effect of signal differentiation for edge detection.

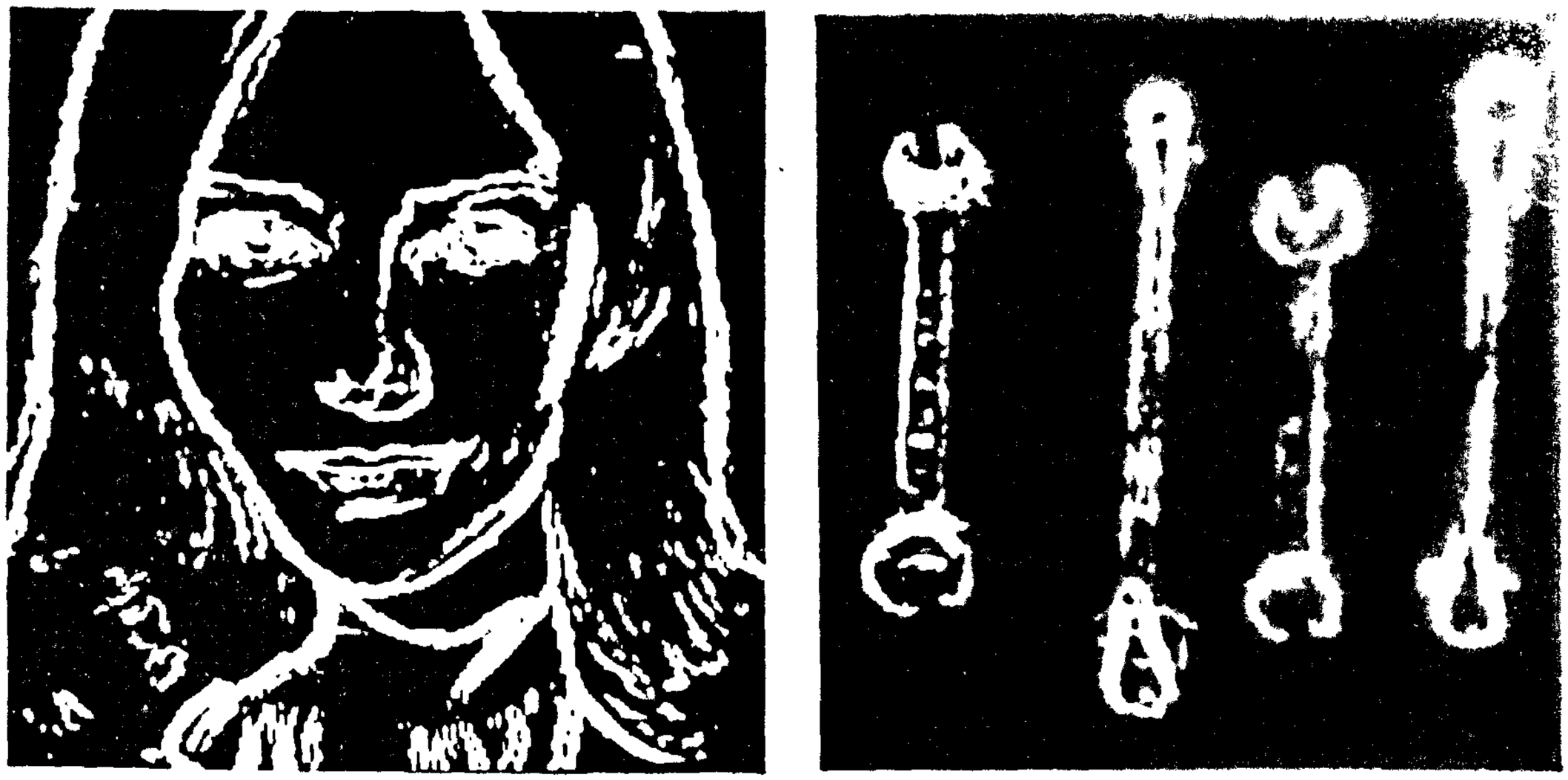


Figure 3.6 Edge detected images using Robert's operator.

Figure 3.7 Edge detection by applying the Laplacian operator.

operators. In each of these methods eight masks are to be considered for convolution, see Figure 3.8.

Each of the 8 edge detectors are applied to the 3 by 3 square of pixels in the way presented in Figure 3.4, and the highest output from the 8 detectors is found and normalised, within the range of 0-255. This value represents the pixel value of the output image in the same location as the centre pixel under the convolution mask.

Figure 3.9 shows experimental results using the Kirsch operator.

3.7 NEW CLASSES OF EDGE DETECTION OPERATORS

Examination of the techniques described, indicate that similarities exist that reveal an underlaying principle. From this information, Frei and Chen [10] have developed sets of orthogonal functions which are closely related to distinctive image features. The properties of these functions suggest ways to minimise the amount of computations and improve decision criterion. Of these, the one chosen by this author for testing is shown in Figure 3.11.

Let the data in local region be labelled B, mask data labelled T, and the product value of the mask and local region

(a)

M1

1	1	1
1	-2	1
-1	-1	-1

M2

1	1	1
1	-2	-1
1	-1	-1

M3

1	1	-1
1	-2	-1
1	1	-1

M4

1	-1	-1
1	-2	-1
1	1	1

M5

-1	-1	-1
1	-2	1
1	1	1

M6

-1	-1	1
-1	-2	1
1	1	1

M7

-1	1	1
-1	-2	1
-1	1	1

M8

1	1	1
-1	-2	1
-1	-1	1

(b)

M1

5	5	5
-3	0	-3
-3	-3	-3

M2

5	5	-3
5	0	-3
-3	-3	-3

M3

5	-3	-3
5	0	-3
5	-3	-3

M4

-3	-3	-3
5	0	-3
5	5	-3

M5

-3	-3	-3
-3	0	-3
5	5	5

M6

-3	-3	-3
-3	0	5
-3	5	5

M7

-3	-3	5
-3	0	5
-3	-3	5

M8

-3	5	5
-3	0	5
-3	-3	-3

(c)

M1

1	2	1
0	0	0
-1	-2	-1

M2

2	1	0
1	0	-1
0	-1	-2

M3

1	0	-1
2	0	-2
1	0	-1

M4

0	-1	-2
1	0	-1
2	1	0

M5

-1	-2	-1
0	0	0
1	2	1

M6

-2	-1	0
-1	0	-1
0	1	2

M7

-1	0	1
-2	0	2
-1	0	1

M8

0	1	2
-1	0	1
-2	-1	0

Figure 3.8 Convolution masks M1 to M8 for (a) Prewitt (template type)
(b) Kirsch
(c) Robinson

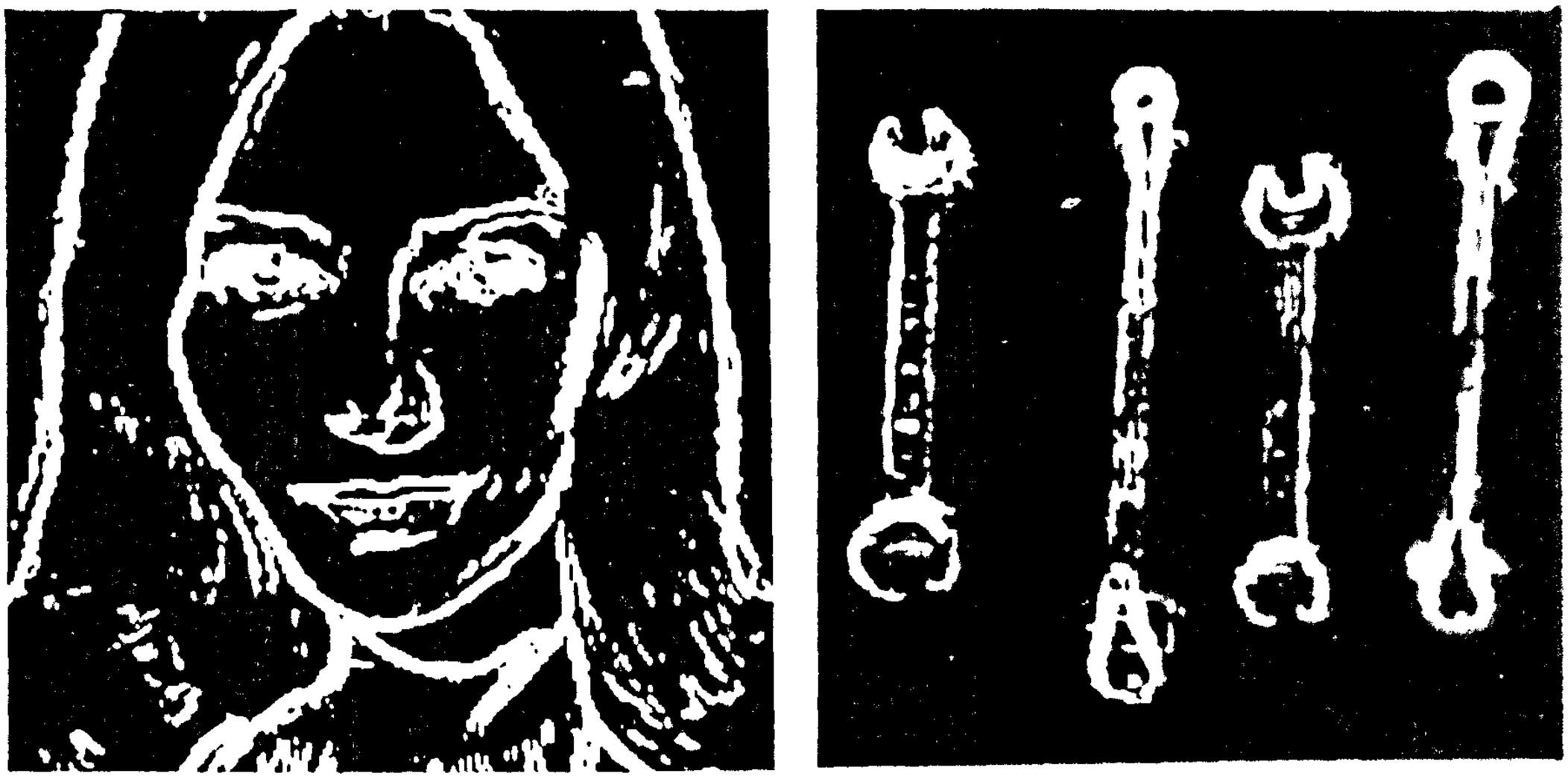


Figure 3.9 Edge detection by applying the Kirsch operator.

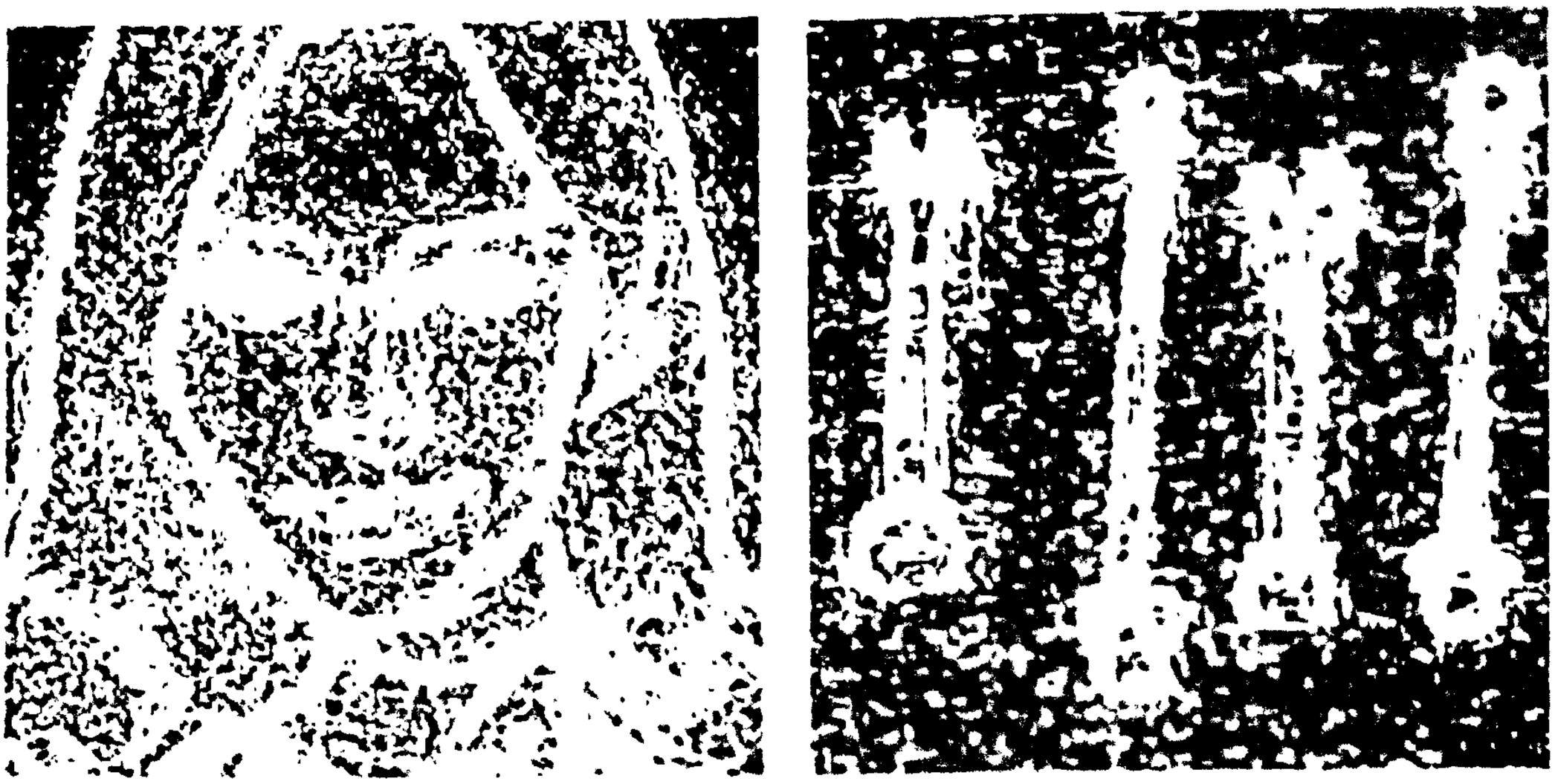


Figure 3.10 Edge detection by applyimg the Frei and Chen technique.

1	$\sqrt{2}$	1
0	0	0
-1	$-\sqrt{2}$	-1

1	0	-1
$\sqrt{2}$	0	$-\sqrt{2}$
1	0	-1

Figure 3.11 The orthogonal bases (masks) used by Frei and Chen.

be (B, T). If $T_1 \ldots\ldots T_n 2$ are orthogonal base masks and $T_1 \ldots\ldots T_e$ of these constitute an edge subspace, the projected angle on the edge subspace is expressed as:

$$\theta = \arccos \left[\sum_{i=1}^{e} (B, T_i)^2 \Big/ \sum_{j=1}^{n^2} (B, T_j)^2 \right]^{\frac{1}{2}}$$

When the value of θ is small, the central pixel is regarded as an edge. If T_1 and T_2 are the masks of average gradient; the edge magnitude is given by :-

$$BP(IX, IY) = \sum_{i=1}^{2} (B, T_i)^2 / (B, B)$$

where BP(IX, IY) represents the edge detected image. The results of using these algorithms are shown in Figure 3.10.

Based on the results obtained using Robert's and Sobel operators and the relatively fast differential operators, an algorithm is developed by this author to provide "options" for the user. The program can be made sensitive to the edges in (i) horizontal direction (ii) vertical direction, or (iii) a combination of both horizontal and vertical directions, the program prompts the user to choose between the ABS (absolute) form or the averaged form of the edge magnitude to be output, see section 3.5 . The values of the terms D1 and D2 are

calculated from:-

$$D1(x, y) = IP(x, y+1) - IP(x, y) \quad \text{and}$$

$$D2(x, y) = IP(x+1, y) - IP(x, y)$$

The weighting functions for this operator i.e. alternative gradient and Robert's gradient operator are shown in Figure 3.12; results using the above operator are shown in Figure 3.13.

3.8 DISCUSSION

The two images chosen to investigate the routines were the human face and engineering tools. They were chosen because of their very different properties. It is evident from the results that some operators are more sensitive to noise and others detect the weak edges better. These results confirm that the attributes of a good edge detection techniques are:

(1) Sensitivity to noise to be minimal

(2) Subtle edges to be detected

(3) Computing cost to be kept low

(4) Dark area edges to be detected

(5) Strong edges to be detected as thinner lines.

Thinning edges involves searching the pixel arrays to find edges whose widths are more than one pixel wide. The maximum intensity pixel across the width of the edge is found and the coordinates stored. A program listing, using this technique and the Kirsch template matching method for edge detection is given in Appendix A1.

Robert's Gradient

1	0
0	-1

0	1
-1	0

Alternative Gradient

-1	1
0	0

-1	0
1	0

Figure 3.12 Weighting functions for two gradient type edge detection operations.

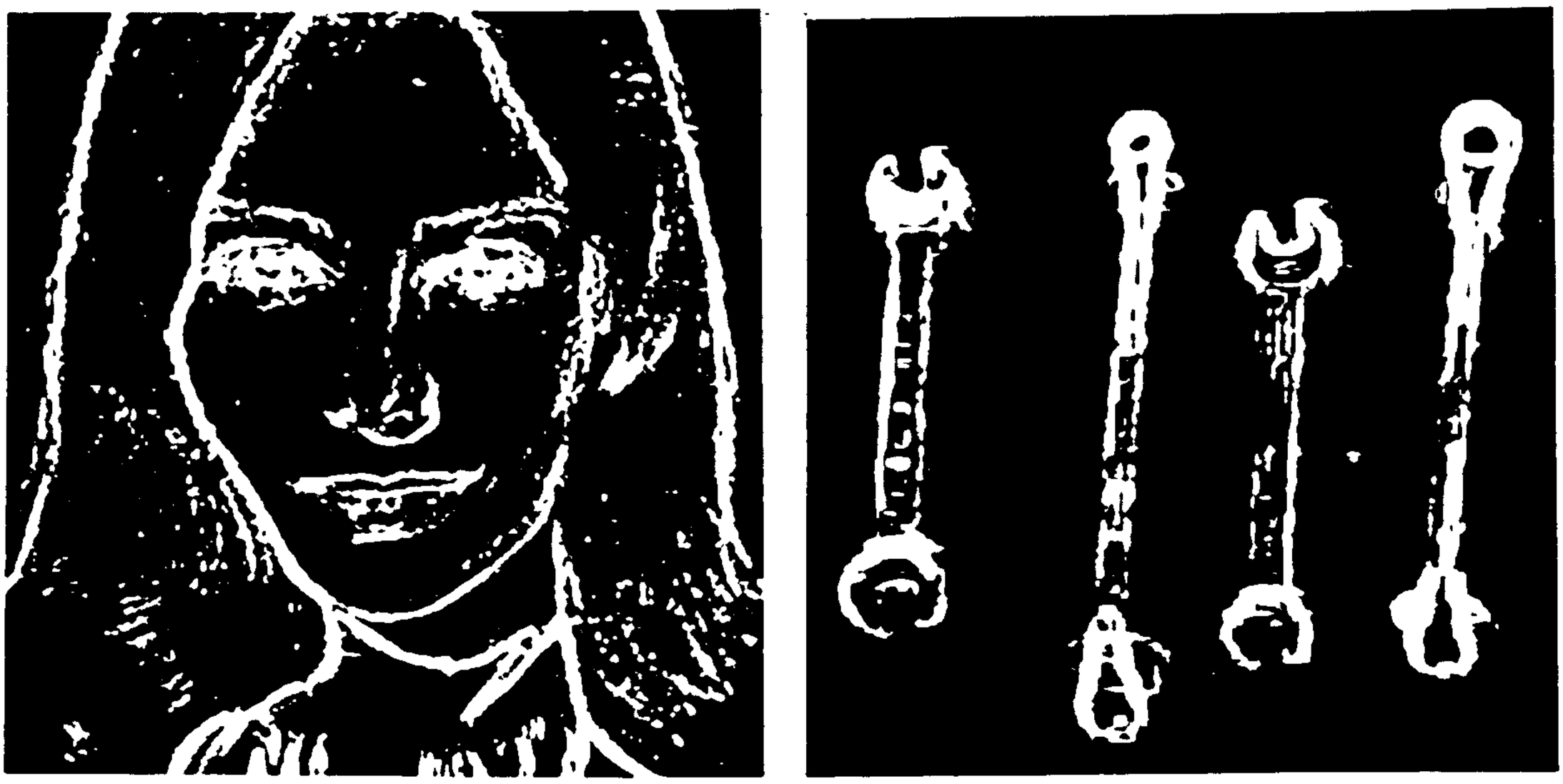


Figure 3.13 Applying the alternative gradient type edge operator.

The techniques tested all have individual strengths that can be exploited, e.g. sensitivity to noise can be reduced as seen in Figure 3.13. No one technique, however, was significantly superior to the others and each different image processing situation will require separate study. To help in this, a summary is given in Figure 3.14 indicating the relative performance of each technique under the heading listed (1) to (4) above. Although each technique has been ranked subjectively, the listing does give a good guide as to the effectiveness of the edge detection methods studied.

Technique	Sensitivity to noise	Detecting Subtle edges and edges in dark areas	Computing Time (average CPU sec.) 10-30 ms :fast 30-50 ms :average 50 ms and above:slow
Roberts	good	good	fast
Sobel	good	good	fast
Prewitt (diff.)	good	good	average
Laplacian	average	average	fast
Kirsch	average	average	slow
Prewitt (templ.)	good	good	slow
Robinson	good	good	average
Frei and Chen	poor	average	average
Proposed (options)	good	good	fast

Figure 3.14 Comparison of different edge detection techniques, based on an average of ten runs.

C H A P T E R F O U R

PATTERN RECOGNITION AND ORIENTATION ESTIMATION USING FREEMAN'S CODE

A qualitative study is made of the application of Freeman's coding model in image recognition and orientation measurement

4.1 INTRODUCTION

In pattern recognition the input is an object image and the output is a description of that image based on a priori knowledge of the expected pattern. The aim of the data processing is to identify the input as one of a usually small set of possibilities depicted in two dimensions. It will be assumed that the part is of a fairly simple shape and can be characterised by a two-dimensional projection, as provided by a single camera view.

The approach adopted is based on the assumption that boundary shape is adequate to recognise the object and that Freeman's coding model [1] will describe the spatial invarient property of the object. To improve user confidence in the object recognition, Geisler [2] estimated the additional properties of object boundary length, area and centre of area.

4.2 OBJECT RECOGNITION

Segmentation is adopted to aid object recognition. The picture stored in the computer memory is partitioned into connected regions, each region is homogeneous in some sense and is identified by a unique label. In this application the grey scale is not important, the stored picture is segmented into object and background by choosing a threshold in brightness. A pixel brightness above the threshold is defined as object and

below the threshold as background, i.e. a binary picture results. The object contour is traced and chain encoded [2]. The perimeter length and area enclosed by the contour are computed as global features and the shape is described by a "contour curvature" which is a function of contour length.

A hierarchical 2-stage classification is used for object recognition. First, checks are made as to whether an object of similar perimeter length and area are stored. If no reference object is found, the object is rejected and classified as unknown. Otherwise, a match is made between the contour curvatures of all the objects with similar global features. If no object with similar curvature is found (within the acceptable tolerance to be explained in section 4.5) the object is rejected.

The orientation is computed as the rotation angle between the object and its stored reference image. The computing approach to the data processing is based on modular software to allow individual sub-program development and algorithm changes to be made independently of the main program.

Figure 4.1 shows the simplified Flow chart of the approach adopted.

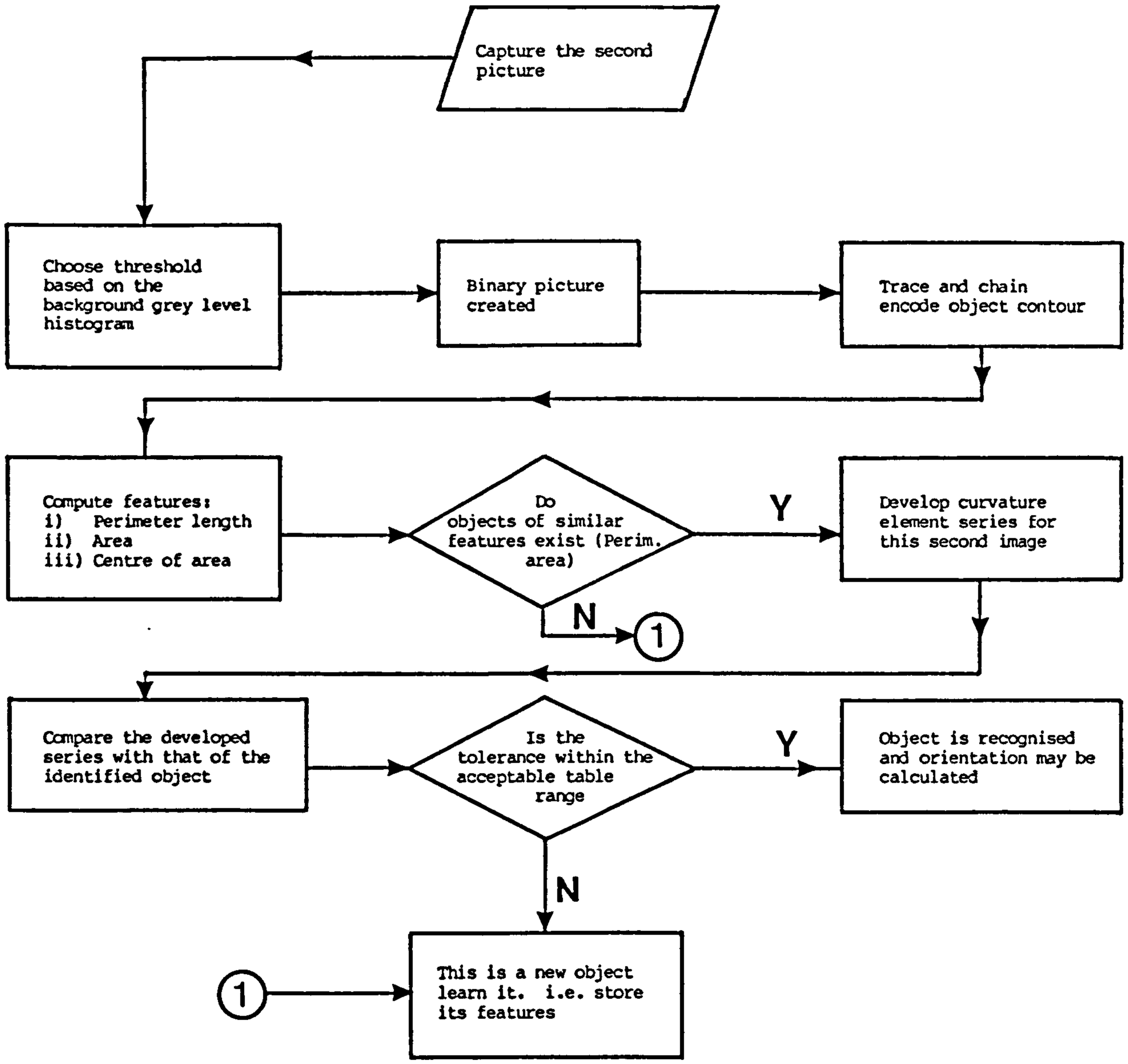


Figure 4.1 Flow chart for object recognition algorithm.

4.3 ALGORITHM DEVELOPMENT

Prior to considering the algorithm for orientation estimation it is important to realise the need for and the essential characteristics of any coding method:-

(i) reducing the size of the arrays containing image data reduces the computational time for data analysis

(ii) the coding system must be spatially invarient, i.e. the sequential order of the elements in an object chain code representation must not change with object rotation, only its start point should change. This condition is satisfied by use of the Freeman code [2,3,4].

4.3.1 The Freeman Code

Freeman's work on two-dimensional boundary encoding and processing is used by many researchers because the resulting image information form is readily accessible and directly amenable to interpretation.

Figure 4.2 shows the three employed types of boundary lines as identified for a binary image:

(i) the geometric boundary, separating the area of black

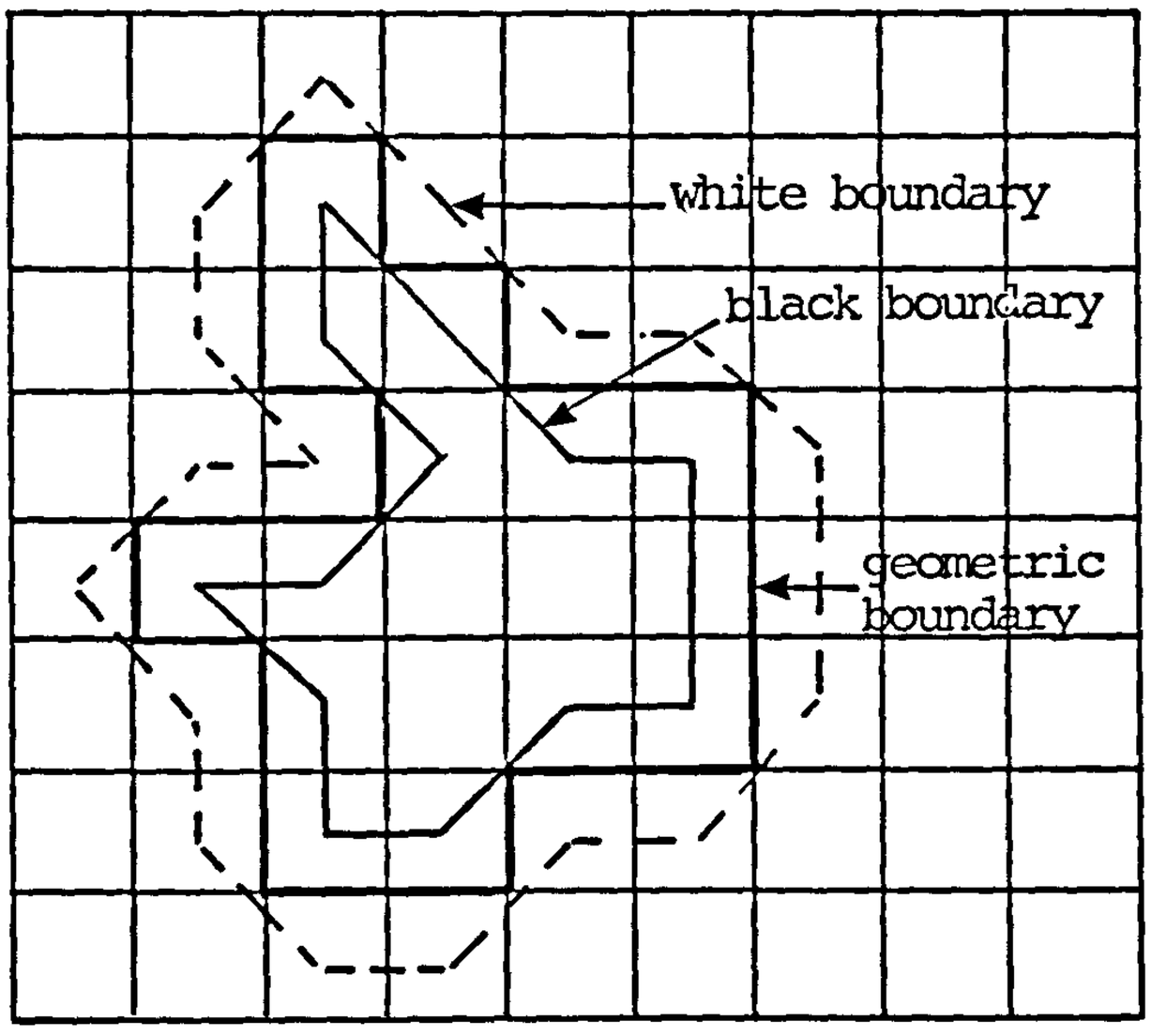


Figure 4.2 The three types of boundary lines as identified for a binary image.

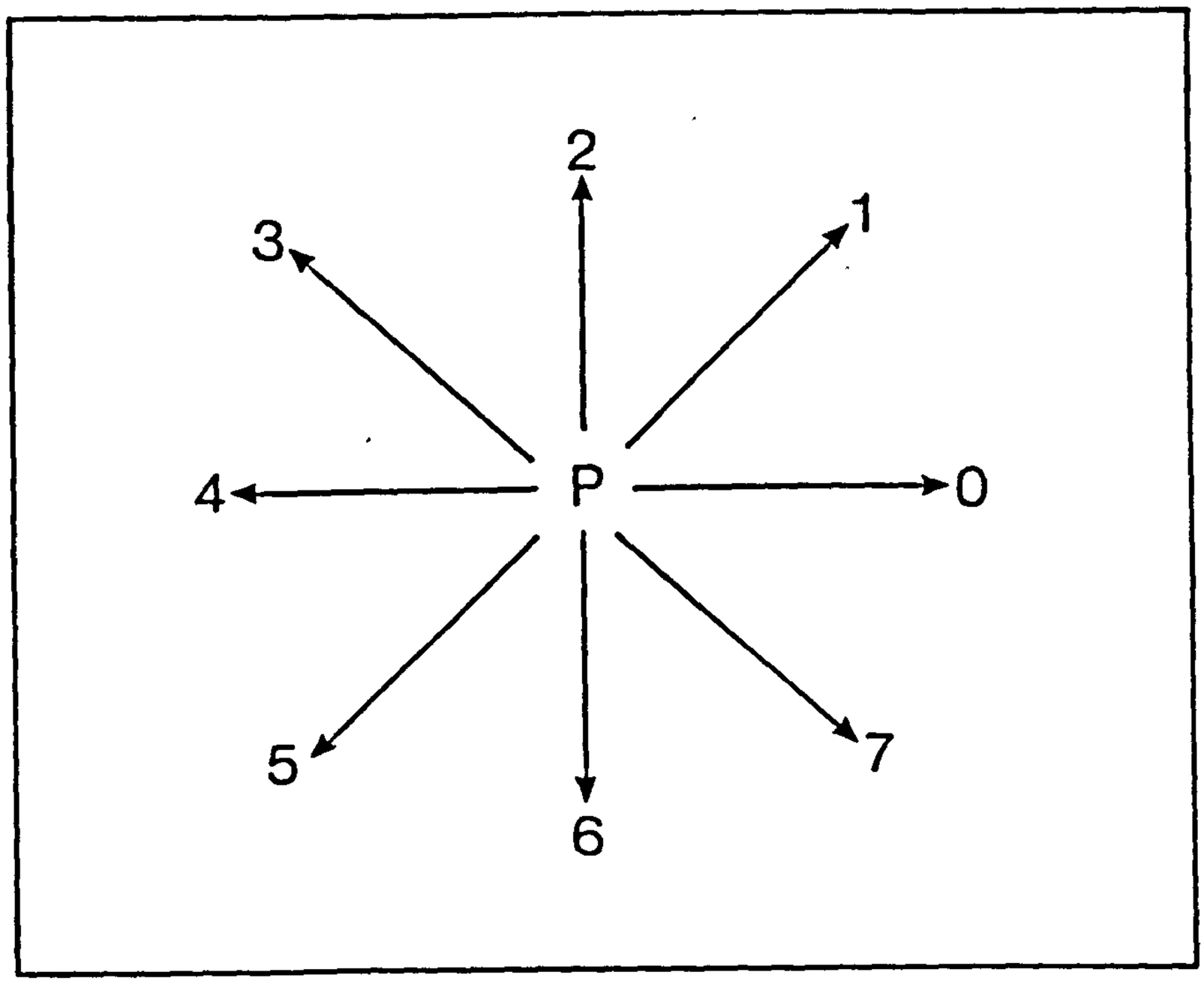


Figure 4.3 Freeman codes.

elements from that of white elements,

(ii) the black boundary elements, the black elements that border on white elements and are obtained by connecting the centres of adjacent black elements,

(iii) the white boundary elements.

It is observed that unlike the geometric boundary line, the upper and lower boundary lines can be either of length equal to the element width or equal to $\sqrt{2}$ times that width. The lines may be collinear or make angles in multiples of 45^{o} with each other.

In the coding scheme of Figure 4.3, the eight possible directions that the boundary may take in going from one boundary element to the next are assigned the octal digits 0 to 7 rotating in a counter-clockwise sense. The resulting number sequence, called the chain of the boundary curve, implicitly preserves the direction of the boundary.

4.3.2 Threshold Selection

The background pixels are separated from the object pixels by a grey level threshold. A number of ways for selecting the threshold value were examined and the approach chosen is based on the analysis of a grey level histogram.

Figure 4.4 shows a typical histogram for a white object on a black background. Two distinct peaks are shown in the figure, one at a grey level of 25, very close to pure black and one at a grey level of 120, a dirty white. If the illumination is well controlled, the best threshold value (mid-way between object and background) will remain sensibly constant and can be chosen interactively during system set up. For the unreliable illumination case, the computer may be required to analyse the distribution of grey levels over the area of interest, mean value plus or minus one standard deviation, to choose an appropriate threshold value.

Incorrect choice of the threshold value will introduce errors in the contour tracing. Too small a value for the threshold level will result in a noisy object boundary; too high a value will cause a loss of information and breaks in the traced contour, these effects are shown in Figure 4.5.

Camera sensitivity does change from the centre of the picture to the edges due to parabolic distortion and/or vignetting, a global threshold may be difficult, therefore to maintain. Figure 4.5(a) shows a strip of white pixels at the left side of the figure and investigation showed this to be due to incident light saturation. By monitoring the level of light i.e. using an oscilloscope to monitor the analogue signal out of the camera, it can be corrected for at system set-up. The three possible lighting conditions are superimposed and presented in Figure 4.6.

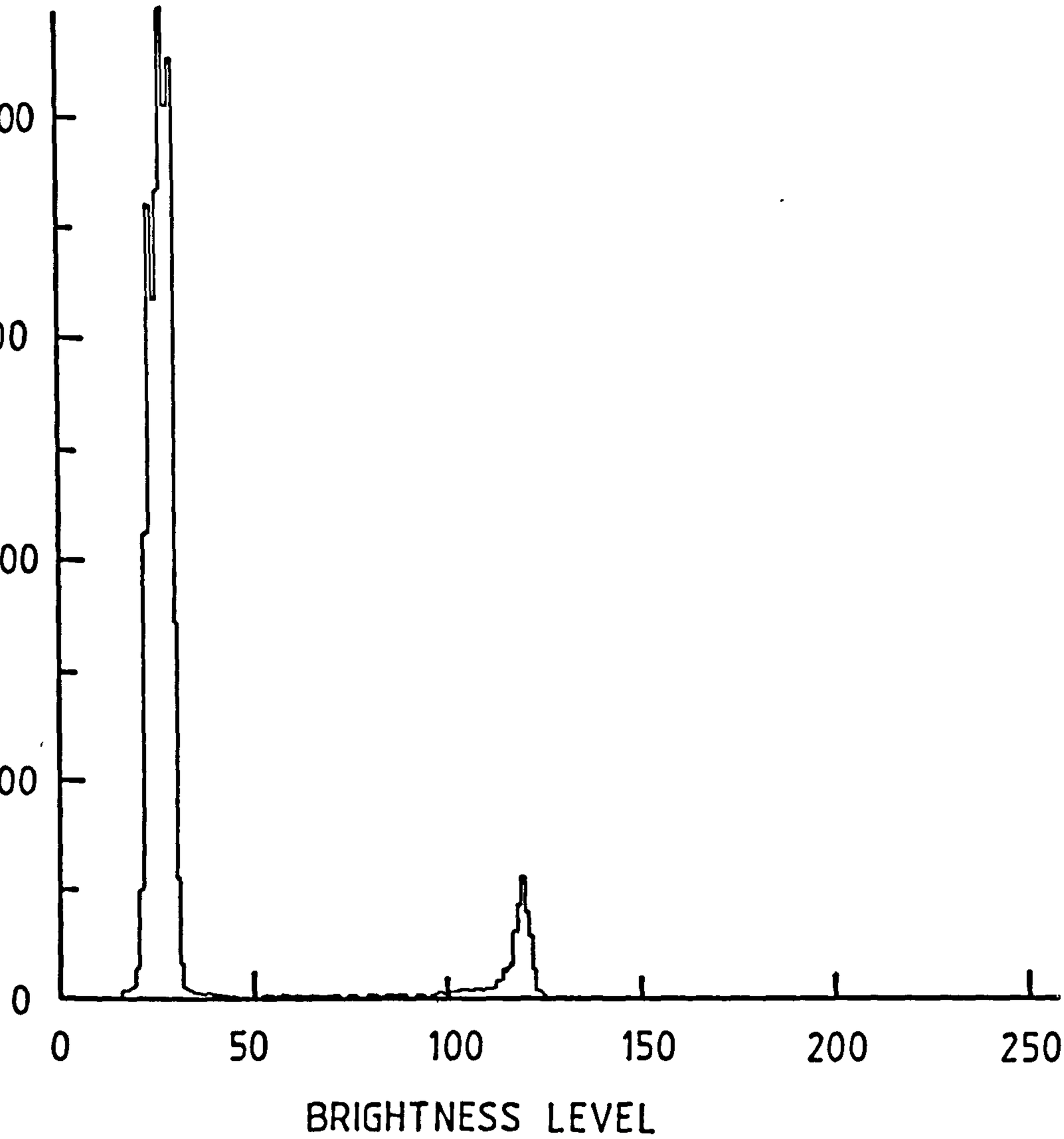


Figure 4.4 Histogram of a picture with a mixture of black and white pixels.

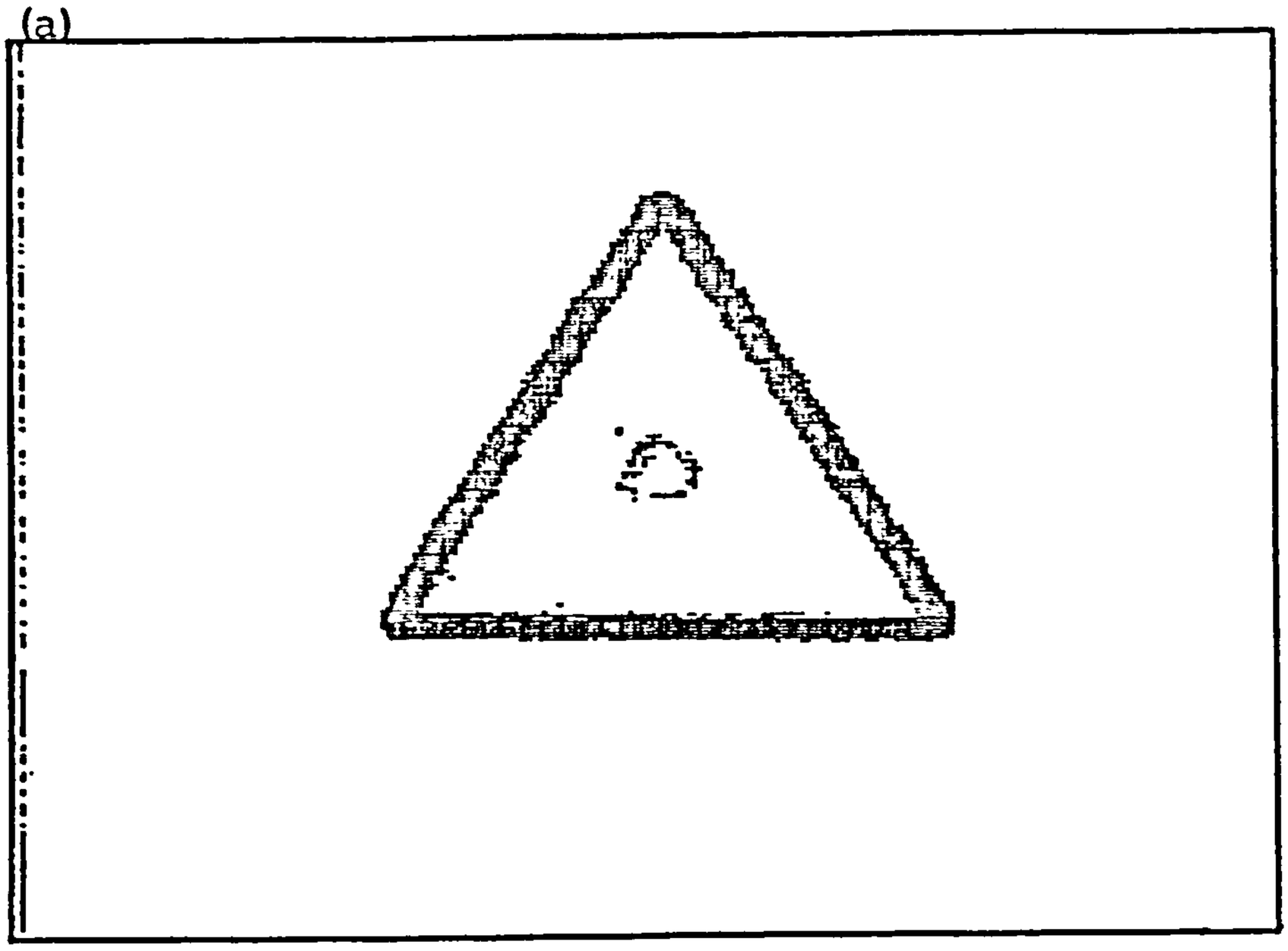


Figure 4.5 Effects of selecting a wrong threshold value
(a) small threshold, noisy contour trace
(b) high threshold value, breaks in contour trace.

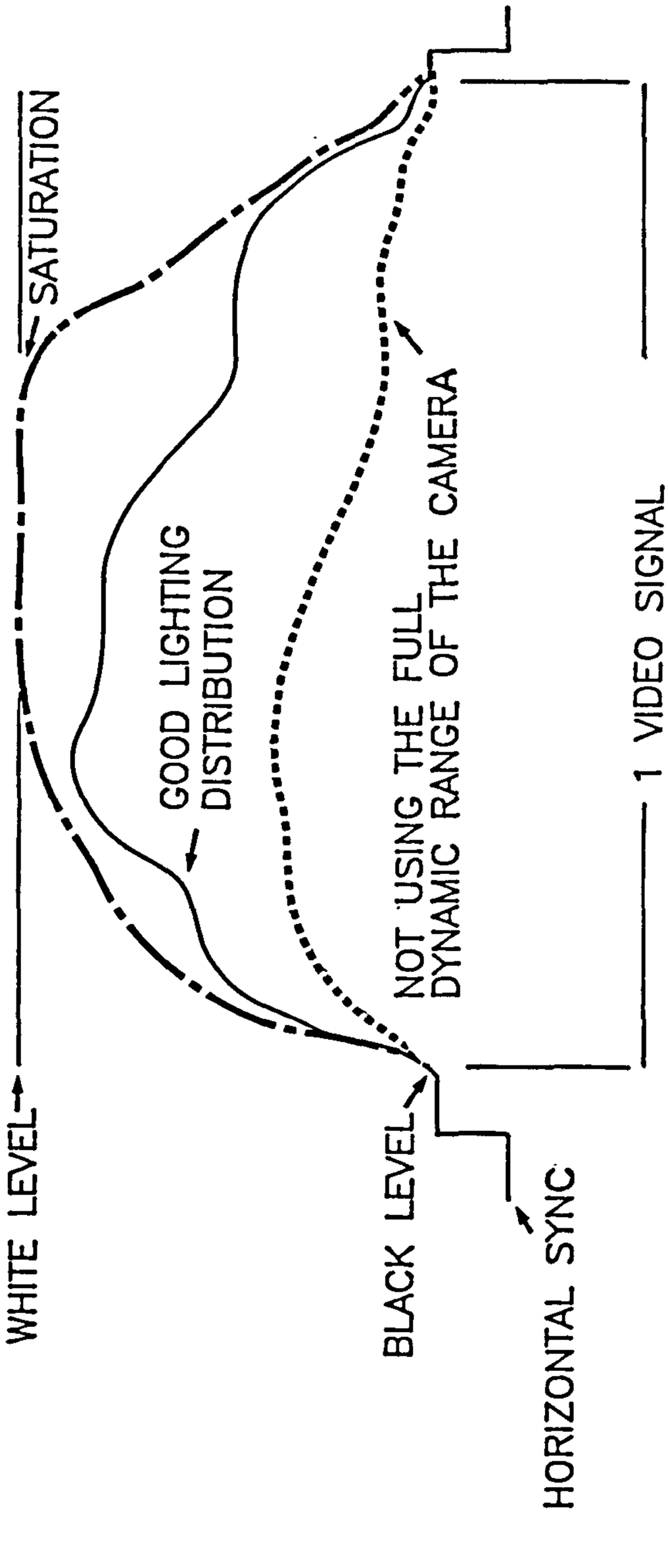


Figure 4.6 Monitoring the illumination, the three possible lighting conditions superimposed.

Initially it was thought that a single threshold would not provide sufficient performance and local thresholds would have to be chosen [5]. It became increasing evident, however, the problems experienced were due largely to light saturation and these problems could be eliminated by:-

(i) controlling the illumination by adjusting the camera's aperture and/or by changing the environment lighting when necessary, or

(ii) disregard the corners and the edges of the frame as invalid data before the start of the boundary search.

4.3.3 Search, Trace and Encoding the Contours

The picture array is scanned top to bottom and right to left until an object pixel is detected (the first contour pixel). From this pixel the object contour is traced in a counter clockwise direction. The elements of the chain code are related to the directions from one contour pixel to the next by the available angles shown in Figure 4.3. The chain code is then stored in an array for further processing and the coordinates of the contour elements are stored in a separate array.

The method adopted by this author for deriving the curvature elements, defined as the angle between the adjacent contour pixels, makes full use of chain encoding system.

The method is based on Table 4.1 which indicates the relative rotation of ordinal elements in the chain code. The derived curvature elements of Table 4.1, are based on:-

ABS (Ordinal No.(n) - Ordinal No. (n-1))

Note that the chain encodes are spatially variant while the curvature elements are rotation invariant. The sequence of curvature elements (L) describes the shape, e.g. $L=l_1, l_2, \ldots .. l_n$, l_n being the last curvature element. This sequence has the following characteristics:

(a) The sequence is periodic with a period of n elements

$$l_{i+n} = l_i \qquad \text{where} \qquad i = 1 \text{ to } n$$

(b) a rotation of an object causes a shift of K places of the curvature elements, i.e. $l_i \longrightarrow f_{i+K}$ where $F=f_i, \ldots\ldots, f_n$ represents the shifted sequence of the curvature elements, the rotation angle depending on the numerical value of K.

Ordinal No. (n-1) \ Ordinal No. (n)	0	1	2	3	4	5	6	7
0	0	135°	90°	45°	*180°	45°	90°	135°
1	135°	0	135°	90°	45°	*180°	45°	90°
2	90°	135°	0	135°	90°	45°	*180°	45°
3	45°	90°	135°	0	135°	90°	45°	*180°
4	*180°	45°	90°	135°	0	135°	90°	45°
5	45°	*180°	45°	90°	135°	0	135°	90°
6	90°	45°	*180°	45°	90°	135°	0	135°
7	135°	90°	45°	*180°	45°	90°	135°	0

*not realizable

(a)

Difference in Ordinal No.	0	1	2	3	4	5	6	7
Curvature Element	0°	135°	90°	45°	*	45°	90°	135°

* not realizable

(b)

Table 4.1 (a) Deriving curvature elements using chain encodes.
(b) Deduction of curvature elements using (a).

4.4 ORIENTATION MEASUREMENT

The reference coded object image can be compared with the actual coded image by defining a similarity term [2]:

$$d(j) = \sum_{i=1}^{n} \left| l_i - f_{i+j} \right| \qquad j=0,\ 1,\ \ldots\ldots,\ n-1$$

where f_{i+j} represents a shift in the f_i element. Shifting all elements by a similar amount is equivalent to the reorientation of the actual object image. Minimisation of the similarity term, Min.d(j), will give the best match between reference object and actual object. The rotation is measured as the angle between the vectors joining the centre of area to the points l_i and f_{i+j} respectively.

The computer flow diagram for estimating the reorientation angle value, after obtaining the "best fit" between the actual and reference objects, is given in Figure 4.7.

4.5 RECOGNITION THRESHOLD

Assuming that the unknown object with the sequence F belongs to the object class of the learnt object with the sequence L, it is most likely that a contour point F_{i+k} is situated in the reference contour interval i to (i±1). As

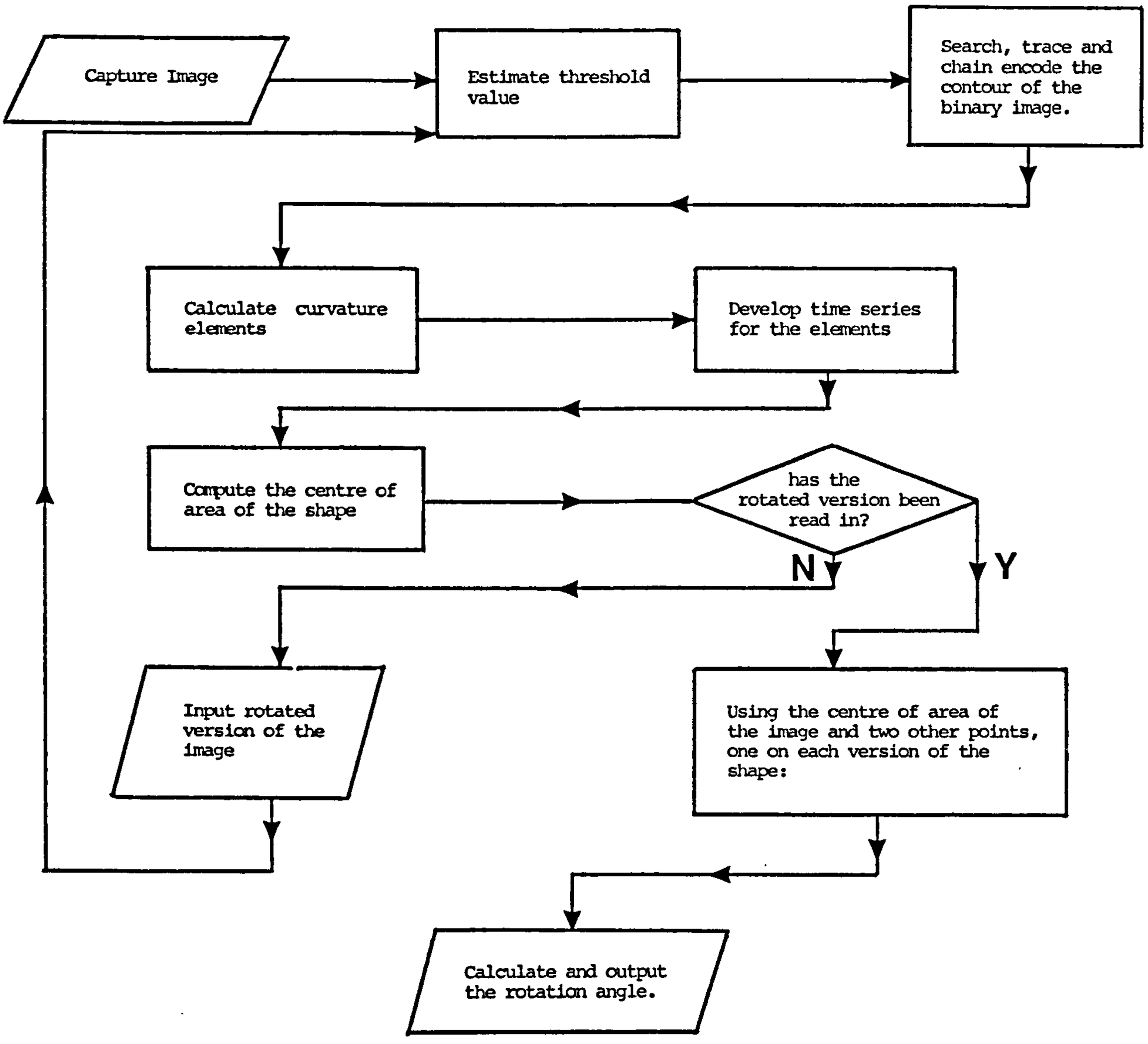


Figure 4.7 Flow diagram for object reorientation estimation.

consecutive curvature elements are correlated, the value f_{i+k} will be found in the interval i to (i+1). The maximum possible difference is measured as:

$$d = \frac{\left| l_i - l_{i+1} \right|}{2}$$

This difference can occur at each point i=1, ..., n and the maximum value for objects belonging to the same class is:

$$sumd = \frac{1}{2} \sum_{i=1}^{n} \left| l_i - l_{i+1} \right|$$

Geister [2] argues that this formula makes no allowance for errors caused by angle computation using a square grid, hence he chooses the rejection threshold to be

$$RTHRS = 2(sumd)$$

This was found to be unreliable for some shapes, and shapes not belonging to the same class were incorrectly recognised. A more reliable threshold, found experimentally, is:

$$RTHRS = 3/2\ (sumd)$$

Figure 4.8 shows the correlation between the curvature elements of various shapes. The correlelograms are obtained by calculating the summation of the difference between the two

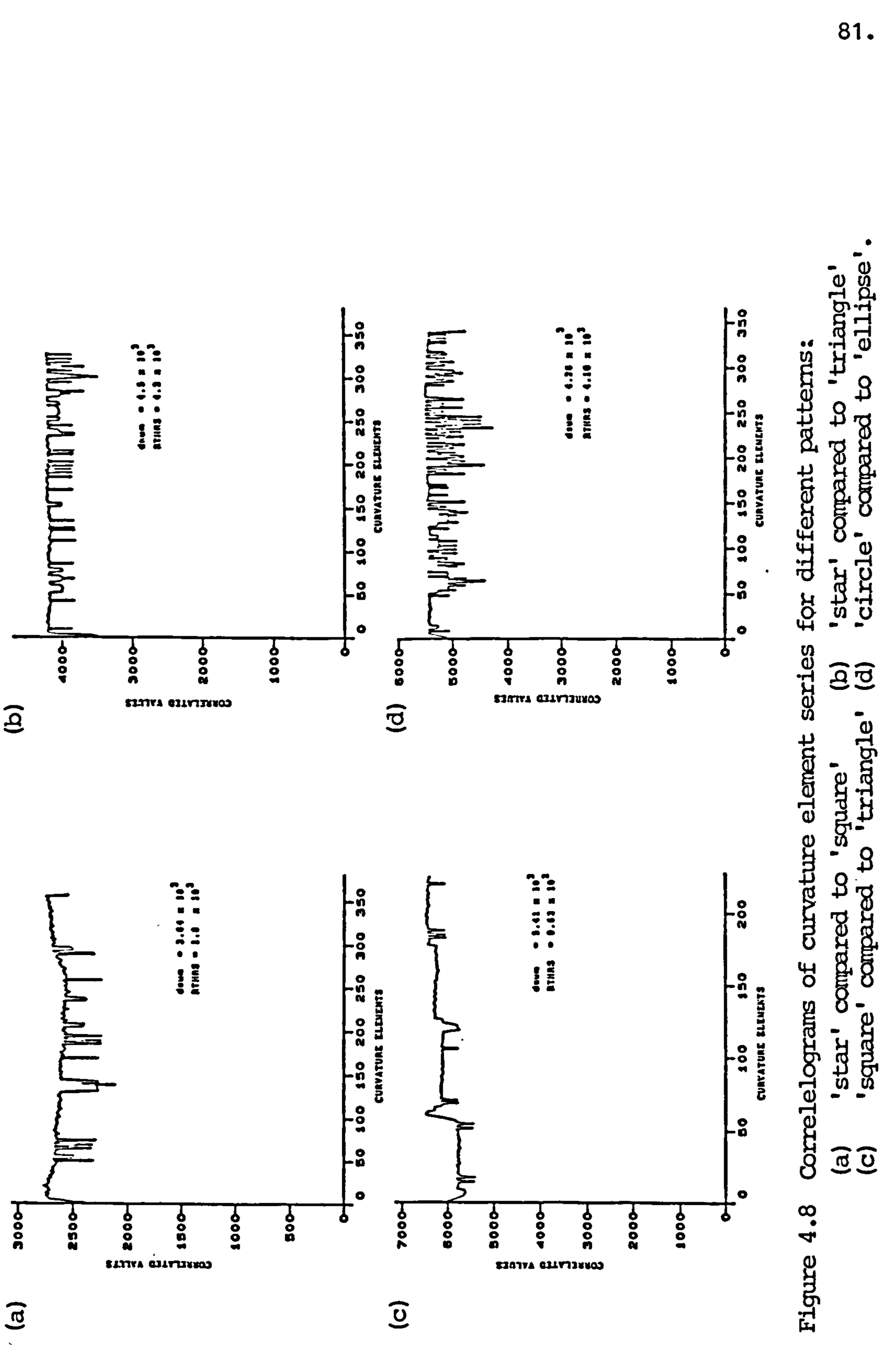


Figure 4.8 Correlograms of curvature element series for different patterns:
(a) 'star' compared to 'square' (b) 'star' compared to 'triangle'
(c) 'square' compared to 'triangle' (d) 'circle' compared to 'ellipse'.

"curvature element series" of the two shapes being compared, keeping the reference series stationary and shifting the one being compared to it an element at a time (thereby simulating a rotation).

As can be seen in Figure 4.8, the two shapes being compared are not in the same class, i.e. two essentially different patterns, the sum of the difference 'd' is greater than 'RTHRS', i.e. dsum > RTHRS, and the new shape is rejected. 'dsum' being the summation of all the differences in the two series of curvature elements.

4.6 RESULTS AND DISCUSSION

There are limitations in using Freeman's model for coding. The most significant are:-

(i) In following the boundary the algorithm searches for changes in directions of one boundary pixel to another, and due to the resolution of the square grid these changes may not always depict the true shape of the object. To avoid confusion the object's area should include at least 1% of the total number of pixels, i.e. sizeable object against the mesh or grid, for 64 K pixel elements, object area always exceeded 640 pixels.

(ii) Changes in perimeter occur after rotating an object due to the square grid approximation, hence there is a feature tolerance before rejecting an object on the basis of having a different perimeter length and area to that of a learnt object. Table 4.2 gives evidence of perimeter length and area changes after rotation due to the grid approximation; changes of up to 8% are observed.

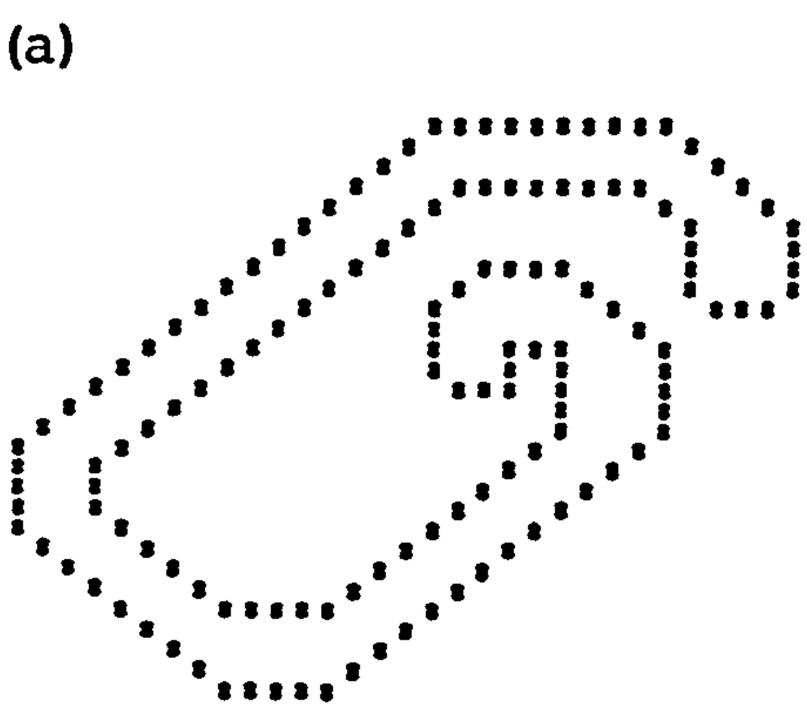
Figure 4.9 shows two arbitrary shapes, 35 by 35 pixels in dimension, identical patterns with one rotated through 90^{o}. This is an example of 'ideal' inputs and is used for testing the algorithms. The curvature elements are derived using Table 4.1, curvature matching is also presented to measure the angle of rotation.

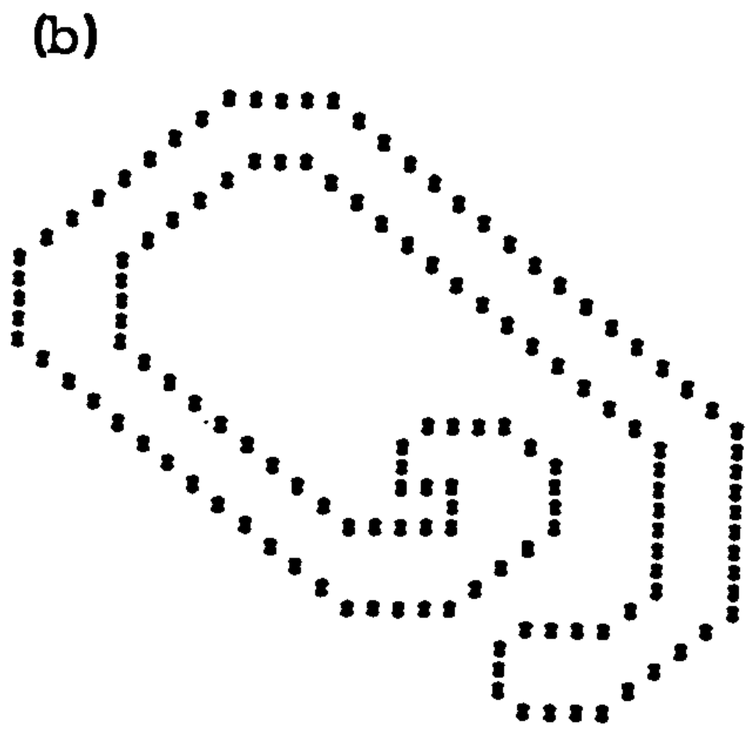
Figure 4.10 represents the matching of the curvatures of two triangles, when one is rotated through 90^{o}, (a) using Table 4.1 to derive the curvature elements, and (b) calculating the curvature elements by measuring the angle between the successive boundary pixels. Figure 4.11 estimates the rotation of the triangle when the actual orientation is 45^{o}.

Orientation estimation, in the trials conducted, based on an average of ten runs, for various rotations, using two identical shapes, is accurate to within 2.5%.

Boundary Shape	Area		Perimeter	
	Before rotation (Pixels)	After rotation (Pixels)	Before rotation (Pixels)	After rotation (Pixels)
Square	8582	8537	400	420
Ellipse	7100	6825	368	398
Triangle	7766	7766	484	484
Star	7194	7805	425	455
Pentagon	8610	8590	495	482

Table 4.2 Area and Perimeter changes as a result of rotation; rotations are either 90^{o} or 45^{o} where most changes were evident.





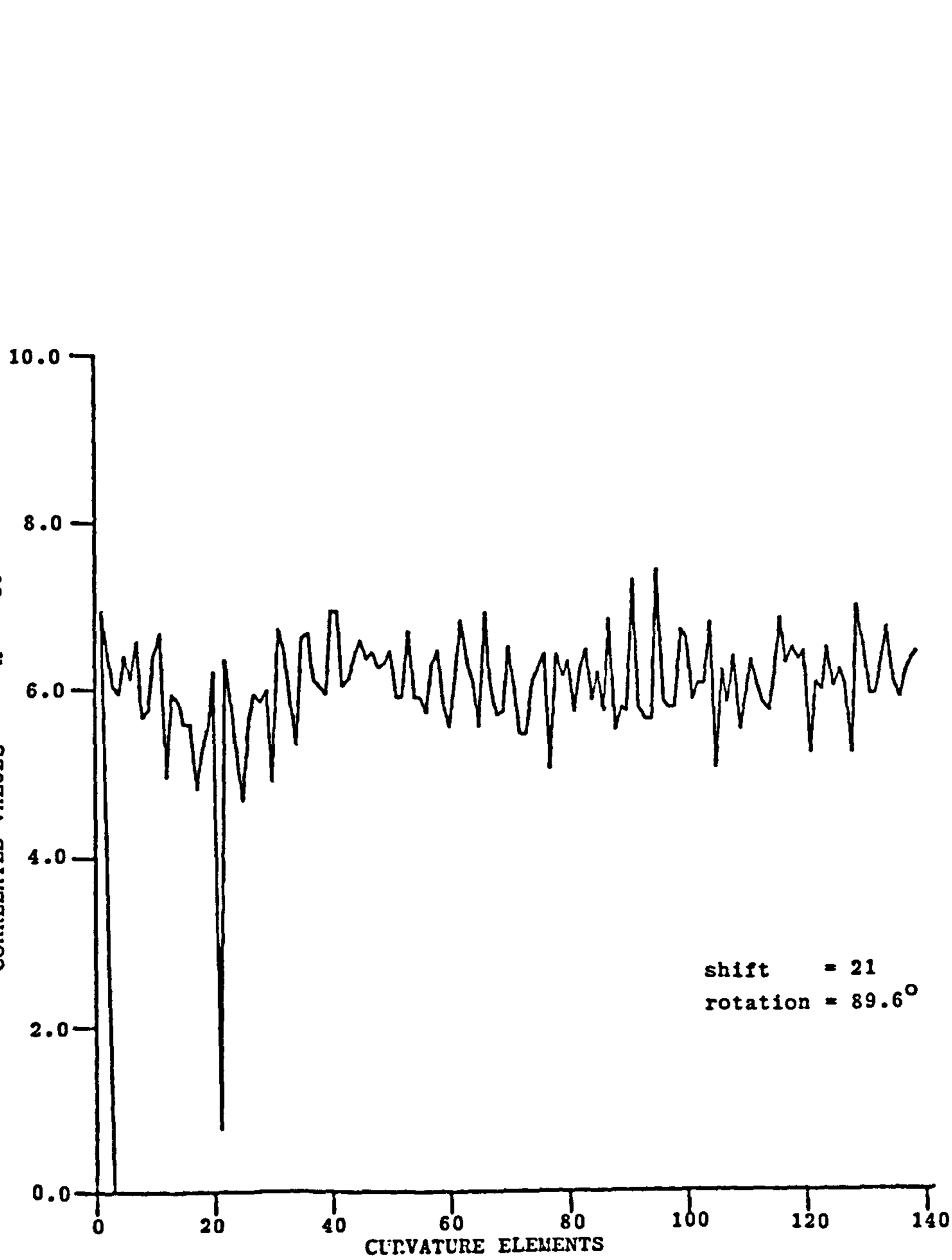
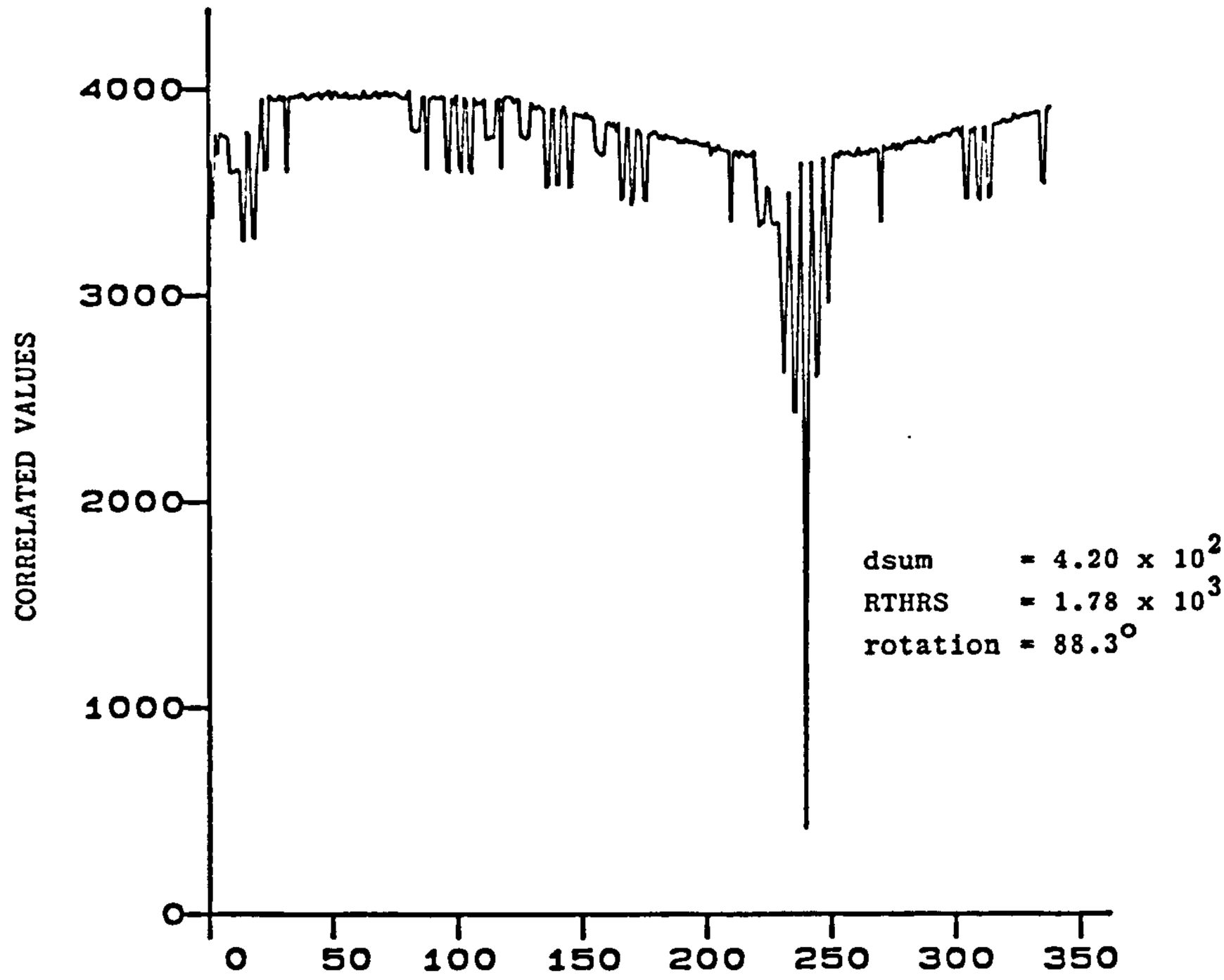
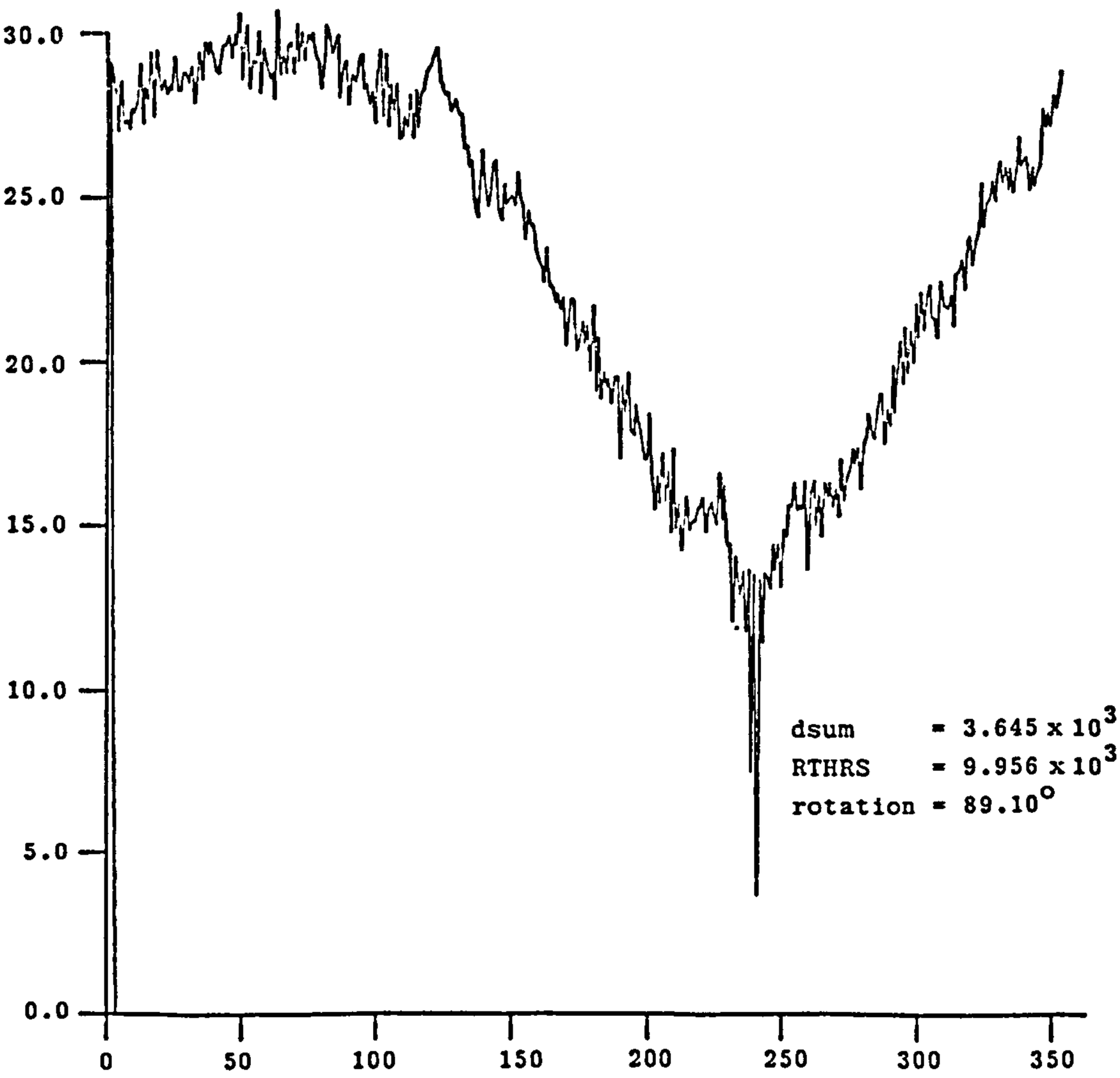


Figure 4.9 Matching the curvature elements of two arbitrary shapes: (a) and (b) 35 x 35 pixels.

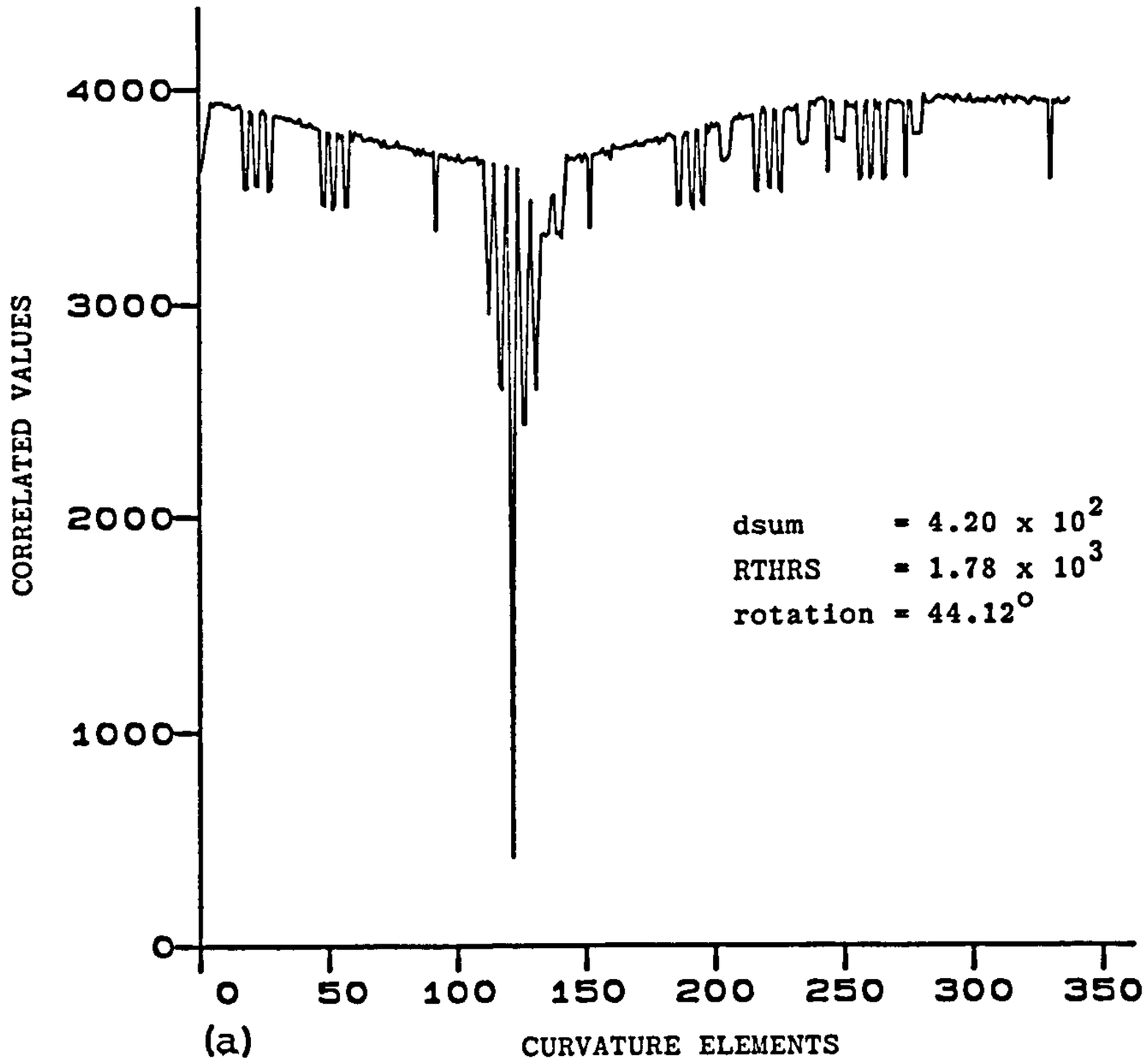


(a)



(b)

Figure 4.10 Comparison of two triangles, i.e. original and rotated through 90° versions,
(a) using table 4.1 and
(b) measuring the angles.



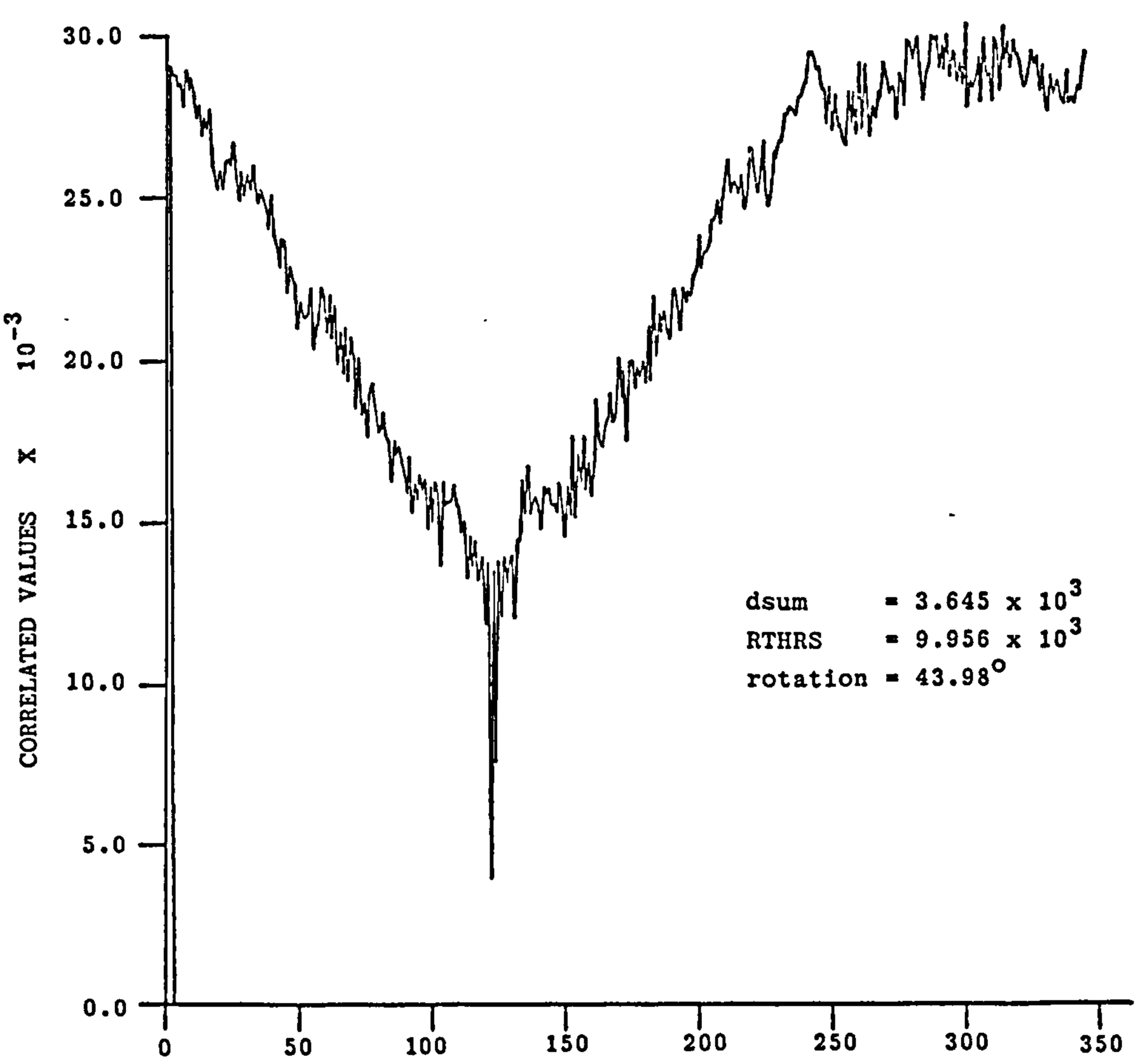


Figure 4.11 Comparison of two triangles, i.e. original and rotated through 45° versions,
(a) using table 4.1 and
(b) measuring the angles.

To increase the efficiency of the algorithms, and to take advantage of the full field of view, the algorithm is made capable of dealing with more than one object in the frame. After detecting and chain encoding a contour boundary, search is continued for more objects in the frame, each boundary detected and successfully chain coded is stored in a separate array location for further processing and comparison with future patterns. In such environments or scenes, the order of the patterns is not important as the new object is compared to each of the separately stored arrays of curvature elements, and the analysis for the best match is carried out. Figure 4.12 represents a scene with four objects for simultaneous analysis.

4.7 CONCLUSIONS

The algorithms developed assume a continuous contour. If there is a break in the object boundary the program terminates with an error message. The algorithm for contour searching is also prone to error if the edges of the shapes are not well defined. It is, therefore, helpful to run an edge detection routine on the shapes before contour searching resumes. Some edge detection routines, as explained and examined in the previous chapter, were tried for this purpose. They included : Robert's operator, Sobel operator and Laplacian (template and gradient) methods; with Robert's operator giving some of the best results in an optimum time.

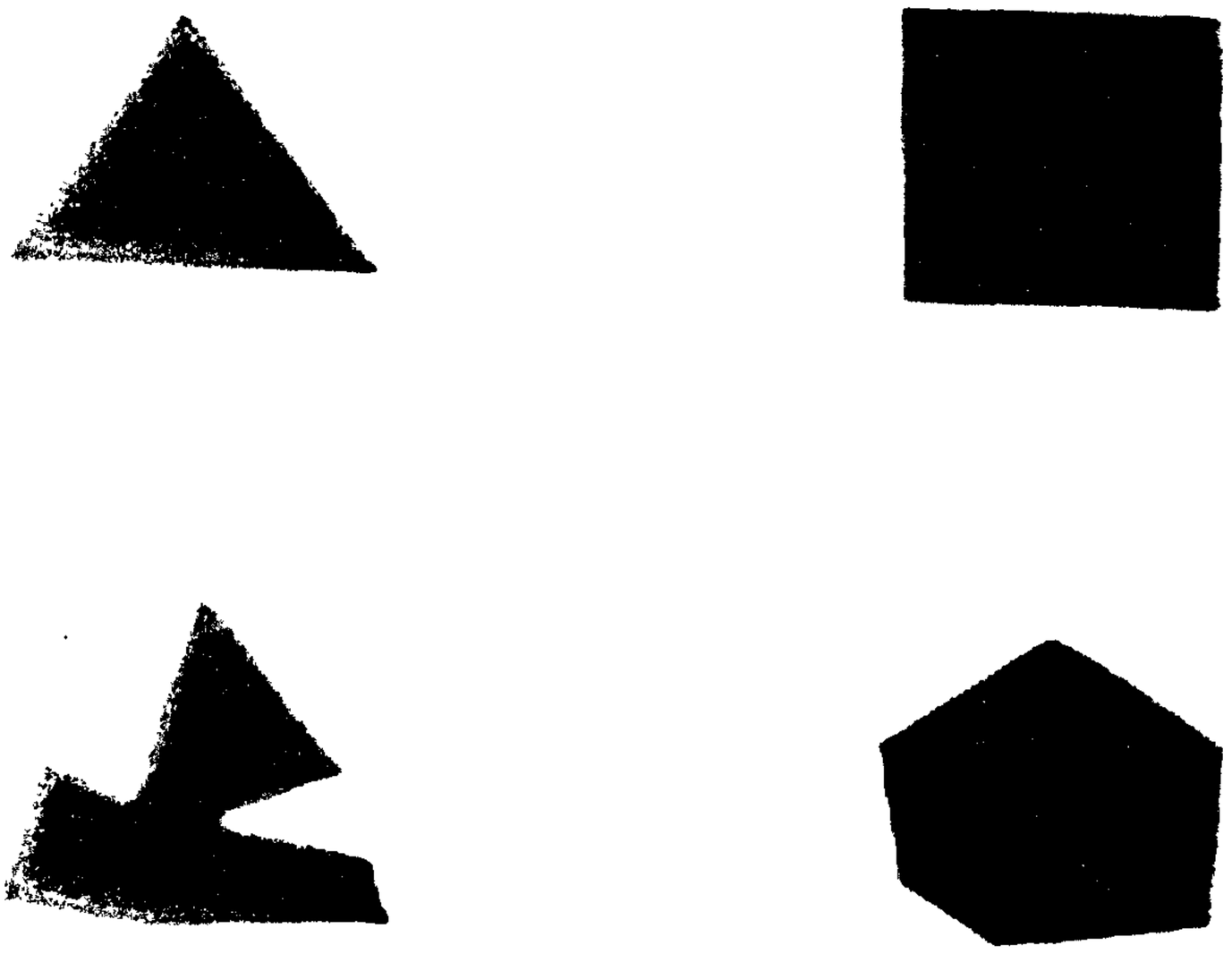


Figure 4.12 A scheme of four objects for recognition analysis.

A further point to note is that the shapes are digitised in a square grid, hence, a rotation causes the number of pixels occupied by the shape to alter, in some cases this can be up to 8% change as shown in the previous section. This may cause an error in the curvature matching method if the total number of contour pixels in both the original and rotated versions are assumed to be the same. As a safe guard, the total number of contour pixels in both the original and rotated versions must be found and the matching calculations should only be taken for as far as the lower number of pixel counts.

The computational time, depending on the method used for feature extraction and edge detection, is relatively short (60 to 100 ms cpu time).

C H A P T E R F I V E

THE FOURIER DESCRIPTOR IN CLOSED CURVE REPRESENTATION FOR OBJECT RECOGNITION

An approach, based on Fourier analysis, for the fast shape recognition of randomly positioned and orientated components is described. Two methods are investigated, with both methods suggesting that a small number of normalised Fourier Coefficients are sufficient to reliably identify and recognise the components.

5.1 INTRODUCTION

The method most frequently used to describe the closed boundary of a manufactured object is either based on the chain code of Freeman [1] or the polygon approximation of Pavlidis [2].

Theoretical and experimental evidence available, however, indicates that the Fourier Descriptor (FD) is a more powerful way to classify a closed contour , [3, 4]. Features which are invariant with respect to translation, rotation and object size (i.e. scale) can be uniquely represented by a normalised FD and this can lead to a significant reduction in object recognition data over the chain code and polygon approximation based methods.

To demonstrate the benefits of this object classification procedure, existing data records, for a number of objects, have been adopted for computing the FD values. Two methods have been considered. In the first, the coordinate values of the contour pixels are used and in the second the actual orientation vectors of the Freeman chain code are employed.

5.2 FOURIER DESCRIPTORS

The development of slope and curvature codes for use in boundary description has led to the more general concept of the intrinsic equation [5]. The nature of contour data satisfies the mathematical constraints for shape representation by Fourier descriptors since any single-valued periodic function may be expressed as a Fourier Transform.

In the first approach using cartesian coordinates, the two discrete series x(m), y(m), m = 0, 1,2,....L-1, are obtained by image segmentation and edge tracking; and since the boundaries are closed curves x(L) = x(0) and y(L) = y(0).

The analysis involves the derivation of the "descriptors" based on the Fourier series for each of the two series x(m) and y(m) defined as

$$x(m) = \sum_{n=-\infty}^{\infty} X(n)\, e^{jn\omega_o m} \qquad (5.1)$$

$$y(m) = \sum_{n=-\infty}^{\infty} Y(n)\, e^{jn\omega_o m} \qquad (5.2)$$

where $\omega_o = \frac{2\pi}{L}$ and X(n) and Y(n) are the complex Fourier coefficients, i.e. $X(n) = a_n - jb_n$. The data interval, used in

the evaluation of the expressions, is taken as the pixel length, and the coefficients are estimated from:

$$X(n) = \frac{1}{L} \sum_{m=o}^{L-1} x(m)\ e^{-jn\omega_o m} \qquad (5.3)$$

$$Y(n) = \frac{1}{L} \sum_{m=0}^{L-1} y(m)\ e^{-jn\omega_o m} \qquad (5.4)$$

where the parameter L is the number of pixels representing the closed contour. It is clear from the analytical derivation, that the Fourier coefficients (X(n), Y(n)) contain no information relating to the translation and orientation of the object. Therefore the descriptor defined as:

$$R(n) = \left[(|X(n)|^2 + |Y(n)|^2) \right]^{\frac{1}{2}} \qquad (5.5)$$

is independent of the orientation and this is confirmed by data given in Table 5.1.

The descriptors defined by Equation (5.5) although invariant to object rotation, are influenced by object size. To overcome this weakness, a normalised descriptor, independent of both orientation and size, can be used and is defined as:

$$S(n) = R(n)/R(1) \qquad (5.6)$$

where R(1) is the first descriptor value computed.

Results to support the use of Equation (5.6) are given in Table 5.2.

a : Fourier Coefficients for 'TRIANGLE'

X_n		Y_n		R_n
Real Part	Imaginary Part	Real Part	Imaginary Part	$(\|X_n\|^2+\|Y_n\|^2)$
-29.36	-0.42	-2.30	23.15	37.46
-6.62	-0.32	-1.12	-5.75	8.85
-0.60	0.11	-1.50	0.43	1.67
-2.56	-0.17	-1.63	1.41	3.35
-1.27	-0.06	-1.31	-0.84	2.01
-0.64	0.07	-1.48	0.27	1.64
-1.18	-0.09	-1.54	0.50	2.00
-0.77	0.01	-1.34	-0.21	1.56
-0.58	0.05	-1.46	0.24	1.59
-0.86	-0.03	-1.51	0.28	1.76
-0.60	0.04	-1.40	-0.01	1.52
-0.57	0.07	-1.45	0.26	1.58
-0.71	-0.02	-1.44	0.25	1.62
-0.54	0.09	-1.38	0.08	1.49
-0.56	0.05	-1.44	0.24	1.57

b : Fourier Coefficients for 'TRIANGLE Rotated' by 60^{o}

X_n		Y_n		R_n
Real Part	Imaginary Part	Real Part	Imaginary Part	$(\|X_n\|^2+\|Y_n\|^2)$
-20.33	-12.73	-16.31	24.26	37.81
-6.10	2.37	-3.46	-5.63	9.29
-0.92	0.42	-0.90	-0.07	1.35
-1.91	-0.89	-2.29	1.55	3.47
-1.73	0.26	-1.17	-0.70	2.22
-0.90	0.24	-0.85	0.02	1.27
-1.14	-0.27	-1.47	0.47	1.94
-1.17	0.07	-1.01	-0.19	1.56
-0.88	0.20	-0.90	0.07	1.28
-0.93	-0.09	-1.24	0.26	1.58
-0.99	0.10	-0.94	-0.01	1.37
-0.90	0.19	-0.88	0.10	1.28
-0.89	0.00	-1.13	0.17	1.45
-0.91	0.10	-0.95	0.08	1.32
-0.88	0.16	-0.88	0.15	1.26

Table 5.1 Influence of rotation on Fourier Coefficients.

'TRIANGLE'

R_n	S_n
37.46	1.00
8.85	0.24
1.67	0.04
3.35	0.09
2.01	0.05
1.64	0.04
2.00	0.05
1.56	0.04
1.59	0.04
1.76	0.05
1.52	0.04
1.58	0.04
1.62	0.04
1.49	0.04
1.57	0.04

'TRIANGLE Rotated'

R_n	S_n
37.81	1.00
9.29	0.25
1.35	0.04
3.47	0.09
2.22	0.06
1.27	0.03
1.94	0.05
1.56	0.04
1.28	0.03
1.58	0.04
1.37	0.04
1.28	0.03
1.45	0.04
1.32	0.03
1.26	0.03

'L' Shape

R_n	S_n
55.06	1.00
15.60	0.28
7.46	0.14
5.41	0.10
2.20	0.04
0.36	0.01
1.63	0.03
0.68	0.01
1.02	0.02
0.99	0.02
0.70	0.01
0.37	0.01
0.60	0.01
0.34	0.01
0.52	0.01

'half sized L'

R_n	S_n
28.22	1.00
7.92	0.28
4.30	0.15
2.79	0.10
1.81	0.06
0.86	0.03
1.42	0.05
0.89	0.03
1.16	0.04
0.97	0.03
1.04	0.04
0.84	0.03
0.98	0.03
0.82	0.03
0.92	0.03

'Quarter sized L'

R_n	S_n
15.42	1.00
4.30	0.28
3.28	0.21
2.16	0.14
2.16	0.14
1.66	0.11
1.90	0.12
1.62	0.11
1.74	0.11
1.58	0.10
1.63	0.11
1.50	0.10
1.54	0.10
1.43	0.09
1.44	0.09

Table 5.2 Effect of scaling or contour dilation on Fourier coefficients

5.3 RECONSTRUCTION OF BOUNDARIES USING FOURIER DESCRIPTORS

It is not obvious from Table 5.1 how many Fourier coefficients will be necessary to enable an object to be identified with confidence. In an attempt to clarify this point, Equations (5.1) and (5.2) with a finite number of coefficients, are used to recreate varying approximations to the original object data. Although these expressions will not provide enough information to allow a virtually continuous series to be obtained, they will allow all the discrete values of x(m) and y(m) to be regained exactly if all the Fourier coefficients are employed.

Figure 5.1 illustrates the reconstruction of a 'triangle' for different numbers of terms in the expressions,Equations(5.1) and (5.2). These figures suggest that no sensible improvement will result from the use of terms in excess of 2 for this shape. A subjective assessment of Table 5.1 would support these results because of the dominance of the first two terms over the remainder listed. The results demonstrate very clearly the compression of the boundary data that can be expected and illustrates the power of the descriptor as an aid to simple shape recognition. In this example six hundred pairs of contour coordinate values have been compressed to two descriptor values.

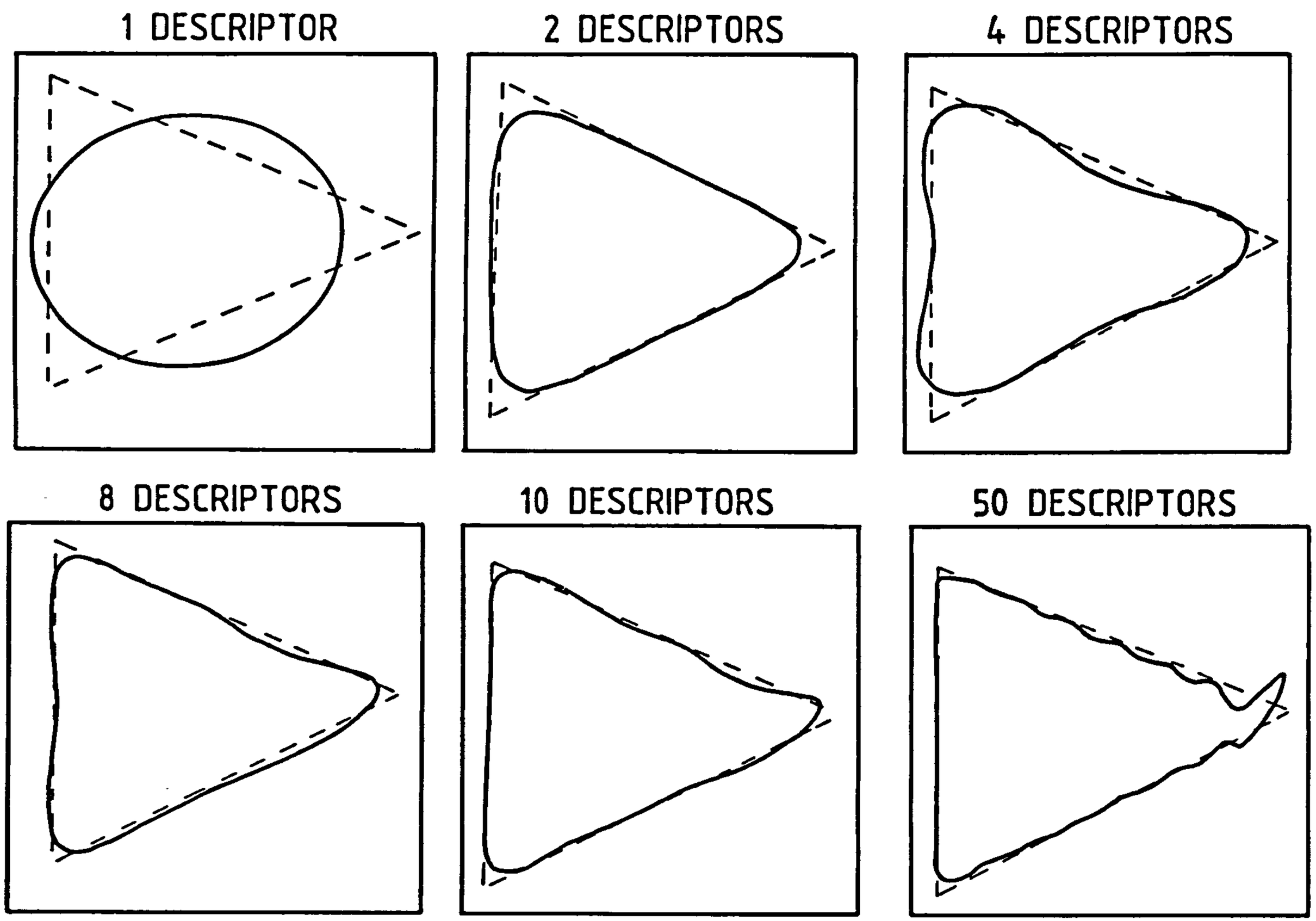


Figure 5.1 Reconstruction of triangle using first method.

5.4 SIMILARITY MEASUREMENT AND RECOGNITION ALGORITHM

The normalised Fourier descriptor vector S for a particular boundary shape could change marginally as a result of scaling and rotation, and to accommodate this likely variation an improved reference feature vector can be obtained by averaging the S descriptor vector that is,

$$Fr_n = \frac{1}{N} \sum_{i=1}^{N} S_i \qquad (5.7)$$

where Fr_n is the reference feature vector whose elements are the normalised Fourier Descriptors averaged over N object specimens of different sizes and orientation. Identifying the feature vector of the test shape as St_n, the similarity between the two independent vectors, Fr_n and St_n, is estimated from the error:

$$E = \frac{1}{K} \left[\sum_{i=1}^{K} (Fr_i - St_i)^2 \right]^{\frac{1}{2}} \qquad (5.8)$$

where K is the number of elements in each vector. In an environment where more than one shape is to be tested for recognition, the parameter E must be evaluated for all the test vectors, the minimum value found yielding the recognised shape.

In the test study conducted, 10 descriptors in each feature vector were adopted and the results are shown in Figure 5.2. In the first, Figure 5.2 (a) a triangle is compared with the

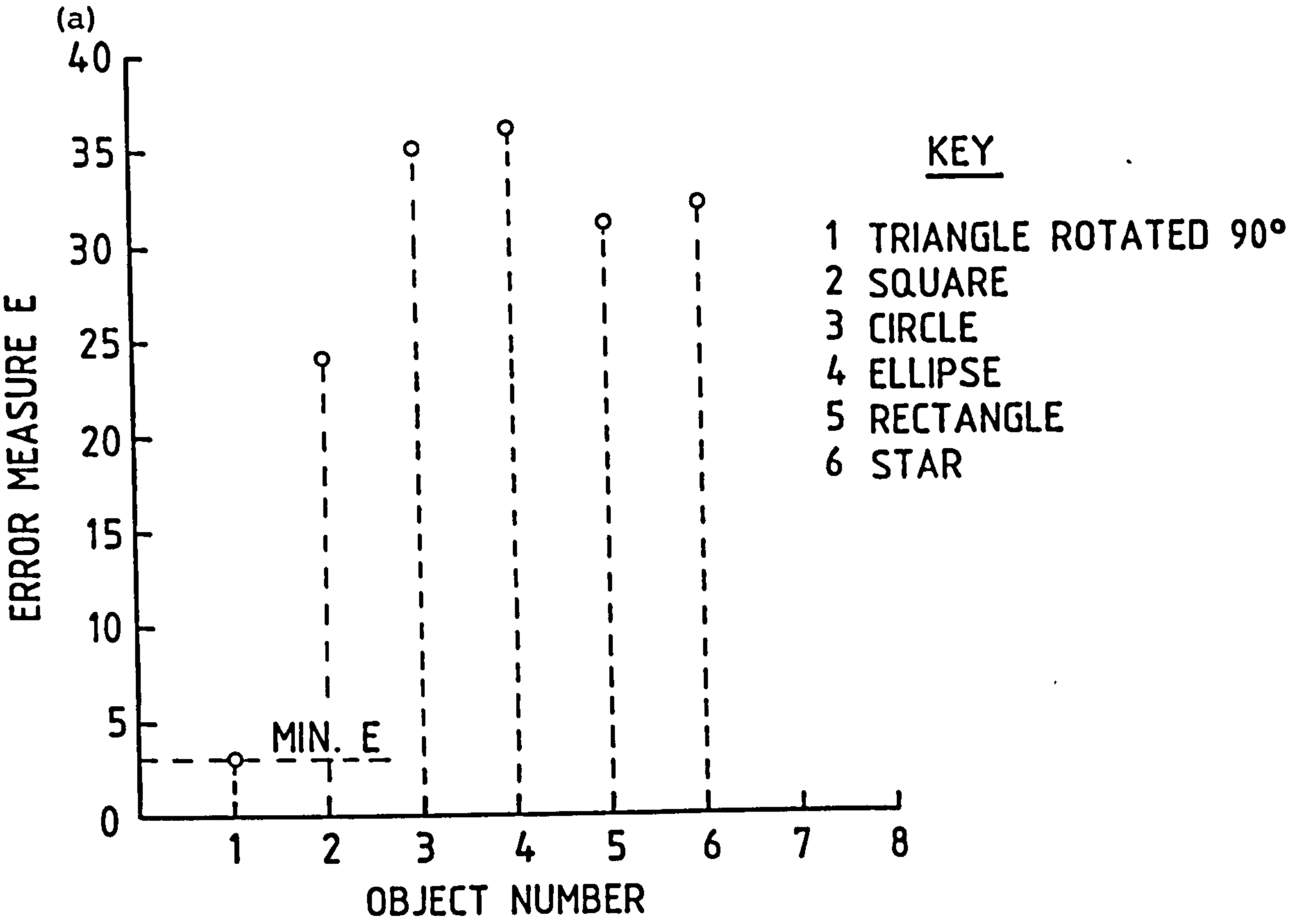


Figure 5.2 Object recognition using the similarity measurement.

other simple shapes listed and the minimum value of E_m clearly indicates the presence of the rotated triangle. In Figure 5.2 (b) the outcome of a trial with an 'L' shaped contour, shows it to be a member of the family labelled 1 and 2 in the figure. The corresponding FD's are listed in Table 5.2. in the raw and normalised form. Appendix A2 gives a program listing for boundary search, orientation estimation, FD calculations and the similarity measurements between two objects.

5.5 DERIVATION OF THE FOURIER DESCRIPTORS USING THE ORIENTATION VECTORS OF THE FREEMAN CODE

To take into account all the information stored in a Freeman chain code contour representation, allowance is made, in the second approach, for the perimeter length between contour pixels.

Consider the chain code C with k elements:

$$C = a_1\ a_2\ a_3\ a_4\ \ldots\ldots\ a_k$$

where each element a_i is an integer number between 0 and 7. The direction of the vector labelled a_i is given by:

$$(a_i) = (\pi/4)a_i$$

If the perimeter length of a contour is t, the length of each

code vector, i.e. link, Δt_i is:

$$\Delta t_i = 1 \quad \text{if } a_i \text{ is an even number}$$
$$\Delta t_i = \sqrt{2} \quad \text{if } a_i \text{ is an odd number} \tag{5.9}$$

From the definition of the chain code, the 'time' required to traverse the first q links at 'constant unit speed' as proposed in [6] is given by

$$t_q = \sum_{i=1}^{q} \Delta t_i \tag{5.10}$$

and the period of the chain code is defined as t_k. The changes in the x, y coordinate values, as the chain elements a_i are traversed are:

$$\Delta x_i = \text{Sgn}\ (6 - a_i)\ \text{Sgn}\ (6 - a_i)$$
$$\Delta y_i = \text{Sgn}\ (4 - a_i)\ \text{Sgn}\ (a_i)$$

where

$$\text{Sgn}(Z) = \begin{cases} 1 & ; \quad Z > 0 \\ 0 & ; \quad Z = 0 \\ -1 & ; \quad Z < 0 \end{cases}$$

If the first chain code element is positioned at the

origin of the x - y axes, the projections on the x and y axis of the first q links are:

$$x_q = \sum_{i=1}^{q} \Delta x_i \qquad (5.11)$$

$$y_q = \sum_{i=1}^{q} \Delta y_i \qquad (5.12)$$

respectively.

Adopting the Fourier representation for a discrete series, the derivative of x(t) is given by:

$$\dot{x}(t) = \sum_{n=1}^{\infty} \left(\alpha_n \cos \frac{2\pi n}{t_k} t + \beta_n \sin \frac{2\pi n}{t_k} t \right) \qquad (5.13)$$

where

$$\alpha_n = \frac{2}{t_k} \int_0^{t_k} \dot{x}(t) \cos \frac{2\pi n}{t_k} t \; dt$$

$$\beta_n = \frac{2}{t_k} \int_0^{t_k} \dot{x}(t) \sin \frac{2\pi n}{t_k} t \; dt$$

then for the interval $t_{q-1} \leq t \leq t_q$

$$\alpha_n = \frac{2}{t_k} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \int_{t_{q-1}}^{t_q} \cos \frac{2\pi n}{t_k} t \; dt$$

$$= \frac{-1}{\pi n} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \left(\sin \frac{2\pi n}{t_k} t_q - \sin \frac{2\pi n}{t_k} t_{q-1} \right)$$

and similarly

$$\beta_n = \frac{1}{\pi n} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \left(\cos \frac{2\pi n}{t_k} t_q - \cos \frac{2\pi n}{t_k} t_{q-1} \right)$$

The time derivative of the Fourier series for x(t) can (by definition) be written

$$\dot{x}(t) = \frac{2\pi n}{t_k} \sum_{n=1}^{\infty} \left(a_n \sin \frac{2\pi n}{t_k} t - b_n \cos \frac{2\pi n}{t_k} t \right) \quad (5.14)$$

and equating like terms in Equations (5.13) and (5.14), yields the terms of the complex Fourier coefficients a_n and b_n

$$\beta_n = \frac{2\pi n}{t_k} a_n \text{ , and on rearrangement this gives}$$

$$a_n = \frac{t_k}{2\pi^2 n^2} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \left(\cos \frac{2\pi n}{t_k} t_q - \cos \frac{2\pi n}{t_k} t_{q-1} \right) \quad (5.15)$$

Similarly,

$$b_n = \frac{t_k}{2\pi^2 n^2} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \left(\sin \frac{2\pi n}{t_k} t_q - \sin \frac{2\pi n}{t_k} t_{q-1} \right) \quad (5.16)$$

For the y(t) series, expressions having the same form can be obtained.

Applying these expressions to a "triangle" and a "wrench" generated the results shown in Figure 5.3 and Figures 5.4(a) and 5.4(b); these demonstrate that a progressive improvement in 'fit' can be achieved with an increasing number of descriptors. The degree of similarity between objects will dictate the number of Fourier descriptors necessary to identify an object. The associated descriptor values are given in Table 5.3 for information.

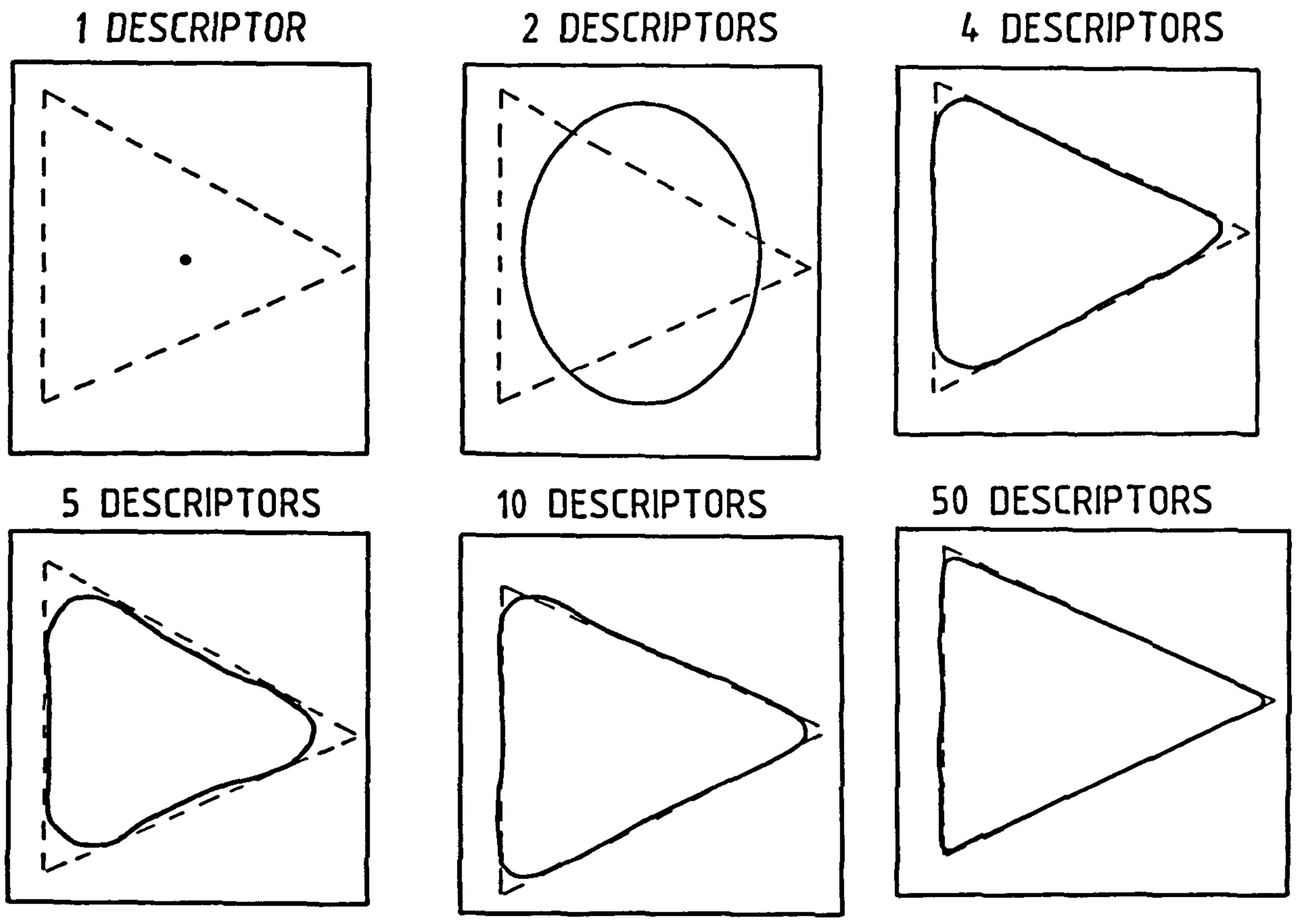
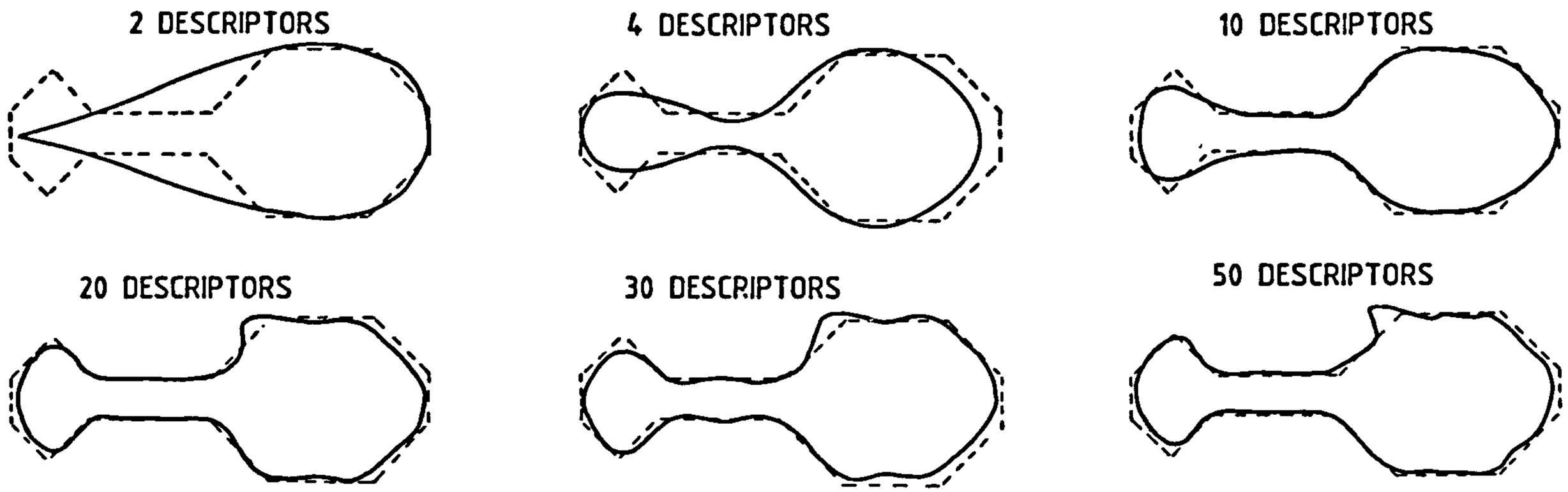
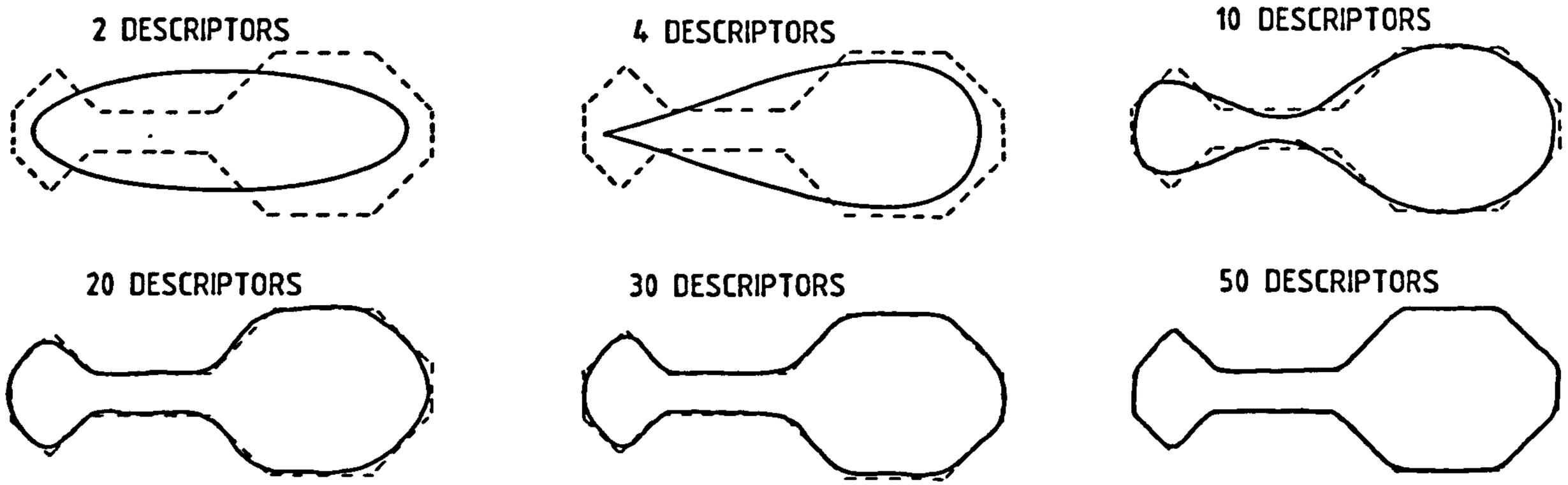


Figure 5.3 Reconstruction of triangle using second method.



(a) Reconstruction of wrench using first method.



(b) Reconstruction of wrench using second method.

Figure 5.4 Reconstruction of wrench using both methods.

	TRIANGLE		WRENCH	
n	R_n	S_n	R_n	S_n
1	73.562	1	99.407	1
2	16.528	0.225	12.643	0.127
3	2.696	0.037	15.536	0.156
4	4.648	0.063	6.603	0.066
5	2.023	0.028	2.559	0.026
6	1.162	0.016	1.549	0.016
7	1.414	0.020	2.254	0.023
8	0.651	0.009	0.810	0.008
9	0.696	0.009	1.854	0.019
10	0.643	0.009	1.163	0.018
11	0.226	0.003	0.302	0.003
12	0.528	0.007	0.768	0.008
13	0.283	0.004	0.337	0.003
14	0.096	0.001	0.056	0.000
15	0.387	0.005	0.342	0.003
16	0.142	0.002	0.478	0.005
17	0.062	0.001	0.437	0.004
18	0.240	0.003	0.082	0.000
19	0.118	0.002	0.112	0.001
20	0.073	0.001	0.290	0.003
21	0.161	0.002	0.297	0.003
22	0.062	0.001	0.164	0.002
23	0.115	0.002	0.142	0.001
24	0.085	0.001	0.109	0.001
25	0.055	0.001	0.160	0.002
26	0.093	0.001	0.155	0.002
27	0.036	0.000	0.220	0.002
28	0.059	0.001	0.047	0.000
29	0.060	0.001	0.084	0.000
30	0.061	0.001	0.172	0.002

Table 5.3 A list of 30 Fourier Descriptors for the triangle and wrench Raw and Normalised versions.

5.6 ERROR APPROXIMATION AS A FUNCTION OF THE NUMBER OF FOURIER COEFFICIENTS EMPLOYED

To obtain an indication of the sub-optimum number of descriptors necessary to ensure object recognition, an error function has been developed.

Let $$\hat{x}_N(t) = a_o + \sum_{n=1}^{N} a_n \cos \frac{2n\pi}{T} t + b_n \sin \frac{2n\pi}{T} t$$

$$\hat{y}_N(t) = c_o + \sum_{n=1}^{N} c_n \cos \frac{2n\pi}{T} t + d_n \sin \frac{2n\pi}{T} t$$

be the Fourier series truncated after N terms for the x(t) and y(t) series and let the actual error ΔE_a be defined as:

$$\Delta E_a = \max_{t=1}^{k} \left[|x(t) - \hat{x}_N(t)|, |y(t) - \hat{y}_N(t)| \right] \quad (5.17)$$

where k is the number of chain code elements available.

To predict the error directly from the chain code data it has been shown by Giardina and Kohl [6] that ΔE can be approximated by the expression:

$$\Delta E_p \leqslant \frac{T}{2\pi^2 N} \max \left[\underset{o}{\overset{T}{V}} (\dot{x}(t)), \underset{o}{\overset{T}{V}} (\dot{y}(t)) \right] \quad (5.18)$$

where the total variation of the 'time' derivative $\dot{x}(t)$ has been symbolised as $\underset{o}{\overset{T}{V}} (\dot{x}(t))$ and of the derivative $\dot{y}(t)$ as $\underset{o}{\overset{T}{V}} (\dot{y}(t))$.

In this context the time period T is equal to t_k. These derivative values are estimated from:

$$\dot{x}_i = \frac{\Delta x_i}{\Delta t_i} \, , \; \dot{y}_i = \frac{\Delta y_i}{\Delta t_i} \text{ for the Freeman code element of } a_i$$

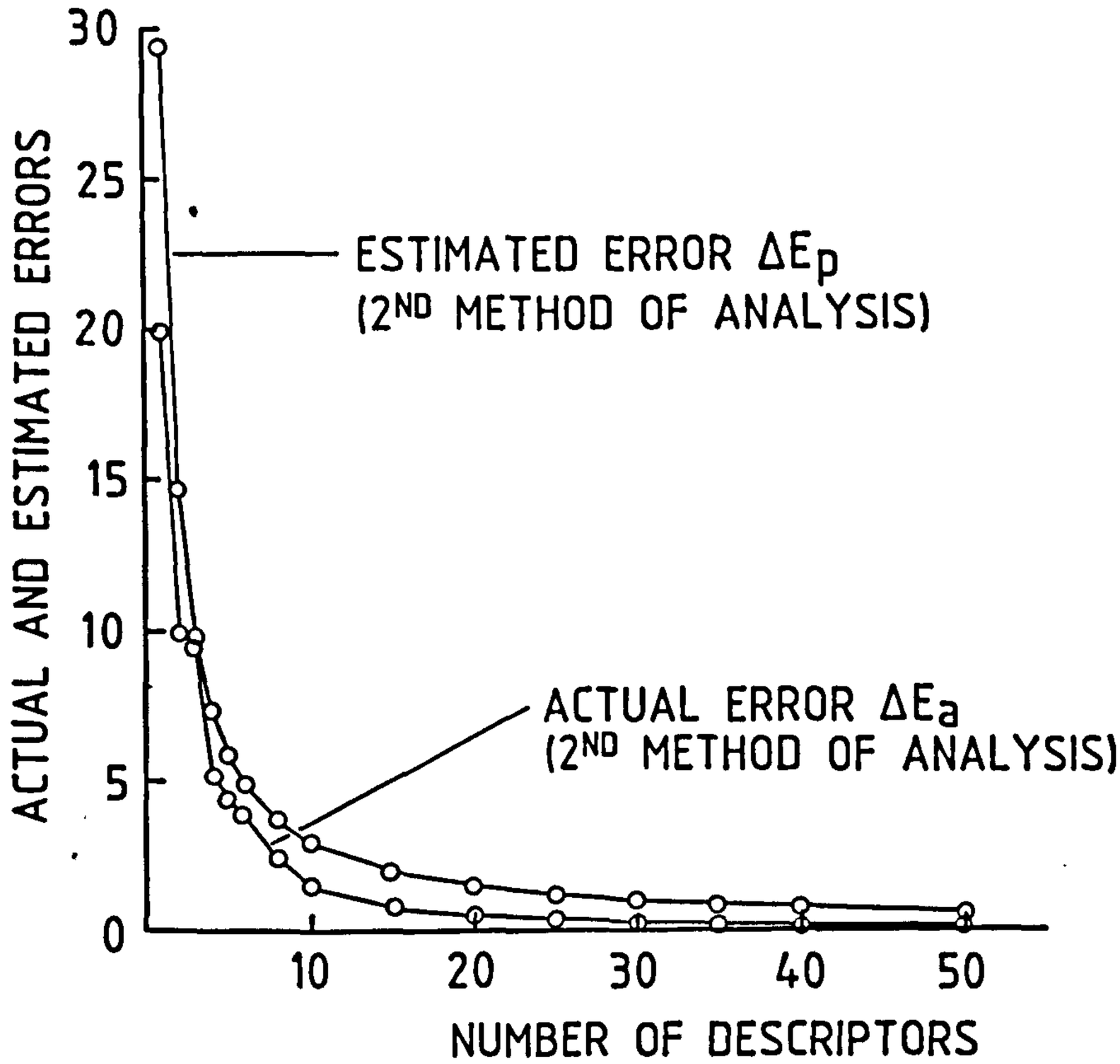
The total variations of $\dot{x}(t)$ and $\dot{y}(t)$ are

$$V_o^T(\dot{x})(t)) = \left(\sum_2^k |\dot{x}_i - \dot{x}_{i-1}| \right) + |\dot{x}_k - \dot{x}_1| \, ,$$

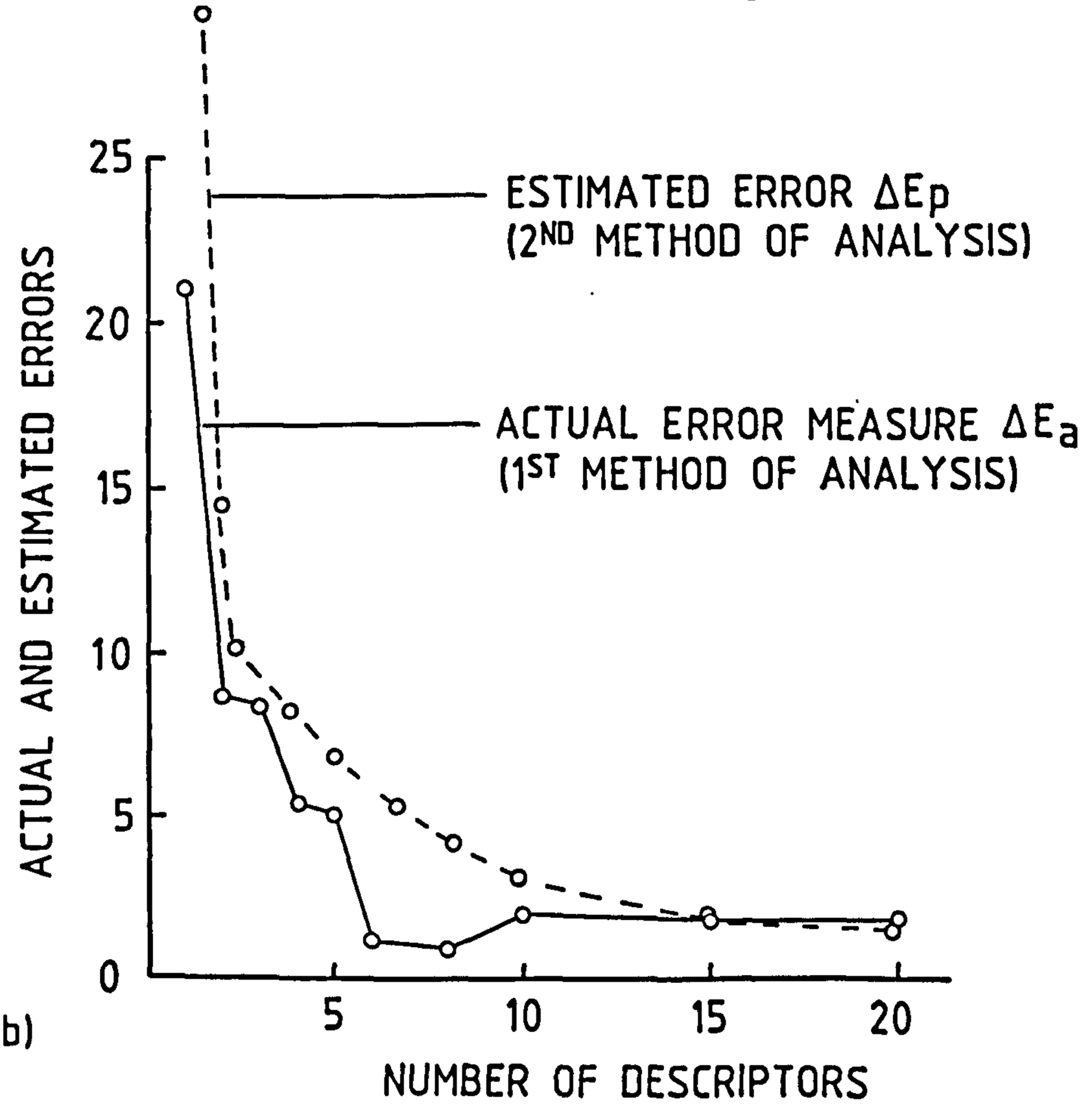
$$V_o^T(\dot{y}(t)) = \left(\sum_2^k |\dot{y}_i - \dot{y}_{i-1}| \right) + |\dot{y}_k - \dot{y}_1)|$$

respectively. The actual error ΔE_a, for the triangle, evaluated using Equation (5.17), is shown in Figure 5.5. The predicted error, ΔE_p estimated from Equation (5.18), is superimposed on the same plot and for completeness a similar error estimation based on the first method has been made for the results of section 5.2, and this is shown in Figure 5.5 (b).

An error curve has also been generated for the wrench, Figure 5.6, and both figures indicate that the primary shape can be recognised by 10 descriptors with a high degree of confidence. It is apparent that as the contour becomes more complex, the predicted boundary becomes more conservative and Kuhl and Giardina [6] have suggested incorporating a heuristic weighting, for example, a function of the number of large angular changes in the contour.



(a) Actual and estimated error curves for the reconstruction of the triangle.



(b) Actual and estimated error for the reconstruction of the triangle.

Figure 5.5 Actual and estimated error curves.

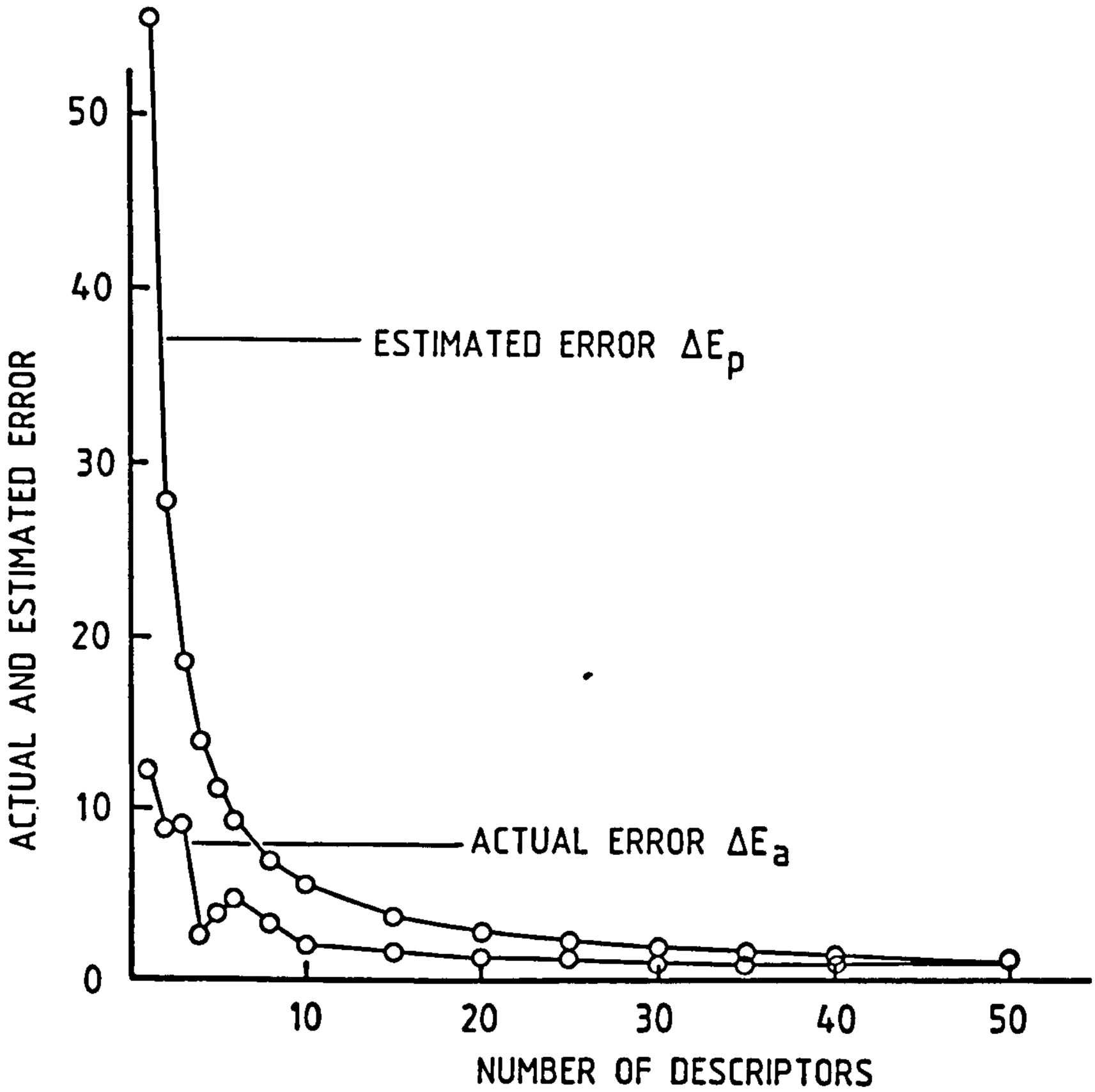


Figure 5.6 Actual and estimated error curves for the reconstruction of wrench.

5.7 ENGINEERING APPLICATION

The results obtained for the simple shapes have been extended to include two engineering components, photographs of which are given in Figure 5.7 for reference. The expected error curves, Figure 5.8, again indicate that 10 to 15 descriptorswill identify the principle shape of the component and that the error measure Table 5.4, evaluated using 12 descriptors, is an effective standard of comparison for object recognition. In this table the error scale demonstrates the potential of the method as a means for detecting faulty components typically missing blades or part blades. In general, the higher error values are associated with the greatest defects.

The CPU time (VAX 8600) for tracing the fan boundary was 40 ms (4 blades) and 60 ms (7 blades) for the two engineering component shown,Figure 5.7.

The CPU time using a simple Fast Fourier Transform algorithm [7] for the evaluation of 50 descriptors using Equations (5.15) and (5.16) was 15 ms for the 7 bladed fan and 10 ms for the 4 bladed fan.

The CPU times stated are within the normal manufacturing and assembly time scales for most industrial applications. The figures, Figures 5.9 and 5.10, show the change in the reconstructed shape of the silhouette contour, for

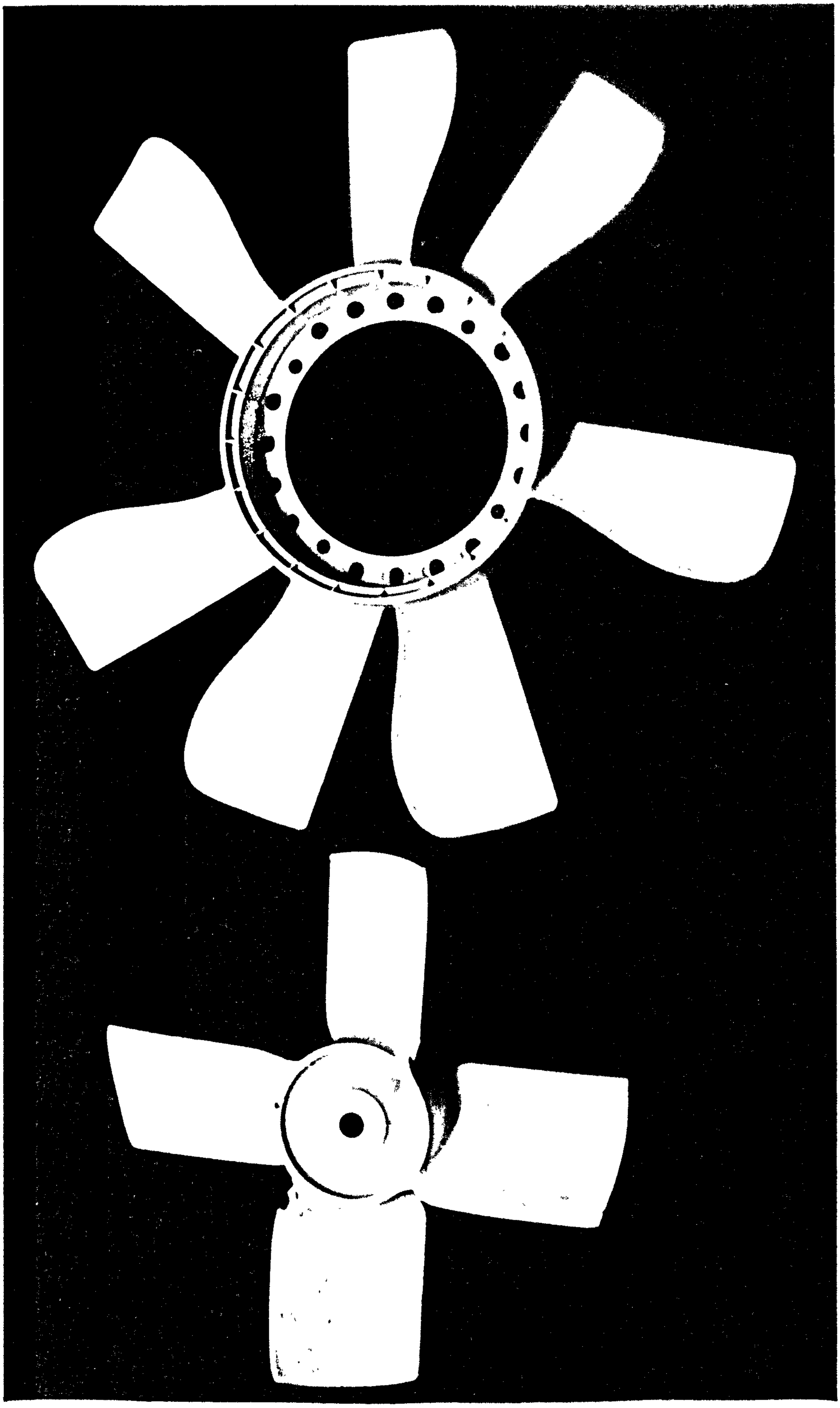
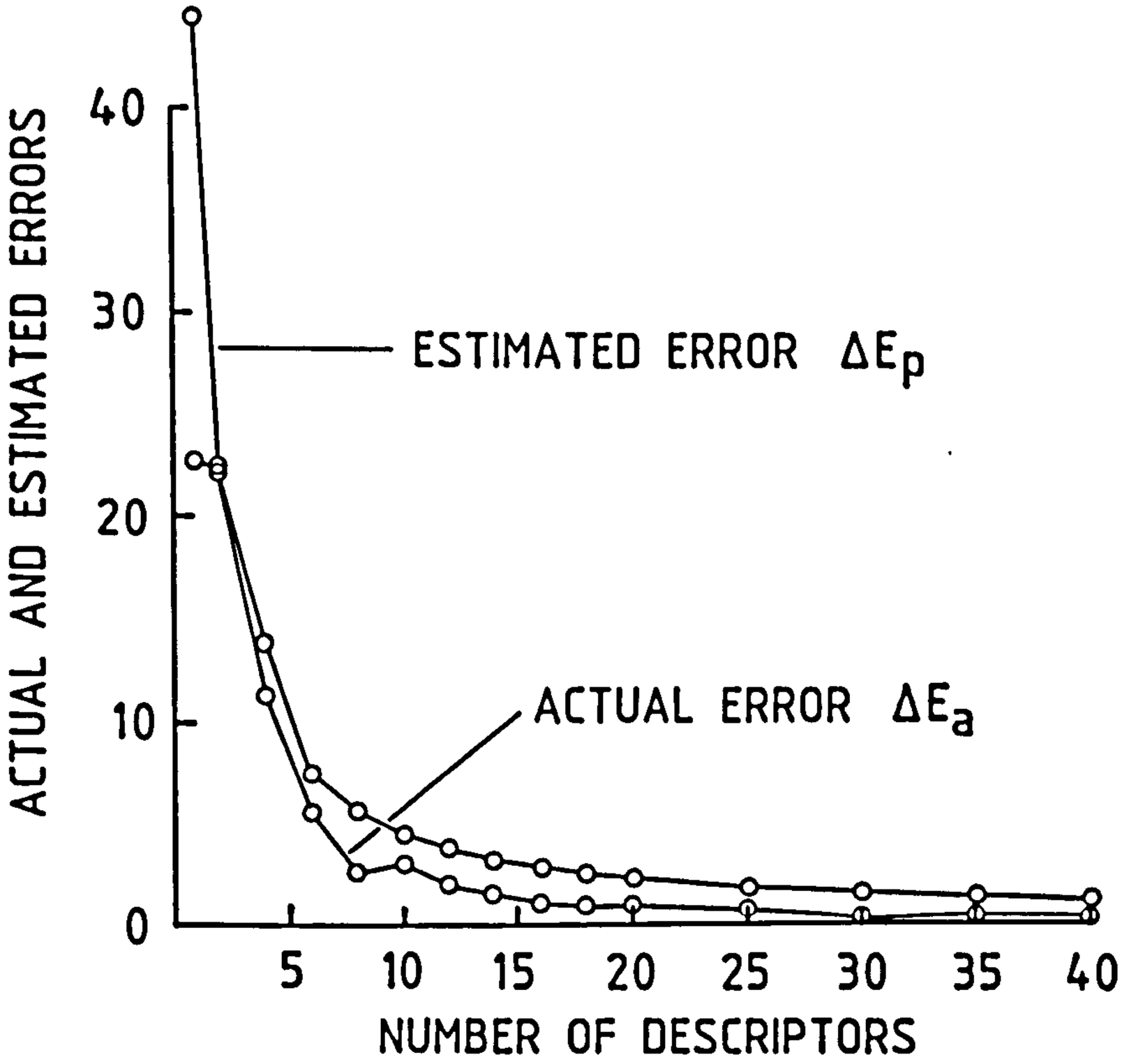
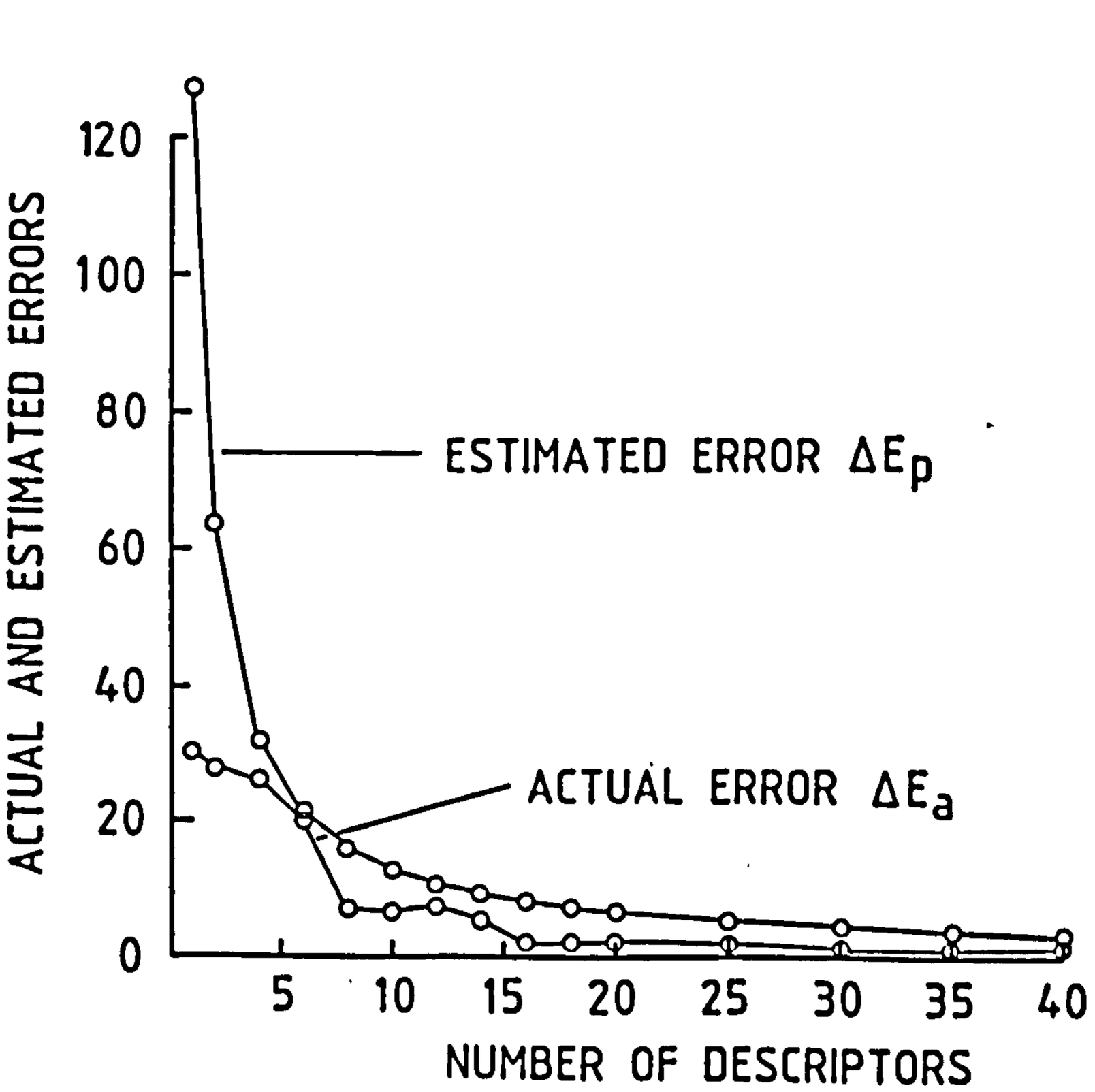


Figure 5.7 Simple engineering components.



(a) Actual and estimated error curves for the reconstruction of four bladed fan.



(b) Actual and estimated error curves for the reconstruction of seven bladed fan.

Figure 5.8 Actual and estimated error curves.

FIRST EXPERIMENT		SECOND EXPERIMENT	
Nature of Comparison	Error Measure	Nature of Comparison	Error Measure
Fan with 7 blades compared to its rotated version.	0.7	Fan with 4 blades compared to its scaled version.	0.2
Fan with 7 blades compared to its rotated version with a different size.	0.8	Fan with 4 blades compared to its rotated version with a different size.	0.3
Fan with 7 blades compared to its scaled version i.e. much smaller version.	0.8	Fan with 4 blades compared to its rotated version.	0.3
2/3 of two of the 7 bladed fan masked and compared with a version where 2 of the blades were completely masked out.	1.0	1/3 of two of the 4 blades masked out and compared to the original 4 bladed fan.	0.6
1/3 of two of the blades of fan with 7 blades masked and compared to its original version.	1.1	2/3 of 2 of the 4 bladed fan masked and compared to a version where 1/3 of 2 of the blades were masked.	0.6
1/3 of two of the 7 blades masked and compared to a version where 2/3 of 2 of the 7 blades were masked.	1.1	2/3 of 2 of the 4 blades masked and compared to a version where 2 of the 4 blades completely masked.	0.7
1/3 of 2 of the 7 blades masked compared to a version where 2 blades were completely masked out.	1.9	2/3 of 2 of the 4 blades masked and compared to the original 4 bladed fan.	1.1
Original 7 bladed fan compared to a version where 2/3 of 2 of the blades were masked.	2.0	1/3 of 2 of the 4 blades masked and compared to a version where 2 blades were completely masked.	1.2
Original fan with 7 blades compared to a version where 2 blades were completely masked out.	2.6	Original 4 bladed fan compared to a version where 2 of its blades completely masked out.	1.6
Fan with 4 blades compared to 7 bladed fan when 2 of the 7 blades were masked out.	3.7	Original 4 bladed fan compared to a version of 7 bladed fan where 4 of its blades were 1/3 of the way masked.	3.6
7 bladed fan compared to the 4 bladed fan.	3.4	4 bladed fan compared to the 7 bladed fan.	3.4

Table 5.4 Similarity measured for the two engineering objects, i.e. the seven bladed and four bladed fans, for different sizes and positions with parts being masked (invisable to camera). Error measure as calculated by Equation (5.8)

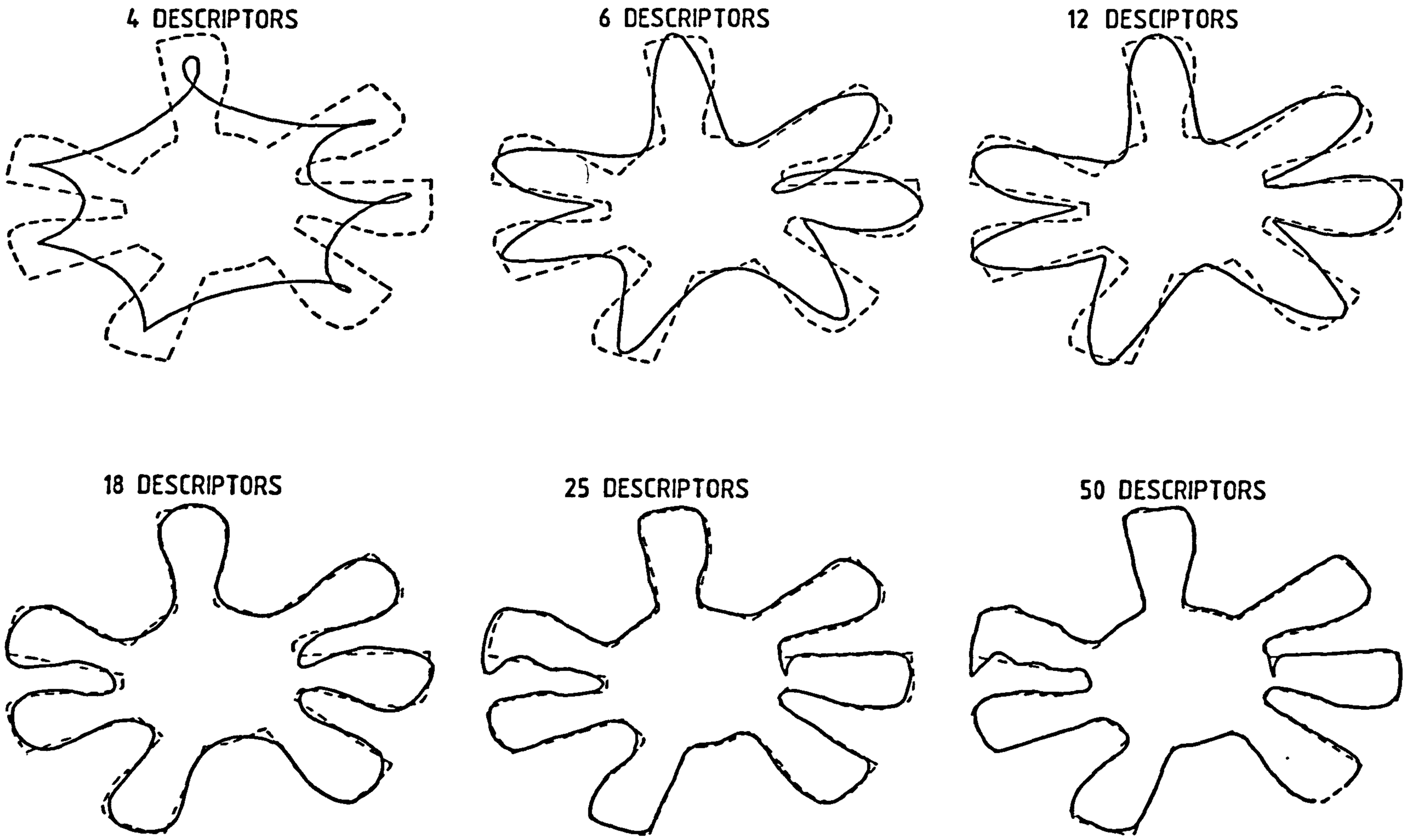


Figure 5.9 Reconstruction of seven bladed fan using first method.

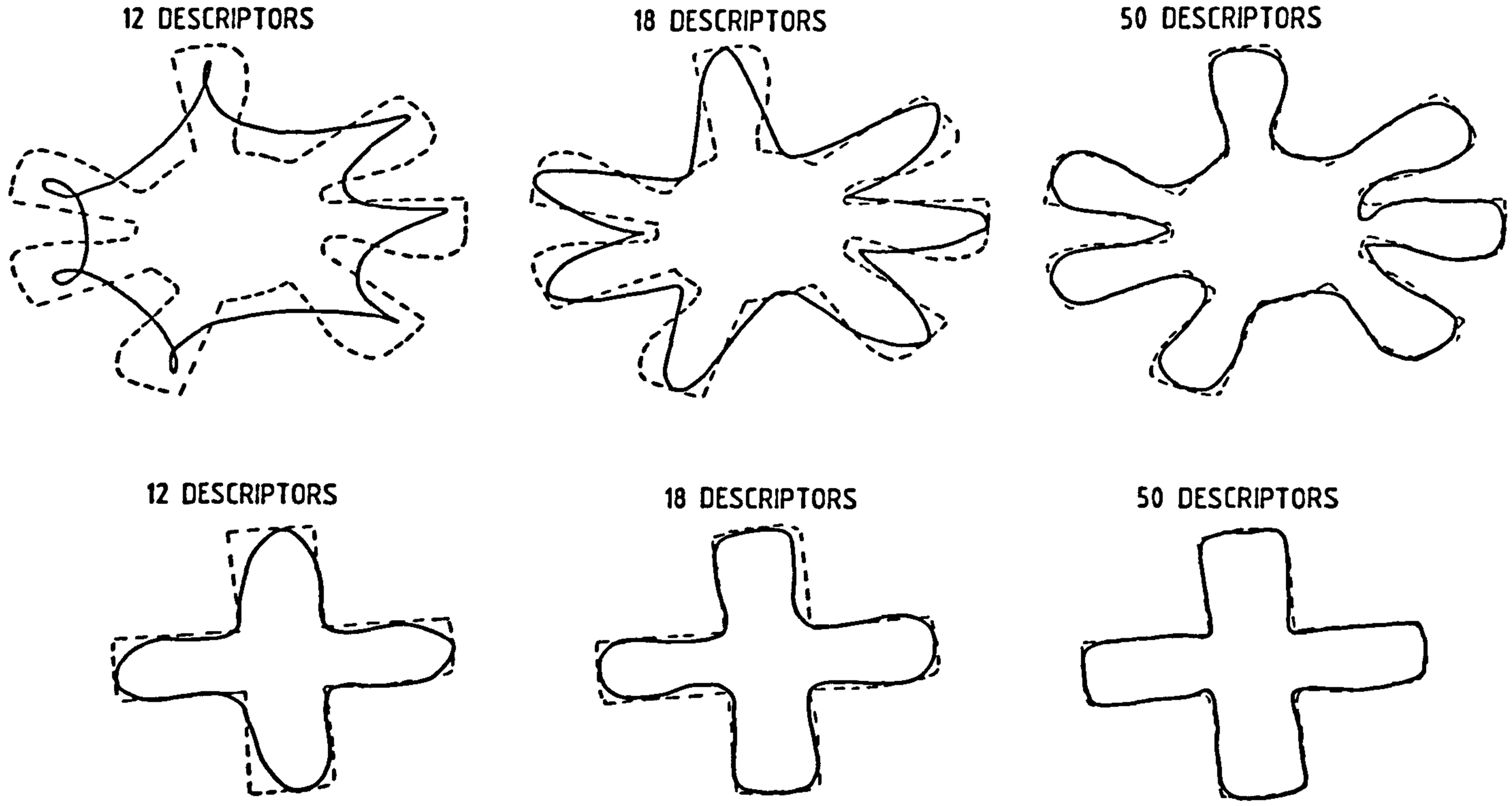


Figure 5.10 Reconstruction of four and seven bladed fans using second method.

each fan, as the number of descriptors is increased and these give a further guide to the number of descriptors necessary to achieve a given precision in the object recognition operation. For reference a typical histogram, Figure 5.11, shows the contrast between background and object for the image processing carried out in the investigation.

5.8 DISCUSSION AND CONCLUSIONS

The results clearly demonstrate the value of the normalised Fourier Descriptor as a method for classifying objects by virtue of their shape. Extensive contour data can be condensed to a relatively small number of numerical values which are independent of the object size and location relative to a fixed set of reference axes.

Both simple 2D shapes and more complex engineering components have been studied and in the latter case the silhouette has been shown to be adequate for recognition and identification purposes.

In the derivation of the Fourier Descriptors two alternative approaches have been taken and comparisons have been made between the two methods based on the reconstructed shape. The first method based on the coordinates of the object boundary gives a rapid test approximation to the object shape but does

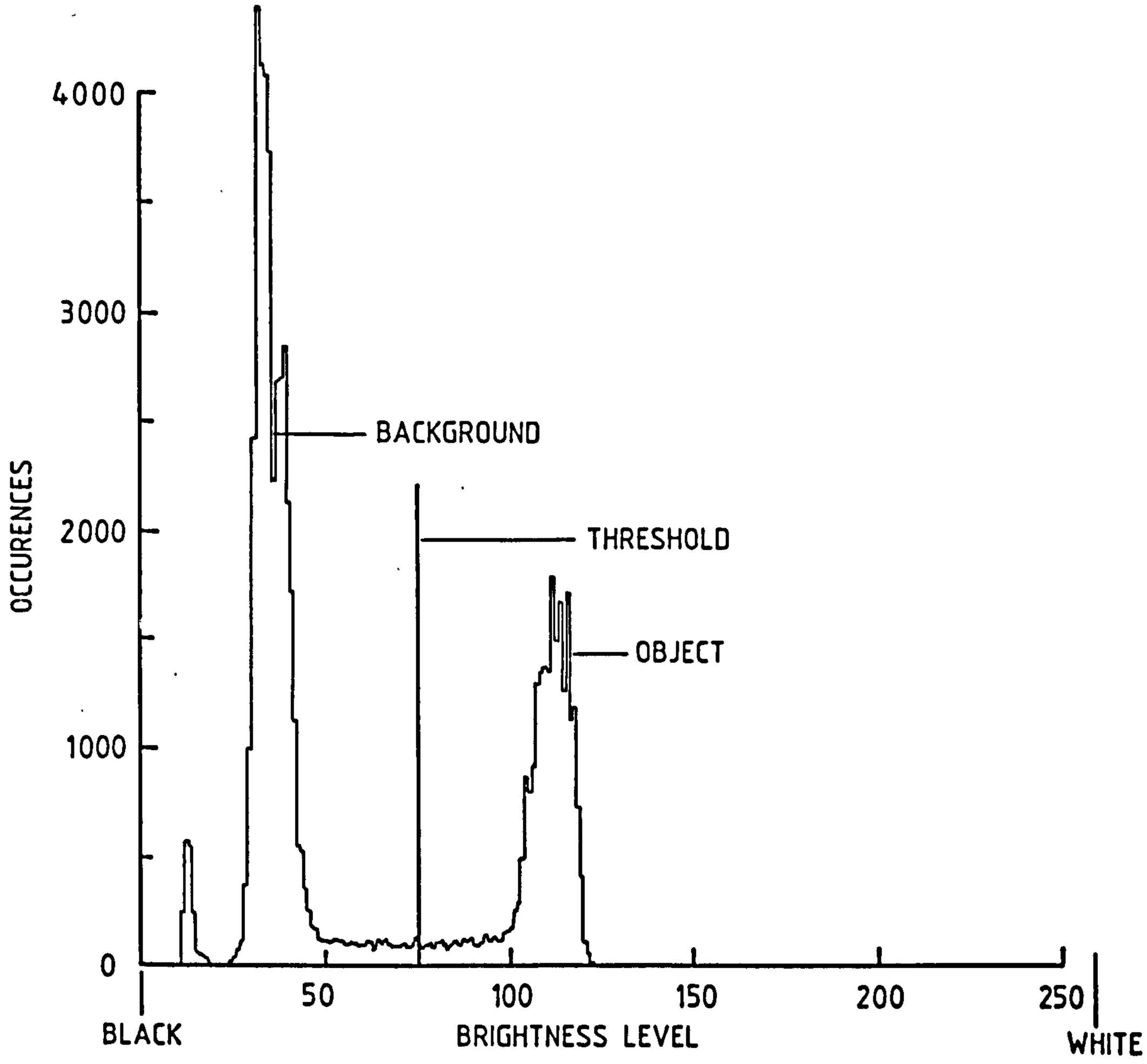


Figure 5.11 Typical histogram for threshold determination.

not possess the convergence and higher accuracy of reconstruction exhibited by the second method when a larger number of Fourier descriptors are employed. The second method based on the Freeman chain code can, therefore, be recommended particularly for the classification and recognition of more complex shapes.

In order to judge the degree of similarity between objects the concept of a reference feature vector has been established. The lack of similarity is measured by means of an error function based on the differences in the normalised Fourier descriptor vectors for the reference and test object. This has been successfully employed in the identification of a number of different geometrical shapes.

To predict in advance the sub-optimum number of Fourier descriptors necessary for reliable recognition and identification, a theoretical error function based on the truncated Fourier series for the boundary data has been established and tested for a range of objects. The error magnitude has been shown to decrease rapidly with increasing number of Fourier descriptors before finally levelling out. This indicates the possibility of establishing an optimum gradient for automatically assessing the required size of the Fourier descriptor vector.

C H A P T E R S I X

OBJECT RECOGNITION FROM TEXTURAL FEATURES ANALYSIS

Methods and applications of shape and textural analysis are examined with particular emphasis on application to object recognition.

6.1 INTRODUCTION

Although the approach of Chapter Five will allow the recognition of simple engineering shapes based on silhouettes, it is unable to detect the relatively major textural details, e.g. the texural details of fan blades, which can vary from one side of the object to the other, and hence is unable to detect which side is being viewed by the camera. To overcome this problem, this chapter examines the problems associated with simple shape representation and image pattern recognition.

The technique adopted by this author is based on the analysis of the objects surface texture (perceived structure) which permits the adoption of some of the ideas of the previous chapters. An attempt is made to use the developed 2D methods on simple objects and the problems that are most likely to be encountered by adopting this approach in an engineering environment are highlighted.

6.2 SURFACE TEXTURE ANALYSIS: REVIEW

In published literature, few researchers have used texture analysis for object surface definition and surface analysis using camera based methods [1, 2, 3]. In their work, range data has been employed as an additional cue for segmentation and detection of surface orientation. Laser range finders are often used to obtain the required data, and

specialised hardware designed to analyse them [1]. Object recognition using texture analysis, under study here, has attracted the attention of a very few researchers and the techniques and numerical analysis employed by this author to overcome various problems at different stages are therefore believed to be original.

6.2.1 Statement of the Problem

A strategy for recognition using the developed texture analysis methods is required. The basic notion of recognition strategy is the feature vector. The feature vector V is a set of measurements $\{v_1, v_2 \ldots\ldots, v_m\}$ which condenses the decription of relevant properties of the image into a Euclidean feature space of m dimensions.

Each point in feature space represents a value for the feature vector applied to a different image (or subimage). The measurement values for a feature should be 'compared' with its class membership.

Different means of 'comparison' are developed and results are qualitatively studied.

6.3 TEXTURE ANALYSIS TECHNIQUES AND APPLICATIONS

To develop feature vectors which contain the surface attributes to be used in object recognition, texture analysis is conducted in either the spatial or frequency domain, the choice is largely dictated by the application objectives.

6.3.1 Spatial Domain

The spatial domain oriented techniques normally employ one of the following methods:

i) Statistics of the grey level distribution of pixels (i.e. 2D histograms) [4].

ii) Detecting and characterising local grey level extrema (minima and maxima) by scanning an image in the horizontal and vertical directions [5].

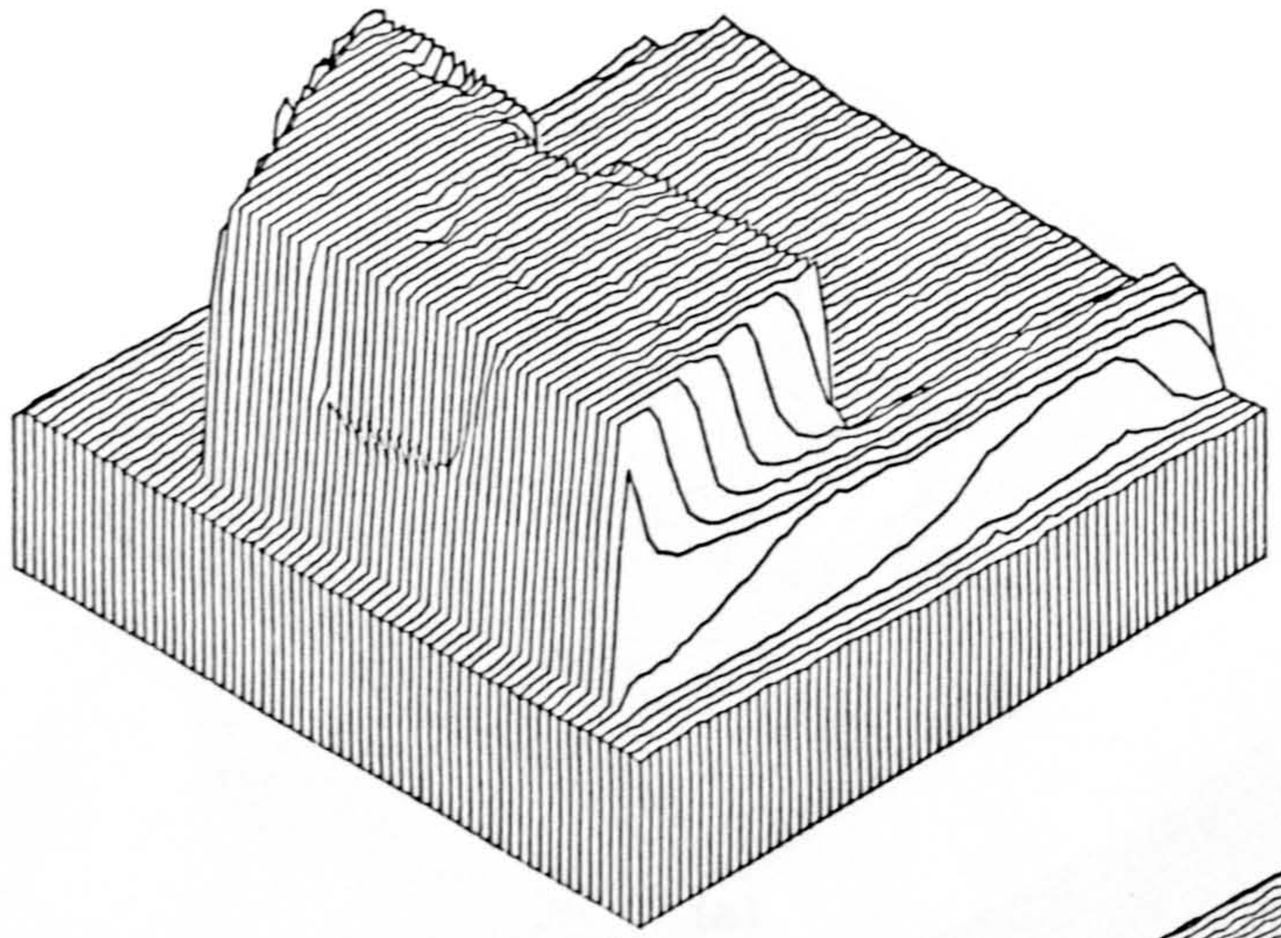
iii) Detecting edge elements using a gradient operator [6].

These methods all involve two-dimensional tasks and this author has used the methods presented in the earlier chapters, e.g. edge detection, histogram routines and Fourier analysis, to determine if usable feature vectors can be identified.

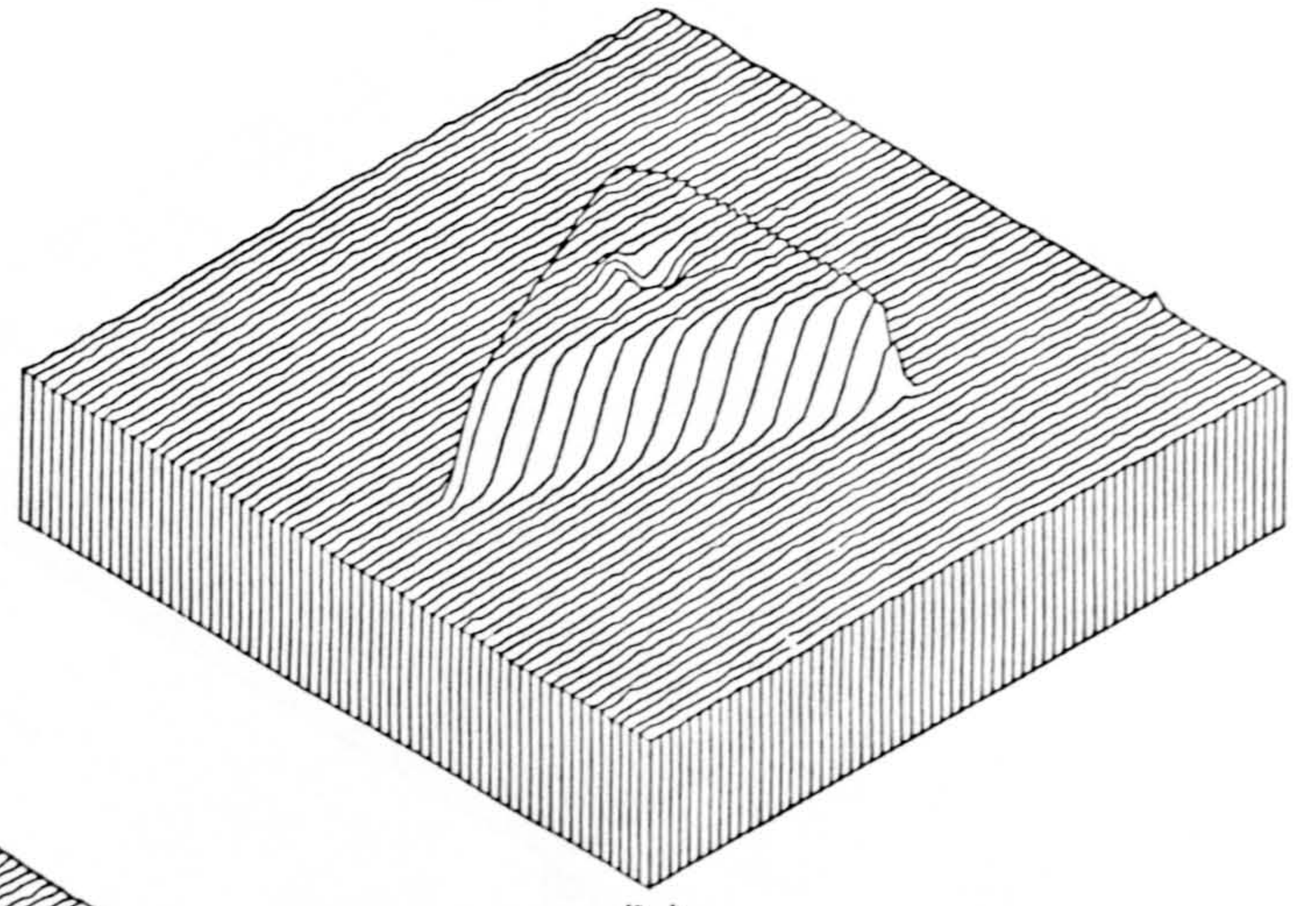
Data contained in the normal object pixel voltage array does contain surface details and the displays of Figures 6.1 and 6.2 are typical of these arrays. A weakness to be recognised, however, is illustrated for a gear-wheel in Figure 6.2(c). The reflections from the wheel centre are more significant than those from the dull toothed outer edge. Therefore, some object pre-treatment may be necessary in a production environment. The 3D intensity histograms developed are shown in Figure 6.3 for three engineering components. These were created by scanning each row of the image and obtaining the grey level histogram for it, i.e. the total number of pixels for each of the grey levels present is recorded and plotted. No obvious way of extracting a feature vector from any of these histograms is apparent and the only obvious use to which these might be put is to study contrast between object and background, indicating changes in reflective properties.

6.3.2 Frequency Domain

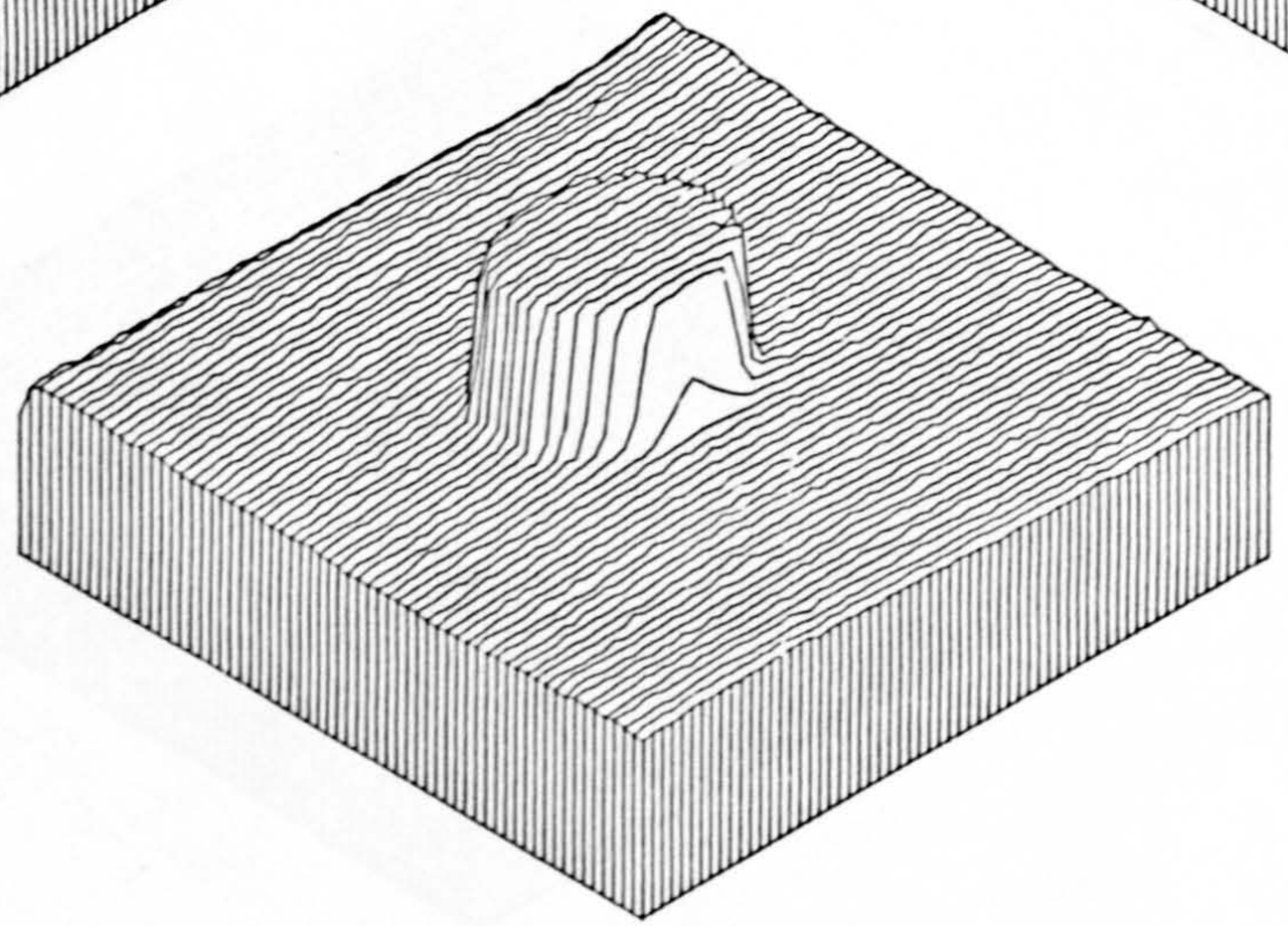
Texture analysis techniques in the frequency domain use the periodicity properties of the texture in a surface [1]. If a surface is at all spatially periodic or directional, its 'power spectrum' will tend to have peaks for corresponding spatial frequencies [7]. These peaks are available to form the basis of a pattern recognition discriminator if an interpretation algorithm can be found.



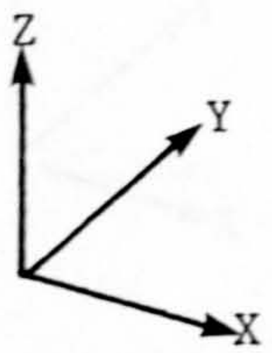
(a)



(b)

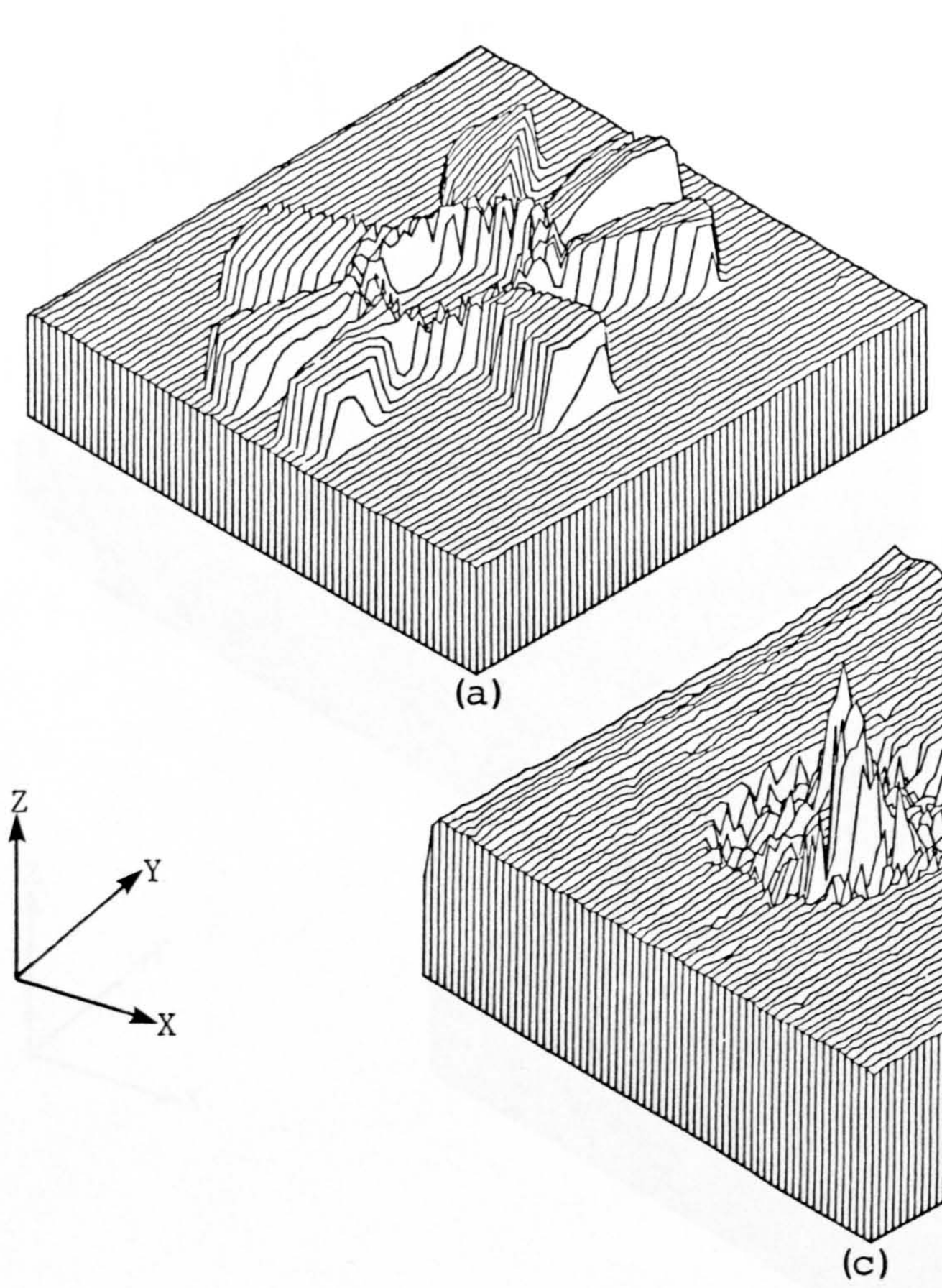


(c)



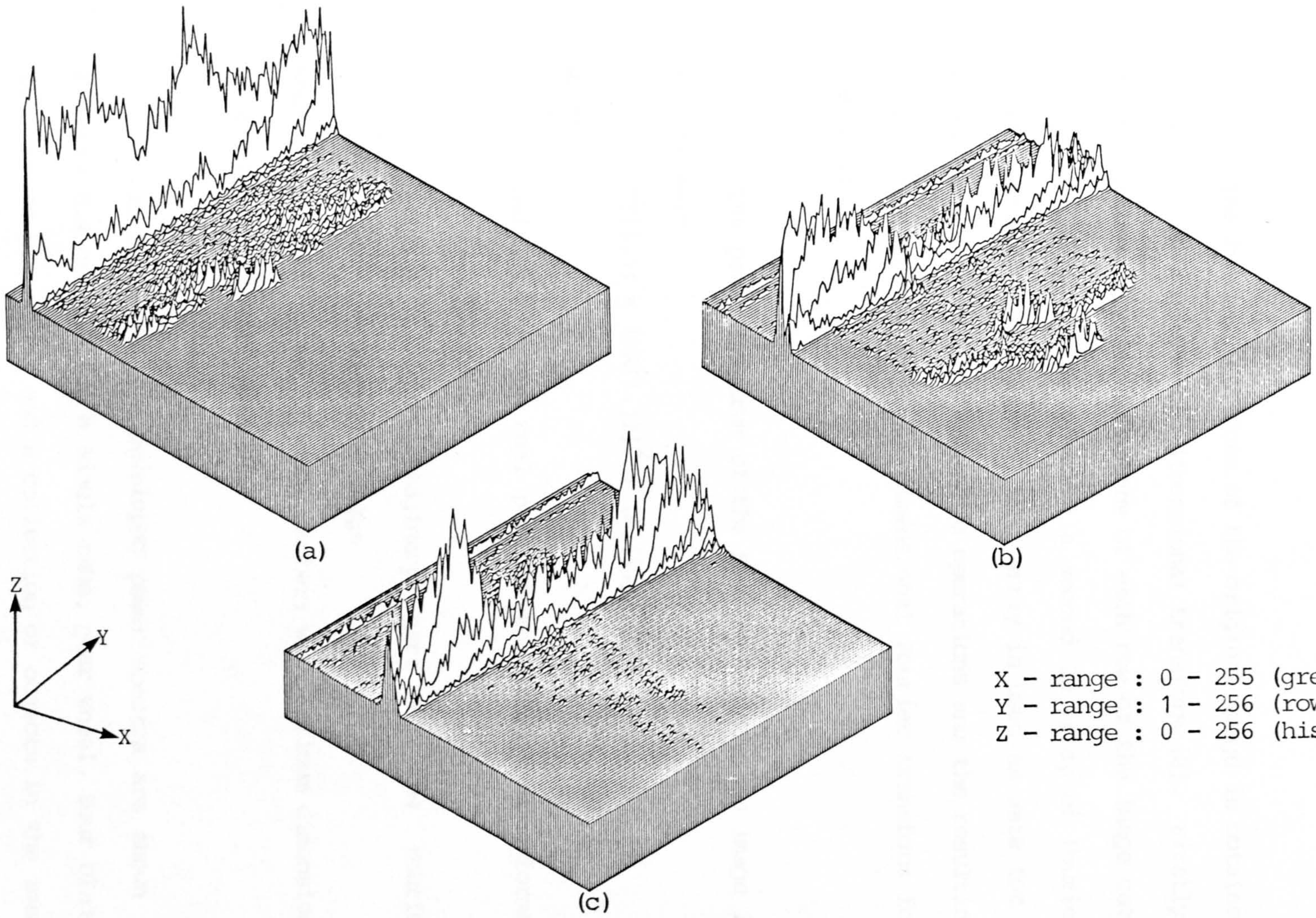
X - range : 1 - 256 (rows)
Y - range : 1 - 256 (columns)
Z - range : 0 - 255 (intensity levels)

Figure 6.1 The isometric view of the pixel voltage displays for:
a) the cube b) the triangle
c) the cylinder (plan view).



X - range : 1 - 256 (rows)
Y - range : 1 - 256 (columns)
Z - range : 0 - 255 (intensity levels)

Figure 6.2 The isometric view of the pixel voltage displays for:
a) the seven bladed fan b) the four bladed fan
c) the gear wheel.



X - range : 0 - 255 (grey levels)
Y - range : 1 - 256 (rows)
Z - range : 0 - 256 (histogram)

Figure 6.3 The three-dimensional histograms for:
a) the seven bladed fan b) the four bladed fan
c) the gear wheel.

The power spectrum, which might provide a basis for the development of a feature vector can be obtained as follows.

The Fourier transform of the original image is obtained by calculating a pseudo two-dimensional transform [8]. Firstly, the traditional Fourier transform of each row of the image data ($X_{i,J}$) is calculated to generate a second 2D array of Fourier coefficients. Each column of this array is used as data for a second series of Fourier transform operations and the resulting array is described as a two-dimensional Fourier transform for the image.

The power spectrum of the Fourier transformed image is defined as:-

$$PS(i,j) = [AR(i,J)]^2 + [AI(i,J)]^2$$

where:

AR(i,J) is the real part of the Fourier transformed image

AI(i,J) is the imaginary part of the Fourier transformed image

where i, J is the pixel 2D array number; with maximum dimensions of 256 by 256 in this study.

Examples of the developed power spectra are shown in Figures 6.4 and 6.5 for a single cube, gear wheel, four bladed fan, seven bladed fan and a collection of objects in the image

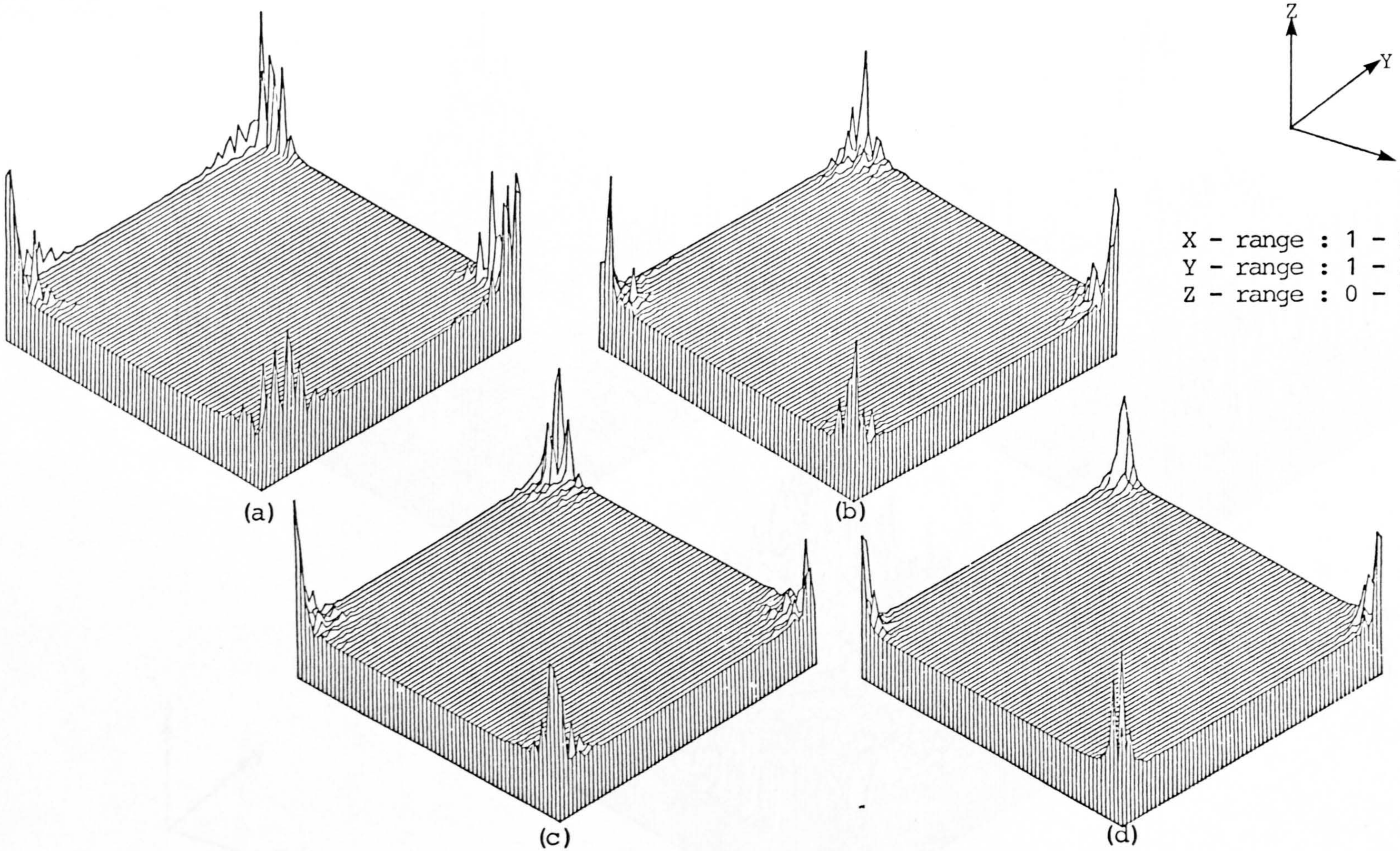


Figure 6.4 The isometric view of the power spectra for:
a) the cube b) the gear wheel
c) the seven bladed fan d) the four bladed fan.

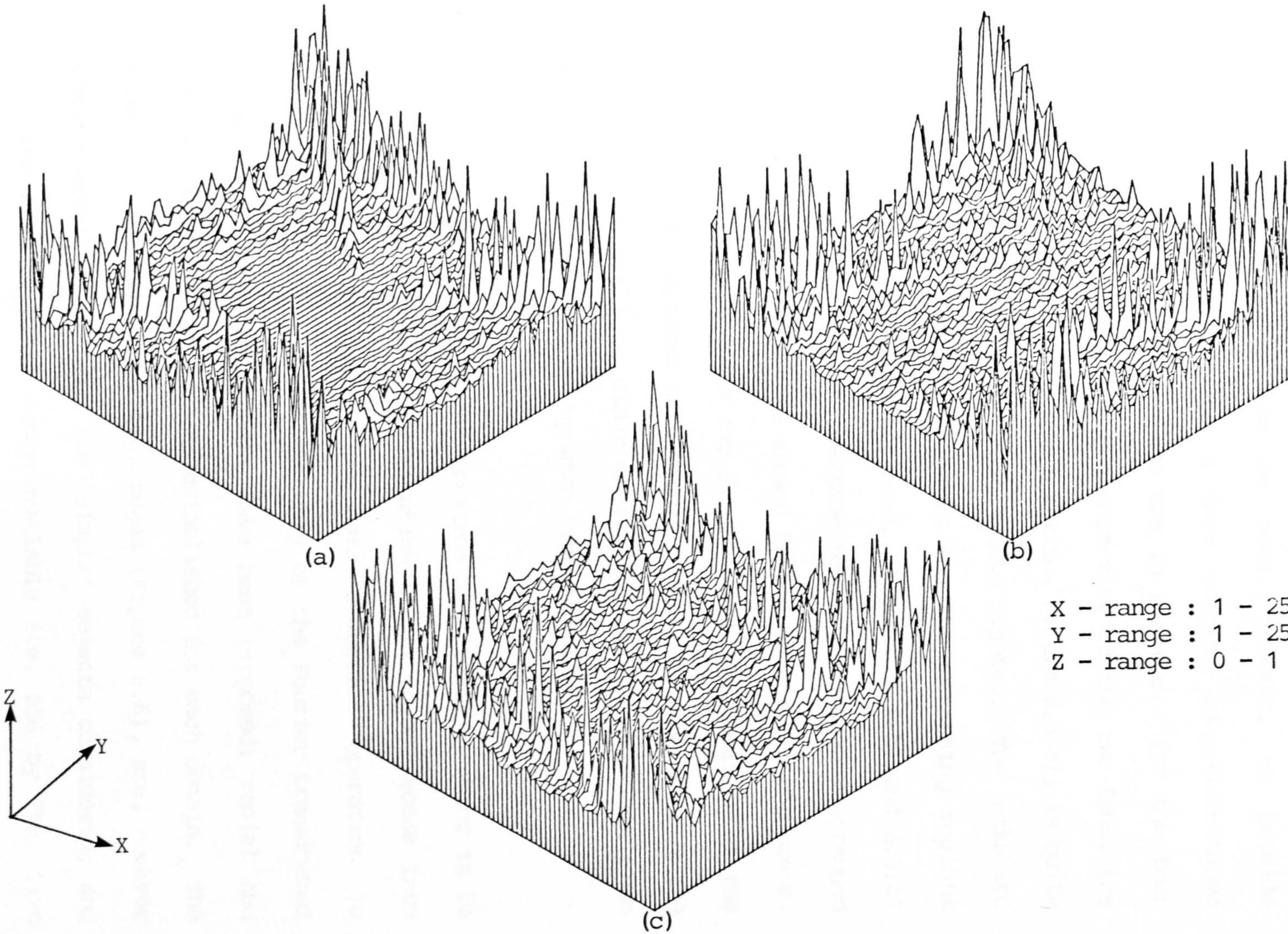
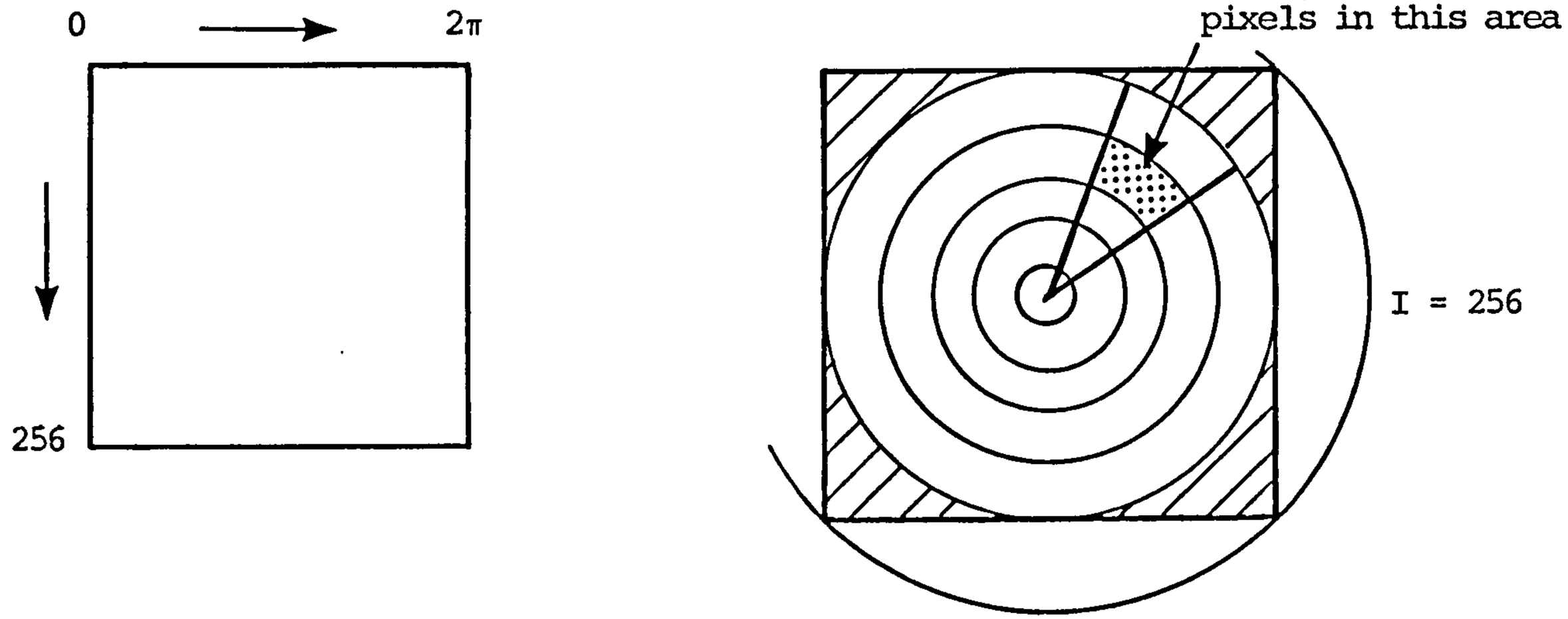


Figure 6.5 The isometric view of the power spectra for the four objects in the scene, a) the original, b) and c) rearranged.

scene. The spectral displays for the first four objects listed are very similar and only indicate the presence of an object. The spectral display can be used, however, to provide information that indicates a more complex composite-shaped object or a number of objects are in a scene. For the four objects in the scene, i.e. the plan view of the two fans, the cube and a cylinder, the power spectrum, Figure 6.5(a), is quite distinct and different from the previous Figures. The peaks are associated with the four objects and the remaining regions represent the featureless surfaces. Figures 6.5(b) and 6.5(c) show the power spectra of the same four shapes when rearranged and ordered differently (as viewed by the camera) in the scene. It is evident from these two Figures that the location of the peaks has been altered as a result of the rearrangement of the objects, but still no other useful information may easily be derived from this type of spectra representation.

One approach to the creation of a feature vector is to partition the Fourier representation of an object scene into domains or bins [9]. The data reduction operation is permissible because of the symmetry of the Fourier transformed image. Two types of domains have been proposed: radial and angular, with feature vectors established for each domain. The number of sub-divisions, i.e. zones (Figure 6.6), are, however chosen arbitrarily. For the 'simple' objects considered, and for the size of pixel array available i.e. 256 by 256, five



$$\text{ring width} = \frac{128}{n}$$

n = number of rings

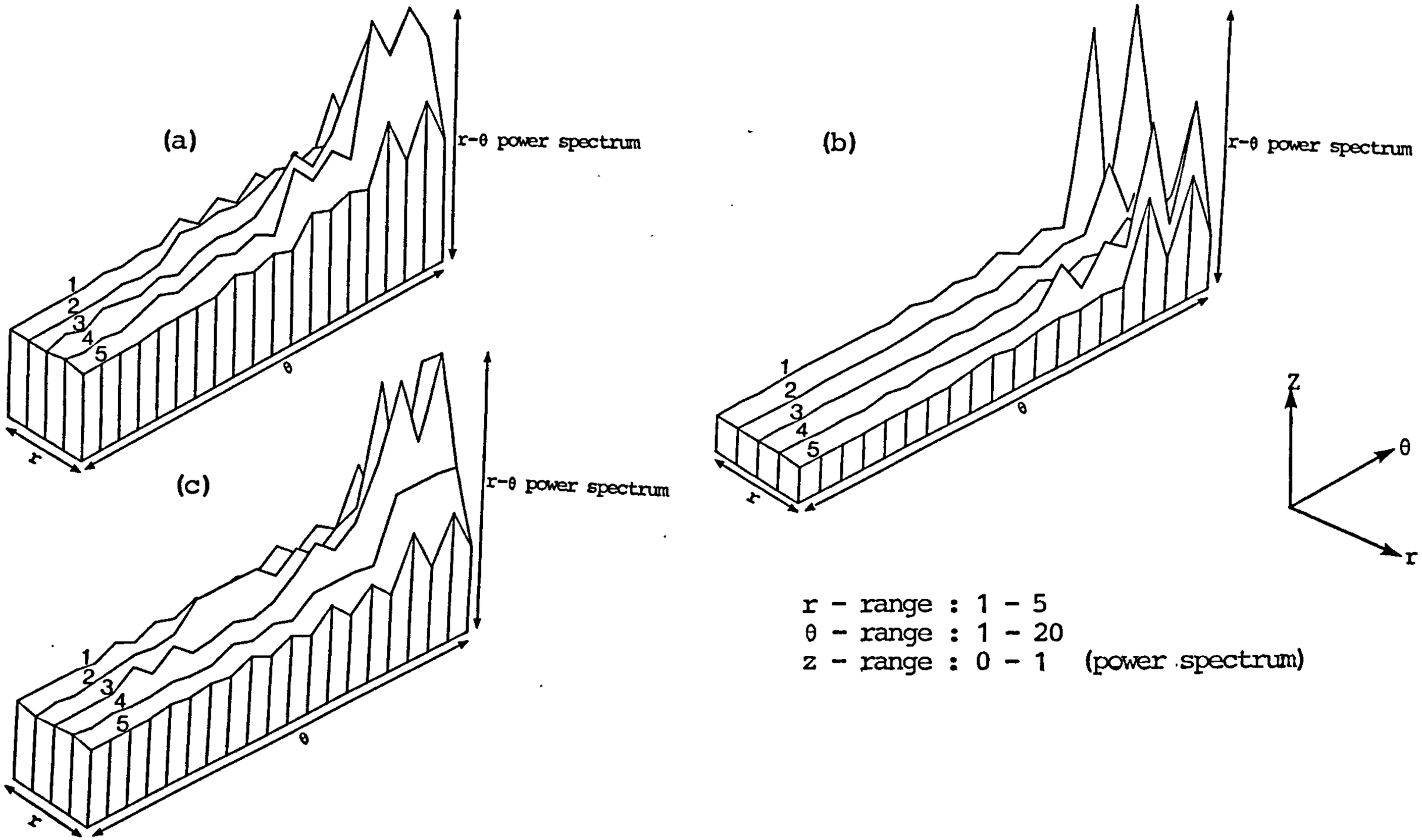
Figure 6.6 Partitioning of the Fourier space into domains.

radial and twenty angular divisions were chosen as appropriate. In a practical application, trials to arrive at the optimum number would be necessary.

Shown in Figures 6.7 and 6.8 are the power spectra of the plan views of the cylinder, the cube, the four and seven bladed fans corresponding to radius-angle zones in the Fourier transformed plane. Five feature vectors, one per radial strip with 20 elements each, can be extracted from the partitioned transforms displayed.

Figure 6.8 (b) shows the spectrum of a rotated version of the seven bladed fan for the same pixel array partitions, and supports the results of chapter 5.

For these simple engineering shapes, the feature vectors remain sensibly constant with object rotations. Although the feature vectors do change for the seven bladed fan if the reverse side is viewed, it is still classified by this method as a member of the 7 bladed fan family. For more complex shapes, additional trails will be necessary to confirm this property. Scaling, however, changes the power-spectra, and therefore the obtained feature vectors, see Figure 6.8 (c). Hence normalisation was considered and performed before the recognition process.



r - range : 1 - 5
θ - range : 1 - 20
z - range : 0 - 1 (power spectrum)

Figure 6.7 The isometric view of the angular - radial power spectra for :
a) the cylinder (plan view) b) the cube
c) the four bladed fan.

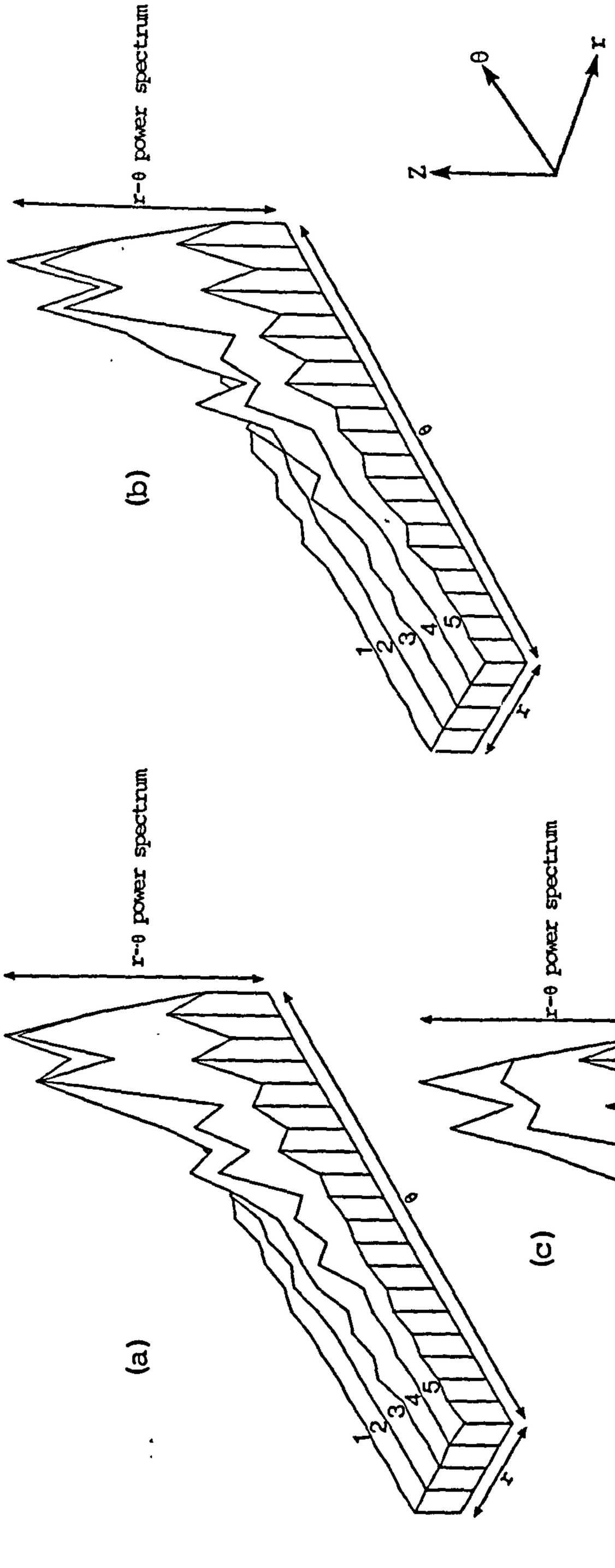


Figure 6.8 The isometric view of the angular - radial power spectra for:
a) the seven bladed fan b) the seven bladed fan (rotated)
c) the scaled version of the seven bladed fan, prior to normalisation.

The foundation of the recognition process is, therefore, based on the actual grey level propagation of the unprocessed image. That is, the resulting feature vectors of identical objects would not be similar as a result of lighting alterations or the presence of shadows.

To measure the variation in the lighting conditions and to get a better feel for the relative changes in the light intensity as the light sources were moved, thereby simulating a condition where controlled lighting would not be possible and where objects cast shadows on the background, a light-meter was used to obtain Table 6.1. Enhancement and segmentation of the object image were, therefore considered and performed prior to the recognition process to minimise the undesirable effects of illumination changes such as low object/background contrast and shadows.

Enhancement was based on the distribution of the grey levels obtained from the two-dimensional image histogram by increasing the separation between the two peaks appearing in this display.

Segmentation, to minimise the impact of shadows, was performed by a technique based on Rosenfeld's method [6], where the difference of average grey level between each pair of neighbourhoods were calculated. Thereby attempting to have a segmented image which would be invarient to changes in lighting condition. The size of a neighbourhood being dictated by the detail in the image.

Condition	Meter Reading (Scale)	Relative Light Intensity
Ambient	11.00	Datum
"Normal" lighting i.e. light source + Amb.	15.00	X 8 times datum
Light source taken 3 feet back	14.25	X 6.5 times datum
Light source taken 6 feet back	13.00	X 4 times datum
Light source taken 9 feet back	12.75	X 3.5 times datum
Light source taken 12 feet back	12.00	X 2 times datum
Incident light at 60^{o} to the normal condition, at reference distance	14.75	X 7.5 times datum

Table 6.1 Light Variation Measurements.

6.4 SEGMENTATION OF TEXTURE IMAGES

When an image includes a variety of textures, regions corresponding to a texture condition can be created by means of grey level averaging (segmentation).

To detect "texture edges", that is, where the average values of local picture properties change rapidly, pairs of adjacent neighbourhood grey levels are merged to create a single grey level region. The Rosenfeld's technique uses the difference of average grey levels between each pair of neighbourhoods to detect the existance of a texture edge. To calculate the average grey level in a neighbourhood, the image is partitioned into square regions and the average grey level estimated from:

$$A_K(X,Y) = \sum_{i=X-2^{K-1}}^{X+2^{K-1}-1} \sum_{j=Y-2^{K-1}}^{Y+2^{K-1}-1} a(i,j)/2^K * 2^K \qquad (6.1)$$

where:

a(x, y) denote the grey level at point (x, y)

K specifies the number of pixels in a side length

When locating the texture edges within partitioned image, of side length 2^K pixels in direction θ degrees, the edge value is obtained from:

$$E_{K,\theta}(x,y) = |A_K(x+2^{K-1}\cos\theta, y-2^{K-1}\sin\theta) - A_K(x-2^{K-1}\cos\theta, y+2^{K-1}\sin\theta)| \qquad (6.2)$$

A combination of horizontal and vertical search is required to generate the boundary pattern for the whole image. A value of K = 4 is shown to be appropriate for the type of objects considered, and the results of applying Equations (6.1) and (6.2) are shown in Figures 6.9. In this Figure, images of uniform grey level (within the square neighbourhoods) are created by forming groups of constant hue and the texture edges detected are illustrated.

6.5 RECOGNITION PROCESS

Using the developed radius-angle power spectra analysis on the images of the objects, with r = 5 and θ = 20, i.e. 5 radii and 20 angular divisions, five feature vectors with twenty elements in each vector, are obtained.

That is, $V_i = \{v_{i1}, v_{i2}, v_{i3}, \ldots\ldots, v_{i20}\}$ i = 1 to 5

Tests are conducted to determine what light variations can be tolerated and still retain the recognition properties of the five feature vectors proposed for the simple shapes adopted.

By normalising these five vectors in the manner as described in Chapter 5, they are made independent of object size. Two methods of comparison are reviewed for object

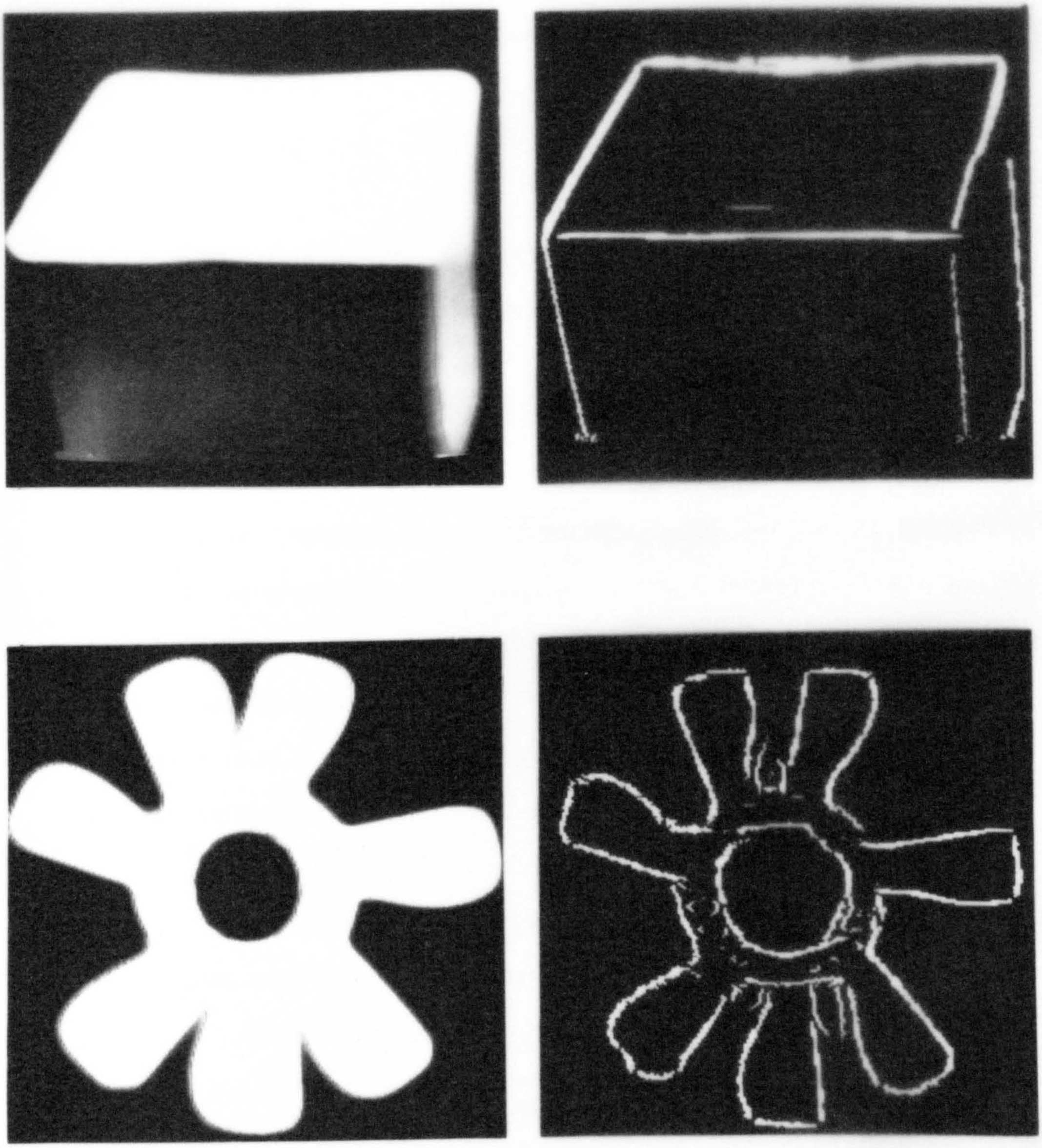


Figure 6.9 Image segmentation and detection of texture edges.

recognition. The first of these employs the method of Chapter 5 and requires the minimisation of:

$$D_1 = \frac{1}{20} \left[\sum_{i=1}^{20} |V_{1r}-V_{1t}|^2 \right]^{\frac{1}{2}}$$

where V_{1r} is the first feature vector of the reference object

V_{1t} is the first feature vector of the test object

and D_1 is the measured difference of the first feature vectors

D_2 to D_5 are obtained in the same way. The largest of these five values of D (D_{max}) is used as the error measure datum. To test that the technique is independent of object orientation and size, appropriate measurements were taken for the same object. An object with a significantly greater value of D_{max} is rejected and not recognised.

The second approach is based on the evaluation of a correlation coefficient [10], defined as:

$$\rho_1 = \frac{\left| \sum_{i=1}^{20} (V_{1r}-mV_{1r})\ (V_{1t}-mV_{1t}) \right|}{\left[\sum_{i=1}^{20} (V_{1r}-mV_{1r})^2 \sum_{i=1}^{20} (V_{1t}-mV_{1t})^2 \right]^{\frac{1}{2}}}$$

where: mV_{1r} is the arithmetic mean of the first reference vector

mV_{1t} is the arithmetic mean of the first test vector.

As the value of ρ approaches unity, the more correlated are the reference and test objects.

The criterion established and used for recognition, was to determine if all five correlation coefficients were greater than 0.6 and the arithmetic mean of the five was greater than 0.7, in which case recognition could be made with confidence. Typical results are shown in Figures 6.10 to 6.13, for the recognition of the seven bladed fan. By examining these results it becomes apparent that object recognition process improves considerably as a result of normalising the feature vectors. Also in these experiments, the lighting conditions were changed from time to time, and the results provided, confirmed that segmentation was a good technique for eliminating the effects of lighting variations.

6.6 DISCUSSION AND CONCLUSIONS

Texture analysis methods were developed in this chapter and it was shown that some information can be extracted from the 3D histograms and pixel voltage displays. These types of display do not, however, provide information that can be used in general image recognition. Techniques based on the power spectra of the object images and the partitioned Fourier transformed space were developed and the resulting feature vectors were shown to provide some basis for the recognition of objects. Segmentation is shown to be effective in minimising possible confusion caused by the presence of shadows and lighting variations.

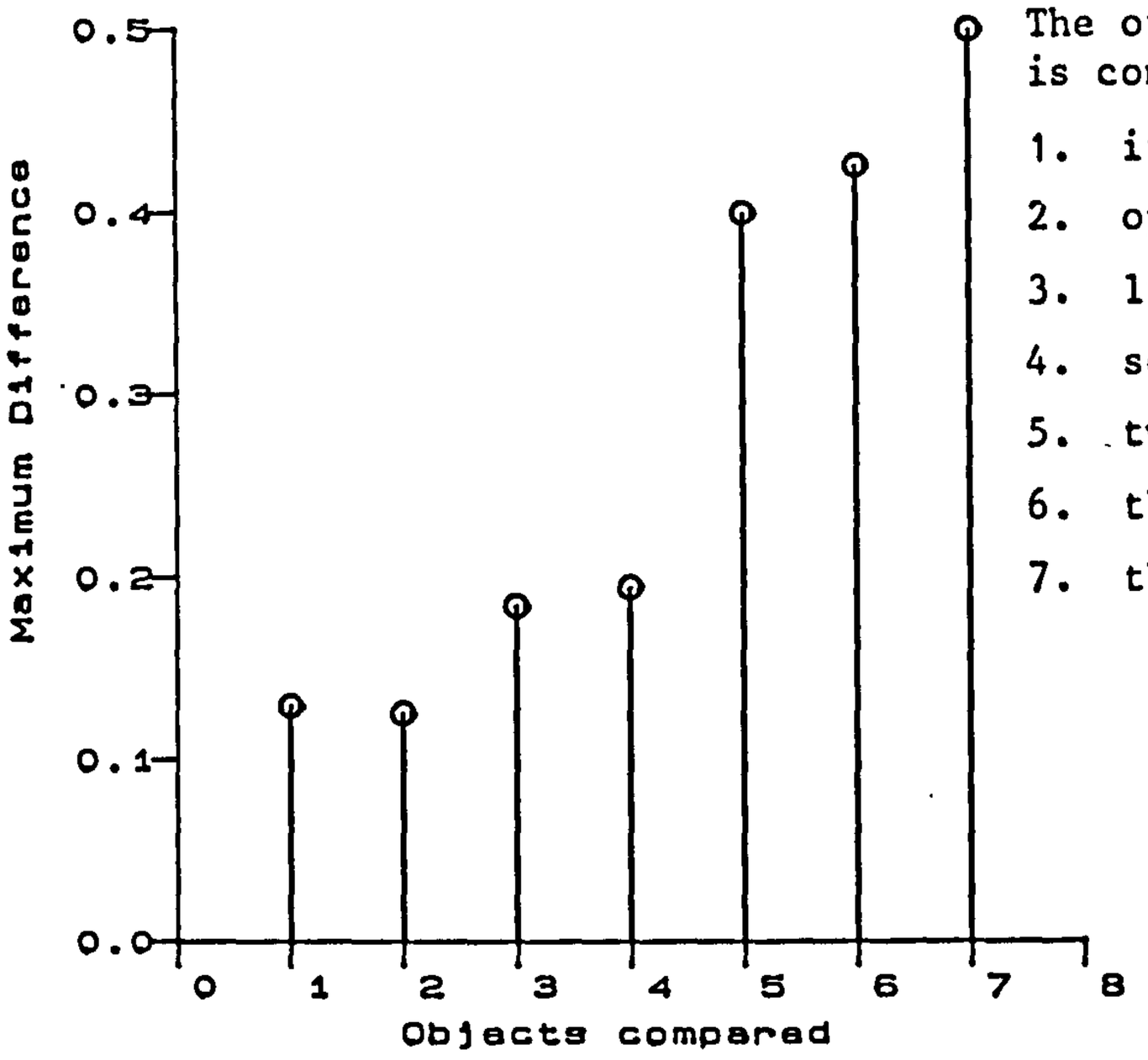


Figure 6.10 Recognition of the seven bladed fan using radius-angle power spectra and the measurement of D_{max}.

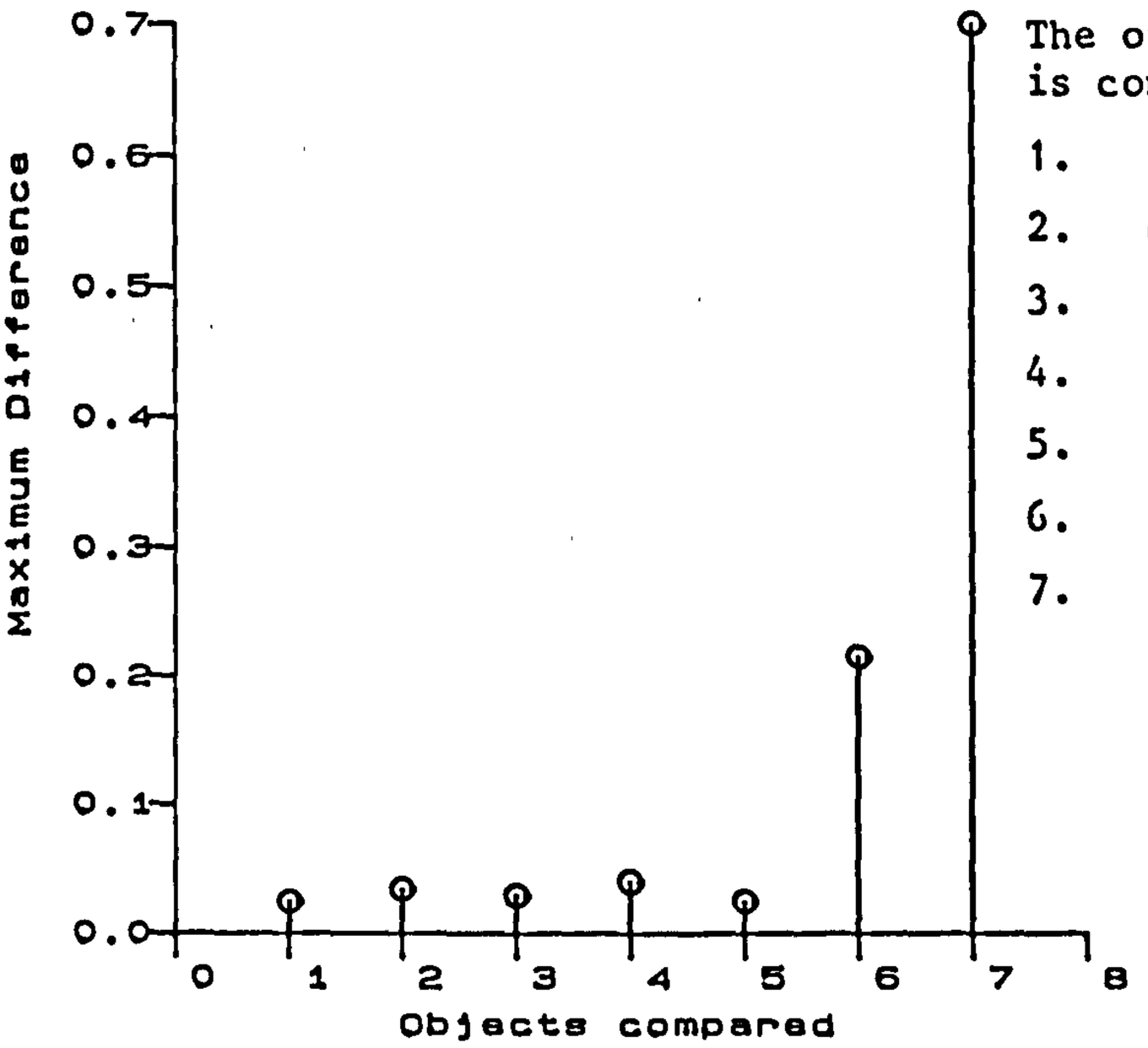
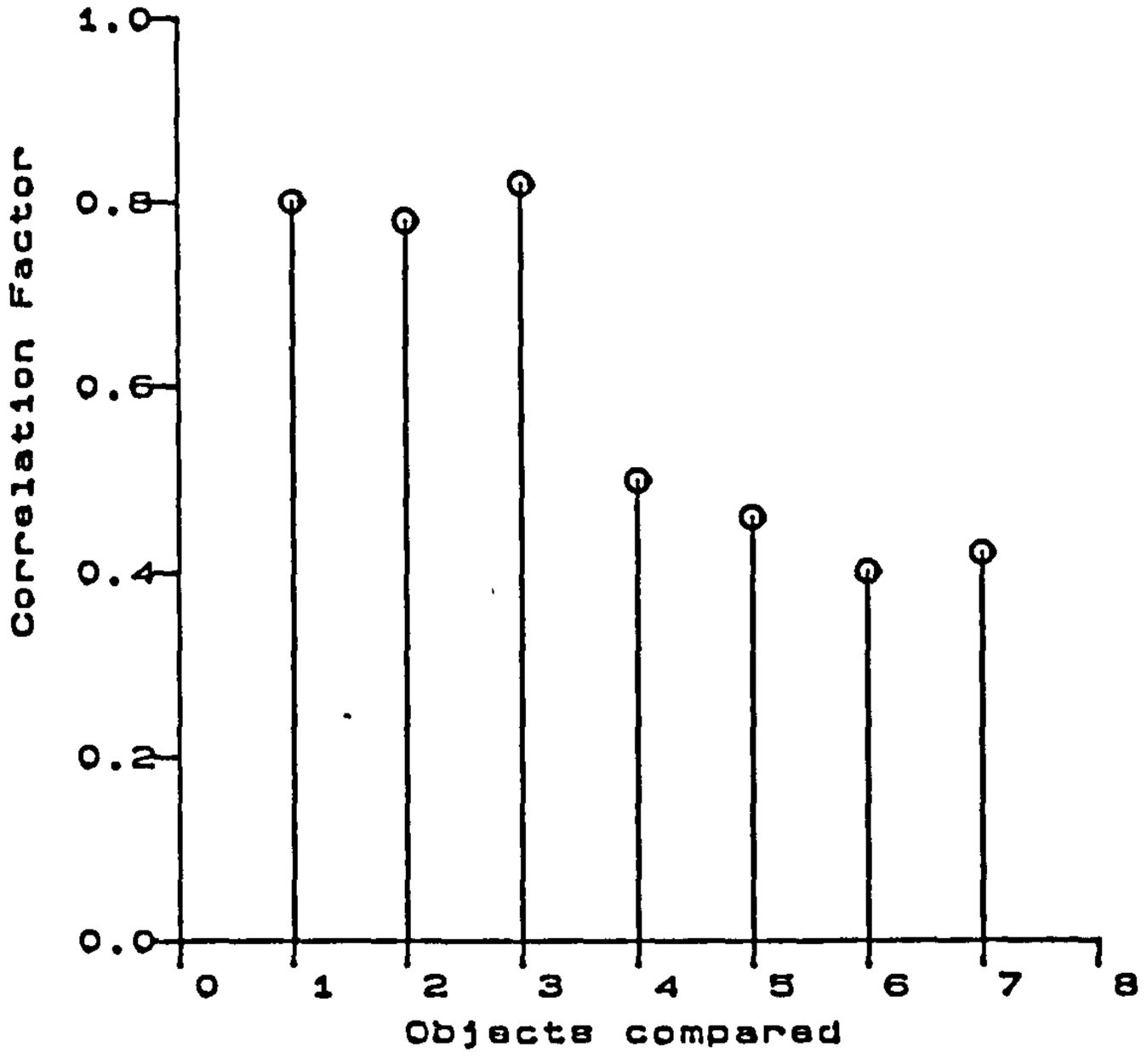


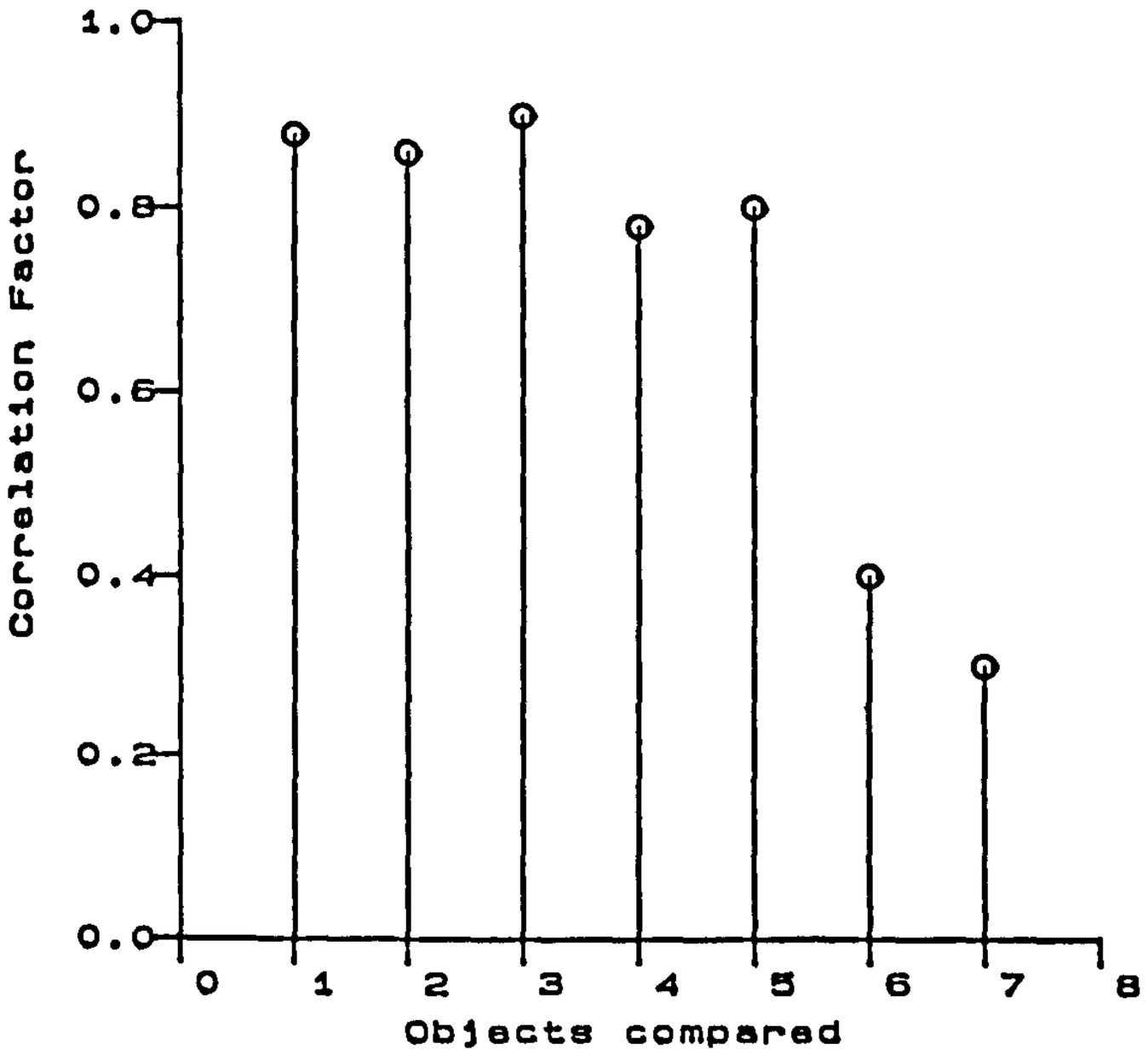
Figure 6.11 Recognition of the seven bladed fan using segmentation radius-angle power spectra, normalisation and the measurement of D_{max}.



The original seven bladed fan is compared to:-

1. its rotation version
2. opposite side viewed
3. light variation (at 60°)
4. scaled version (20% larger)
5. two blades missing
6. the four bladed fan
7. the gear wheel

Figure 6.12 Recognition of the seven bladed fan using radius-angle power spectra and the measurement of correlation factor.



The original seven bladed fan is compared to:-

1. its rotated version
2. opposite side viewed
3. light variation (at 60°)
4. scaled version (20% larger)
5. scaled version (10% larger)
6. the four bladed fan
7. the gear wheel

Figure 6.13 Recognition of the seven bladed fan using segmentation radius-angle power spectra, normalisation and the measurement of correlation factor.

It is also confirmed, as expected, that texture edges when used do not provide good feature vectors for object recognition, as most of the texture information is disregarded. Segmented images (the averaged neighbourhood partitions) provided a good basis for the resulting feature vectors, whilst texture edges could be useful in some other aspects of this type of study, e.g. contour tracking.

As a 'first check', prior to developing the feature vectors for the recognition process the 'compactness factor' of the images were considered. It is noted that for a given shape approximated by the square grid the factor of $(\text{perimeter})^2/\text{area}$ remains sensibly constant, even after scalings and rotations. It was shown in Chapter 4 that changes of up to 8% can be expected in area and perimeter measurements due to the square grid approximation and as a result of rotation. Therefore, in the recognition process the compactness factor of the test shape was obtained and checked against that of the reference shape. A change of up to 10% was tolerated, to allow for the errors due to the approximations and bad lighting conditions (thereby causing errors in perimeter and area approximations); if a change of greater that 10% occurred the recognition process was not continued. This is a much quicker 'first check' than that of tracking the boundary and developing Fourier Descriptors for recognition of the outer boundary, as described in the previous chapter.

The techniques developed for the recognition of objects takes the whole image into consideration and not the outer contour as it did in the previous chapter. The feature vectors developed are based on the surface textures of the object, yet the method is unable to distinguish which side of the 7 bladed fan is viewed, i.e. it could not recognise the textural differences in the boss geometry which vary from one side to the other. Methods used in image synthesis, e.g. developing surface normal vectors and the study of surface reflectivity and spectral properties, might provide the extra information to overcome this recognition problem for a simple engineering component.

CHAPTER SEVEN

APPLICATION OF SHADING MODELS IN IMAGE ANALYSIS

The techniques utilized in 'shape from shading' are examined and an attempt is made to employ image synthesis methods to discriminate between various components.

7.1 INTRODUCTION

It was shown in the previous chapter that the methods developed for the analysis of images, using surface features, could not recognise the differences in the boss geometry of the seven bladed fan. Image synthesis techniques search for attributes which provide extra information that might be employed to discriminate between surfaces of different properties (reflection and spectral properties) as well as for different surface shapes of the same property.

This chapter examines the 'shape from shading' technique that could replace the need for direct surface measurements and in addition provide information for surface recognition. Several models for surface shading prediction are examined and the process of applying these surface shading models for image analysis and depth preception is presented.

7.2 SHAPE FROM SURFACE SHADING

The prediction of shape from surface shading is an important attribute that might aid image synthesis. It involves the assigment of a grey level value to every picture element in the image, thereby simulating the viewing situation.

The shading of a surface point depends on the surface reflection characteristics, the surface geometry, and the lighting conditions [1]. Each of these properties must be considered in the development of a surface shading model.

The most important property in describing the contribution of the surface geometry to surface shading is the 'surface normal vectors' at the point of intersection of the incident light [2]. The normal vector is proportional to the reciprocal of the gradient of the surface at the point. Hence, if $\underline{i}$, $\underline{j}$, and $\underline{k}$ are the unit vectors in the x, y and z direction and $\underline{n}$ is the normal vector for a plane surface, f, represented by:

$$f(x, y, z) = ax + by + CZ + d = 0$$

then, the normal vector is [3]:-

$$\underline{n} = \frac{\partial f}{\partial x}\underline{i} + \frac{\partial f}{\partial y}\underline{j} + \frac{\partial f}{\partial z}\underline{k}$$

or

$$\underline{n} = a\underline{i} + b\underline{j} + c\underline{k}$$

A second characteristic of a surface is the reflection property corresponding to a given surface type. Surface properties include: spectral reflectance, determining how the surface reflects light of specific wavelengths; texture, determining the 'diffusivity' and 'specularity' components of reflection; and transparency, which determines the amount of

light that is refracted by the surface. These characteristics of surfaces identify the shading model required for surface emulation [1].

The shading models most frequently encountered in the literature are,

(i) Lambert's Law for diffuse reflectors [4].

(ii) Phong's model for simulation of specular reflectors [5].

Lambert's Law states that a surface will diffuse incident light equally in all direction. For a Lambertian surface, the quantity of reflected light is proportional to the amount of incident light. The intensity of a surface point (I) may then be modelled by the cosine of the angle between the normal vector ($\underline{n}$) to the surface point and the light source vector ($\underline{l}$), see Figure 7.1. The cosine of this angle (θ) may be computed by the inner product of the two vectors. The Lambertian model may, therefore, be summarised by [4]:

$$I = \begin{cases} \cos\theta & -\pi/2<\theta<\pi/2 \\ 0 & \text{otherwise} \end{cases} \qquad (7.1)$$

Phong's model for simulating the reflection characteristics of a specular surface suggests that more light is reflected in the direction of the source, forming an equal

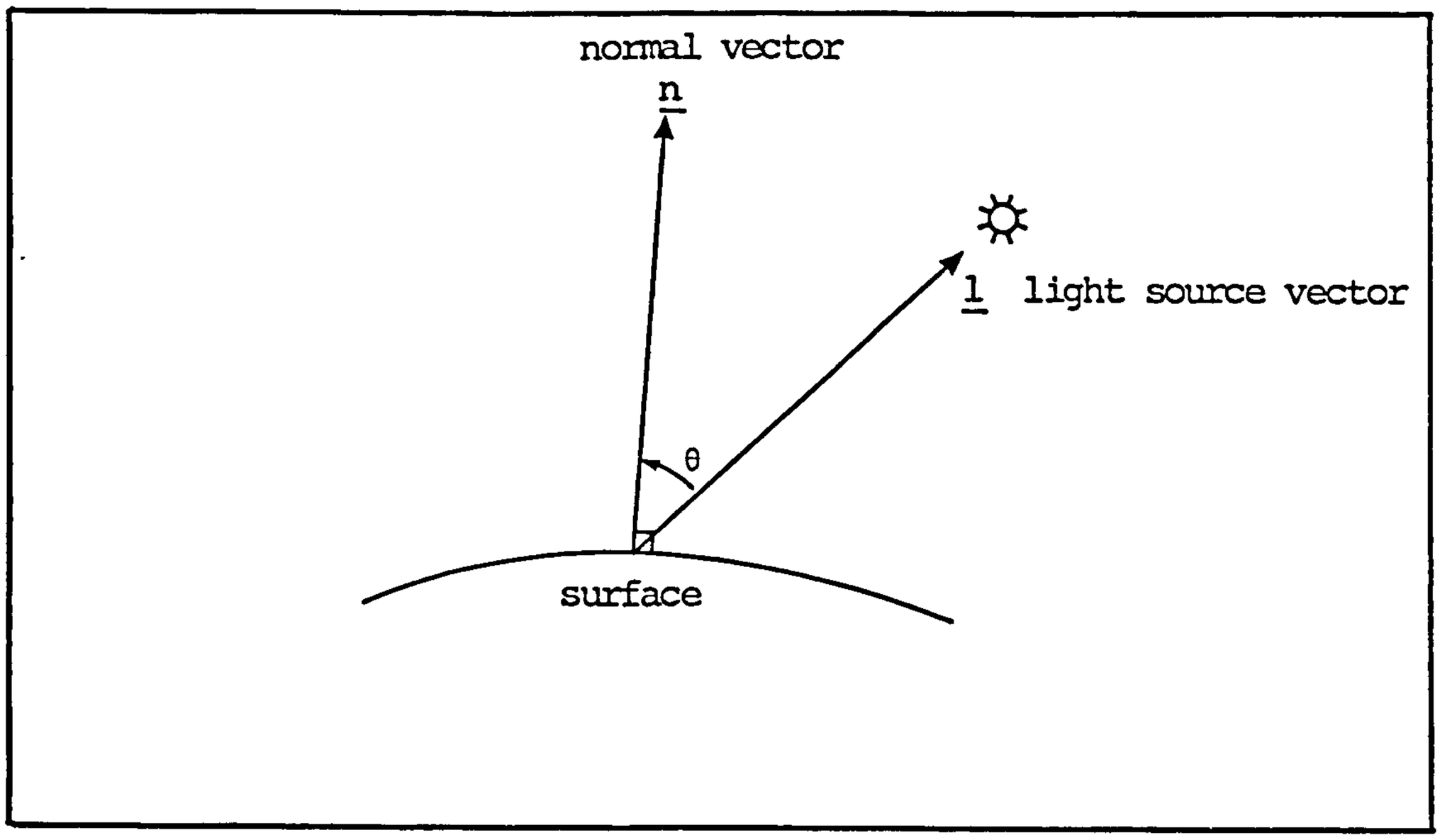


Figure 7.1 Lambert's law for diffuse reflector.

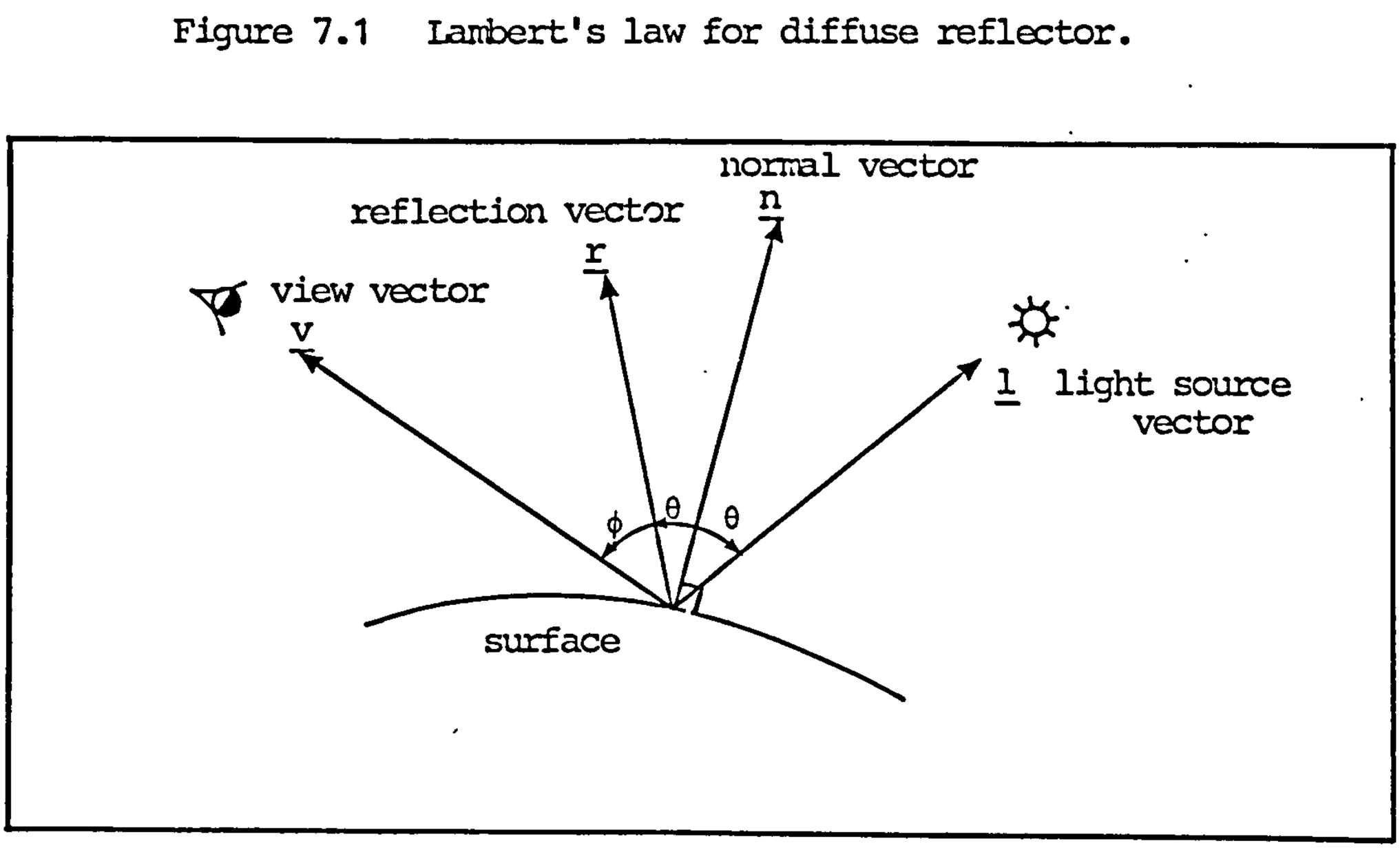


Figure 7.2 Phong's model for specular reflection.

but opposite angle to the normal vector, see Figure 7.2. This reflection vector may be determined by solving:

$$\underline{r} = \sqrt{2-2\cos(2\theta)} \quad \underline{n} - \underline{l}$$

where θ is the angle between $\underline{l}$ and $\underline{n}$. The intensity of a point on a specular surface is dependent on the viewing geometry, unlike an ideal Lambertian surface. For a mirror-like surface, the intensity as seen by the camera, would be zero everywhere except for the points which reflect the light exactly in the direction of the camera. For a duller surface, the intensity falls as the angle between the viewing vector and reflection vector increases. The relationship between the intensity and the angle ϕ, i.e. the angle between the view and the reflection vectors, may be modelled by [5]:

$$I = \begin{cases} \cos^{m}\phi & -\pi/2<\phi<\pi/2 \\ 0 & \text{otherwise} \end{cases} \qquad (7.2)$$

where the coefficient, m, is a measure of the shininess of the surface. A mirror-like surface would correspond to a larger value of m, and a duller surface would exhibit a value near unity. Typical values for surfaces such as nickel would be between 10 and 100, depending on the surface condition [6], i.e. dull and matt to cleaned and polished.

Most real surfaces are neither ideal diffuse reflectors nor ideal specular reflectors [7]. A model which combines the effects of diffuse and specular reflectance with ambient light produces good synthesized images. The ambient component is

light which is assumed to be uniformly distributed over the surface and the combination of these components yields the reflection model:

$$I = a \cos(\phi) + b \cos^{m}(\phi) + c \qquad (7.3)$$

The terms a, b and m depend upon material and surface roughness and c is the ambient or natural light term. The received intensity depends on these parameters, as well as the surface and viewing geometries [7]. The intensity, I, will have to be in the range of 0 to 1 to satisfy the inverse cosine requirement, the sum of the coefficients a, b and c must, therefore, never exceed unity. Most materials, such as copper or bronze, when smooth and glossy, have a large specular component and a small diffuse component (b = 0.7 to b = 1.0 and a = 0.0 to a = 0.3). Rough non-metallic surfaces, such as carbon or rubber, have large diffuse components and a small specular component.

The three models described by Equations (7.1),(7.2) and (7.3) may be used to obtain three-dimensional information that represent the surface shape by examining the observed intensity of a set of surface points.

7.3 APPLICATION OF THE SURFACE SHADING MODELS FOR SURFACE GEOMETRY DEFINATIONS

The shading of surface in image synthesis is modelled by functions of the surface geometry hence it is possible to derive the surface geometry from the shading information. Horn [3] suggests that the intensity of a surface point identifies the solution space for the normal vector to the surface at that point if the viewing geometry and lighting conditions are known. The process of determining the surface normal at a point consists of performing an inverse process of image synthesis. This process has been applied to polyhedral objects by Macworth [2] and shown to give good results for surfaces in which the reflection characteristics are known. McPherson [1], also, uses the surface shading models discussed in the previous section to obtain the surface geomtry of an object from the shading of the surface in the image.

Generally, the technique of surface measurement from surface shading provides a set of normal vectors for the surface points that are viewed by the camera. Unfortunately, the data obtained is not in the form of discrete surface point coordinates and hence the derivation of surface geometry is not simply a problem of curve fitting.

Most man-made objects can be described by multiple intersecting planes (polygonal solid), by multiple intersecting quadrics, or a combination of planes and quadrics [8]. Having obtained a function that describes the surface and estimates of the surface normals, the unknown parameters of the surface tangent at a point on the surface can be evaluated. To achieve this, the first derivative of the surface function is equated to the derived surface vector at the measured data point. The coordinates of the centre of the camera lens and a knowledge of the camera characteristics provide further data and subsequently permits the unknown parameters of the mathematical model of the surface to be calculated [1].

7.4 SURFACE NORMAL VECTOR ESTIMATION AND APPLICATION IN SURFACE RECOGNITION

The Lambertian model for diffuse surfaces, Equation (7.1), measures the intensity of a surface pixel by the cosine of the angle between the light source vector and the normal vector to the surface at that point. The equation may be rewritten, for the arrangement shown in Figure 7.3, as:

$$I = \cos (\alpha - \beta)$$

Clearly, for any intensity (pixel voltage) there are two possible solutions for the angle θ i.e. $(\alpha - \beta)$. If the direction of the light source (α) is known, then two possible

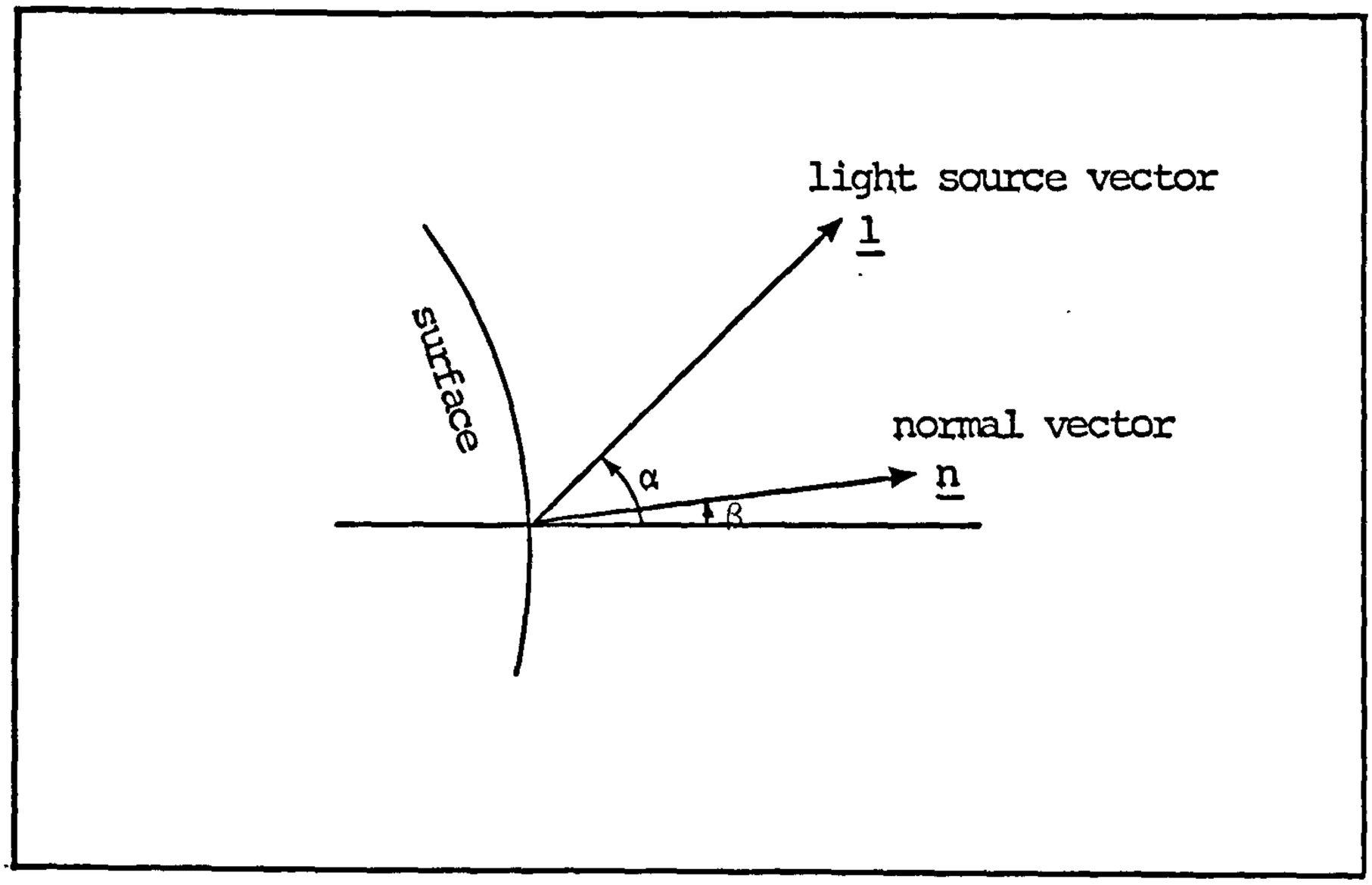


Figure 7.3 Angles associated with diffuse model.

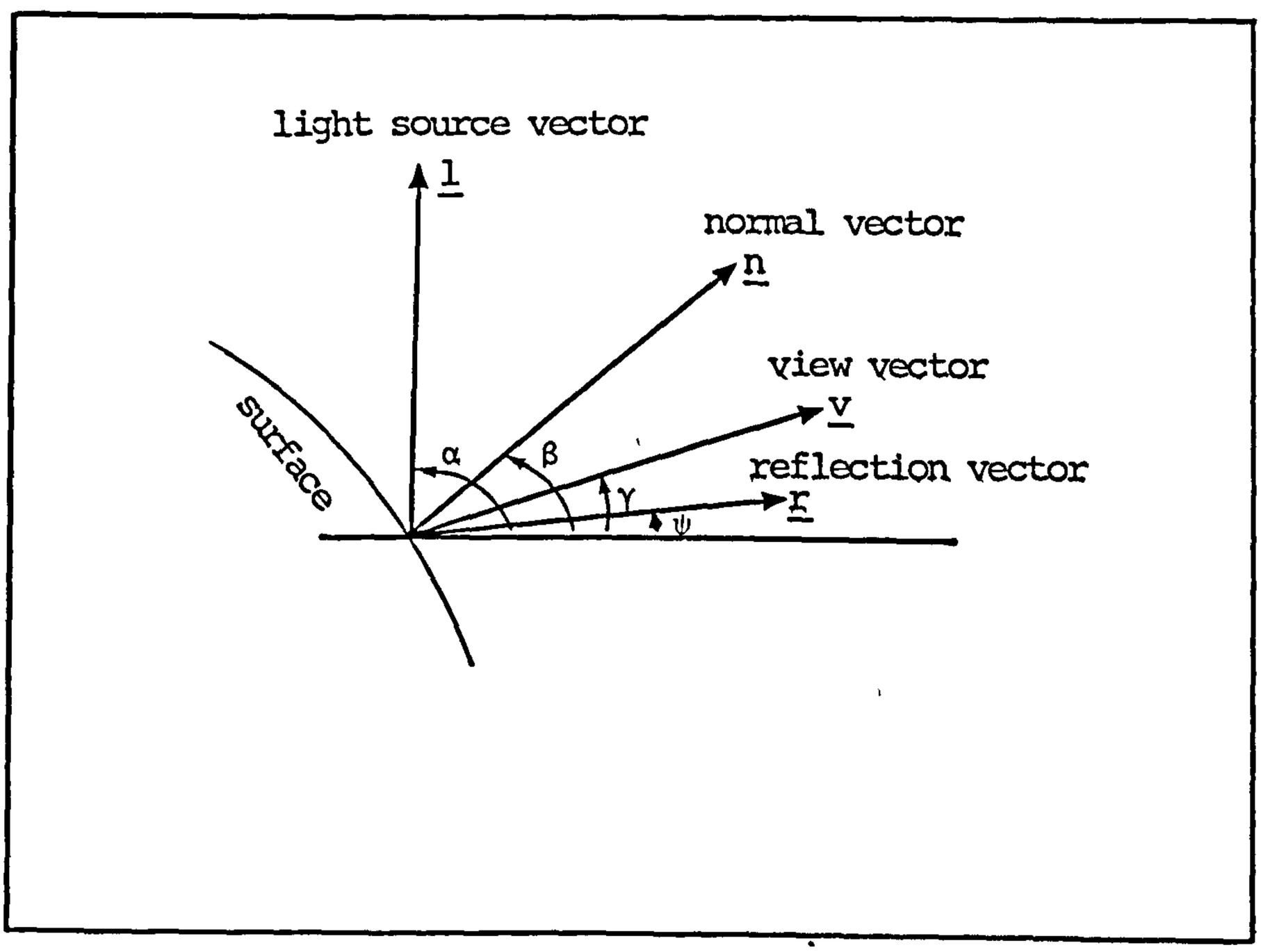


Figure 7.4 Angles associated with specular model.

solutions exist for the direction of the normal vector (β). To be able to identify the correct normal vector, two different positions of the light source must be used and the common value from the data is the true normal.

For the specular model, if the angular displacement of the view vector, $\underline{V}$ in Figure 7.4, is represented by γ and the angular displacement of the reflection vector ($\underline{r}$) by ψ, Equation (7.2) becomes:

$$I = \cos^{m} (\gamma - \psi) \qquad (7.4)$$

Hence, for the specular model, as shown in Figure 7.4, the direction of the normal vector (β) is halfway between the reflection vector (ψ) and the light source vector (α), therefore:

$$\beta = \frac{1}{2} (\psi + \alpha)$$

or

$$\psi = 2\beta - \alpha$$

Substituting for ψ in Equation (7.4) the specular model of surface shading becomes:

$$I = \cos^{m} (2\beta - \alpha - \gamma)$$

For both models the direction of the surface normal is obtained from:

$$\text{(diffuse)} \qquad \beta = \alpha \pm \cos^{-1}(I) \tag{7.5}$$

$$\text{(specular)} \qquad \beta = \frac{1}{2}\left[\alpha + \gamma \pm \cos^{-1}\left((I)^{1/m}\right)\right] \tag{7.6}$$

In an attempt to distinguish between the different sides of a component, experiments were carried out to estimate the directions of the surface normals at points of interest. In the first experiment a white matt cardboard cube was examined. The reflectance properties of such surface finishes resemble those of the Lambertian type. These artificially created surfaces do not have ideal diffuse reflection properties, therefore values of m=1, m=2 and m=4 were chosen for the specular shading model of Equation (7.6), and the direction of surface normal vectors for the plan view of the cube were obtained, i.e. with a priori knowledge that they should be close to 90^{o}. To identify the true direction of the surface normal, the light source was initially positioned vertically above the cube, whence,

$$\underline{l} = \underline{j} \qquad \text{or} \qquad \alpha = 90^{o}$$

The second position of the light source made an angle (α) of 120^{o} with the top face, yielding

$$\underline{l} = -0.5\underline{i} + 0.866\underline{j}$$

The viewing position, i.e. the camera's position, remained unaltered throughout, at an angle of $\gamma = 45^{o}$ to the face, i.e.

$$\underline{v} = 0.707\underline{i} + 0.707\underline{j}$$

The 'mean' and 'standard deviation' values for the pixel voltage distribution, in the face considered, were calculated to examine the light variation in the flat surface as the light source position was altered. The standard deviation remained within the range of 0.002 to 0.02 from the mean value, or in percentage terms, 0.2% to 2.3% + or - the mean, exhibiting a uniform light distribution over the surface. The pixel values were normalised to fall within the range of zero to unity whereby each pixel was divided by the maximum pixel value detected on the surface region. The separate regions were monitored to give some indication of the spread of results to be expected. Table 7.1 summarises the results obtained.

The trials were repeated for the fans shown in Figure 5.7, in an attempt to identify differences in blade curvature between opposite sides. It was anticipated that the concave and convex surfaces would have different reflective properties and

KEY

MEAN:	Mean of the grey levels.
STANDARD DEVIATION:	Standard deviation of the grey levels.
I:	Intensity of the point being considered.
NV:	Estimate direction of the Normal Vectors.
Df:	Diffuse Modle.
Sp:	Specular Model.

Light Direction		90°	90°	90°	120°	120°	120°
MEAN		0.9928			0.8725		
STANDARD DEVIATION		0.005			0.02		
I		0.9887	1.000	0.9896	0.8498	0.8683	0.8995
NV using DF		81.378°	90°	81.729°	88.190°	90.262°	94.092°
Variation, in percentage from 90°		9.6%	0	9.19%	2.01%	0.29%	4.55%
NV using Sp, M = 1		71.810°	67.5°	71.635°	83.405°	82.369°	80.454°
Variation, in percentage from 90°		20.2%	25%	20.4%	7.3%	8.5%	10.6%
NV using Sp M = 2		70.551°	67.5°	70.472°	78.901°	78.139°	76.741°
Variation, in percentage from 90°		21.6%	25%	21.7%	12.3%	13.2%	14.7%
NV using Sp M = 4		69.658°	67.5°	69.570°	75.617°	75.068°	74.064°
Variation, in percentage from 90°		22.6%	25%	22.7%	15.9%	16.6%	17.7%

Table 7.1 Estimation of the direction of the surface normal for one face of the cube.

hence the sets of derived normals would be unique for each side of the fan blades.

The maximum curvature changes occur at the tips of the blades, i.e. from inlet to outlet of the cross-section. Table 7.2 is provided to confirm this property. The Table is obtained by measuring the surface normal directions, directly from a trace of the blade surface being examined, using a Talycontor. A variation of up to 30^{o} is evident in the normal directions of the concave side of the blade tips, and up to 20^{o} for those of the convex face. Hence, in calculating the surface normals only the cross-section of the tips were examined.

Table 7.3 illustrates a typical set of results obtained for one of the four blades of the four bladed fan. The two shading models, and different values for 'm' in the specular model have been used to investigate which model and shininess factor would result in estimated normal directions close to those of Table 7.2. The lighting and camera position remained unchanged from those of the previous Tables. An average value for the direction of the normals has been calculated where the light variation could not result in deriving an identical solution for the angle β, Equations (7.5) and (7.6).

In the trials of the above experiments the position of the camera and the light source directions i.e. α and γ, are constant and the only variable is the intensity of the surface

	Inlet β										Outlet				Range
Measurements of the surface normals (β) for the concave side of the blade	47°	55°	57°	59°	62°	67°	70°	74°	75°	75°	77°	78°	79°	77°	30°
Measurements of the surface normals (β) for the convex side of the blade	78°	76°	79°	78°	76°	75°	76°	72°	71°	67°	62°	60°	58°	58°	20°

Table 7.2 Measurements of the surface normal direction (β) for the tip of one blade using Talycontor equipment.

KEY

I Intensity of the point being considered.
CV Concave side of the blade.
CX Convex side of the blade.
NV Estimated direction of the Normal Vectors.
Df Diffuse model.
Sp Specular model.

			0.8878	0.8693	0.8876	0.9198	0.8691				
			0.8391	0.8292	0.8481	0.8769	0.8183				
			0.8491	0.8892	0.8571	0.8261	0.8781				
			0.8269	0.8186	0.8391	0.7998	0.8489				
			75.0°	73.3°	75.5°	79.3°	72.8°				
			72.2°	74.0°	73.2°	69.6°	74.9°				
			74.3°	74.5°	74.4°	73.2°	74.4°				
			75.3°	74.1°	74.7°	75.8°	75.6°				
			74.5°	74.6°	74.6°	74.8°	74.6°				
			75°	74.3°	74.9°	75.2°	74.7°				
			74.7°	74.7°	74.6°	75.4°	74.3°				
			74.8°	74.6°	74.5°	75.2°	74.3°				
I, CV	$\alpha = 90^{o}$	CV									
I, CV	$\alpha = 120^{o}$	CV									
I, CX	$\alpha = 90^{o}$	CX									
I, CX	$\alpha = 120^{o}$	CX									
NV, Df		CV									
NV, Df		CX									
NV, Sp	M = 5	CV									
NV, Sp	M = 5	CX									
NV, Sp	M = 10	CV									
NV, Sp	M = 10	CX									
NV, Sp	M = 20	CV									
NV, Sp	M = 20	CX									

Table 7.3 Estimation of the normal vectors for the two sides of a blade of the 4 bladed fan.

pixel. For the fan blades considered, the radii of curvature are not small enough, for the lighting conditions tried, to produce intensity changes which could result in a better estimation of the normal directions , i.e. concavity and convexity of the blades are not sufficiently obvious. However, as the technique is intended to be used as an extension to the previous recognition processes, Chapters 4, 5 and 6, the shading of the surface is required to be minimum thereby avoiding high density shadows. It was stated in the previous Chapter, that the effects of low density shadows, in the recognition process could be minimised by segmenting the textured image. High density shadows cannot however, be tolerated in the recognition process, and whilst they provide useful criteria for the image sythesis techniques their effect on the recognition methods could be detrimental.

Table 7.4 is provided to present the possible shading effects which may be obtained for the same blade, by altering the lighting conditions. The small light source makes an angle of 150^{o} with the horizontal in this trial. The light variation on the surface provides a good basis for deriving normal directions which are distinct for each side of the blade, thereby recognising the different faces. Upon examining the surface intensities listed in Table 7.4, it becomes evident that changes of up to 40^{o} are detected in calculating the surface normal directions using the diffuse model, and up to 25^{o} using

	Inlet			Outlet			
Normalised intensities of the concave face of the blade	0.8989	0.8591	0.8015	0.7010	0.4028	MEAN of the intensity over surface 0.6772	STANDARD DEVIATION 0.2885
Normalised intensities of the convex face of the blade	0.4119	0.5581	0.6218	0.6981	0.8210	MEAN of the intensity over surface 0.6518	STANDARD DEVIATION 0.2391

Table 7.4 Shading of the blade when high density shadows are present.

the specular model, i.e. by keeping light source direction and camera's position unchanged and considering the inverse cosine of the listed intensity levels. Figure 7.5 is provided to summarise the results obtained in Tables 7.2, 7.3 and 7.4.

7.4.1 The Effect of Shadows in the Recognition Process and Further Application of the Specular Model

As discussed above, the presence of high density shadows cannot be tolerated in any of the recognition processes developed in the previous Chapters. To substantiate this fact, the four bladed fan was again examined to establish the effects of the presence of the heavy shadows, as used in deriving the intesity levels in Table 7.4, on the recognition processes. The 'similarity measure' of Chapter 5, rejected the fan when compared to its original shape, i.e. with 'normal' lighting condition. The same lighting condition, i.e. the presence of shadows, was repeated on the 7 bladed fan, and the recognition process in this case identified the fan as being close to the 4 bladed fan. These miscalculations resulted after thresholding the image to obtain a binary, object/background, image. Further investigations showed that intensity variations i.e. shading, of up to 10% could be tolerated, and the shadows presented could then be classed as being of low density. In Table 7.4, changes in intensity of up to 60% are present, well outside this allowable range. However, the shadows could still provide useful information by adopting a two stage recognition strategy.

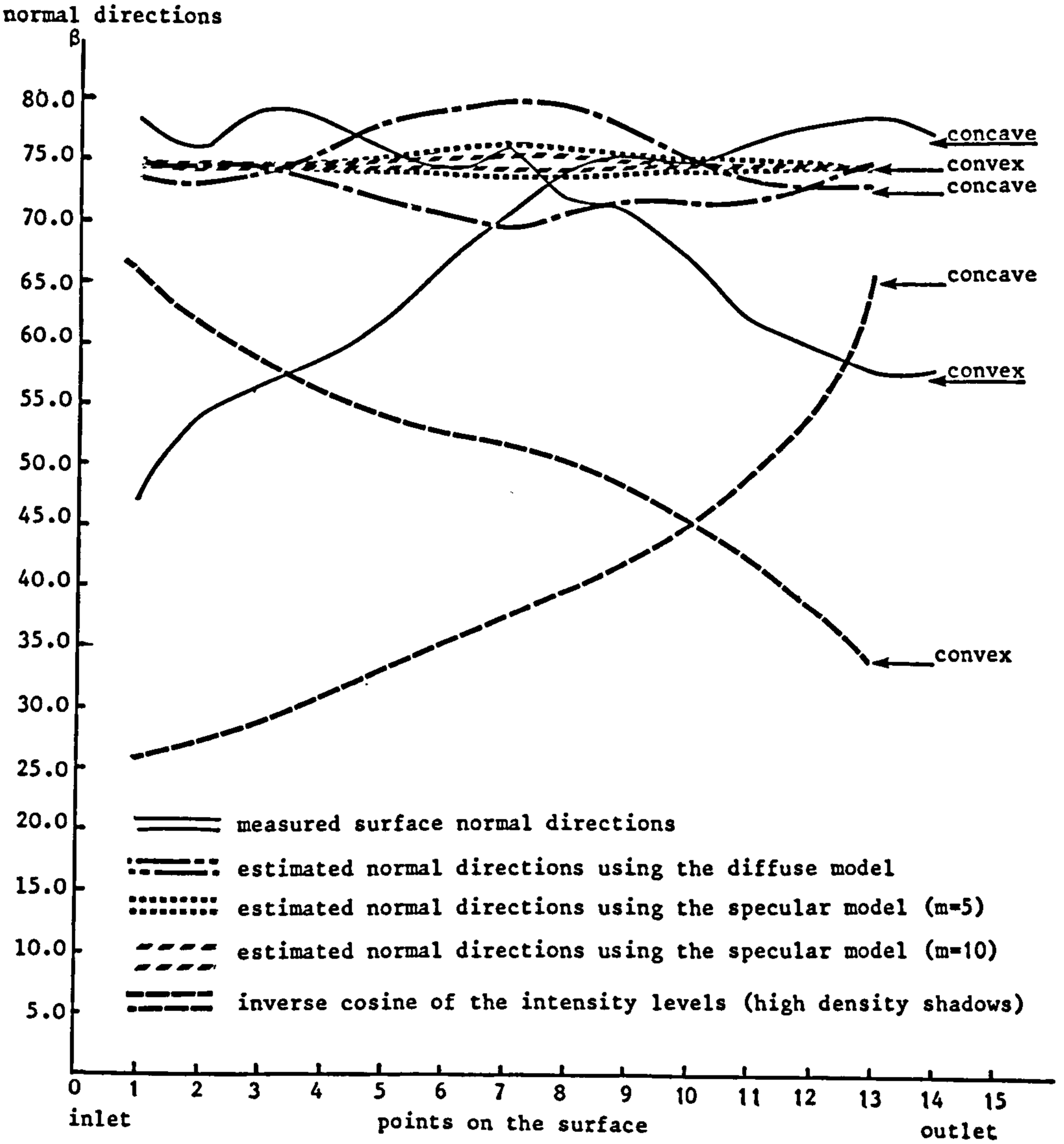


Figure 7.5 Actual and estimated normal directions for opposite sides of one blade of the four bladed fan.

That is, the recognition of the component must be confirmed under lighting conditions which will minimise the presence of shadows. Then, lighting may be adjusted to use the surface geometry details to create shadows and utilize the shading models to distinguish between the different faces of the object.

The specular model may be used to derive further information regarding the surface geometry even when the light intensity does not change significantly over the surface. As mentioned earlier, the small variations in the direction of the normals, Table 7.3, are due to the small changes in intensity and can be determined from the term $\cos^{-1}\{(I)^{1/m}\}$ of the Equation (7.6). To study these variations and to investigate the impact of the shininess factor, i.e. m, on the calculation of the surface normal directions for the four bladed fan, Table 7.5 has been provided. In this, the shininess factor has been altered by: first, masking the tip of the blade being considered by shiny masking tape, thereby making the surface more reflective; secondly, the blade considered was sprayed with thin transparent oil, making the surface even more reflective. The value of m was estimated, respectively, as 20 and 30. The percentage changes in Table 7.5 were obtained by considering, only, the cosine term of the specular model and by taking the smallest intensity point as the datum, calculating the percentage variations from this datum point.

'm' = 10	Mean value of the grey levels on the surface	0.8438	concave			
		0.8280	convex			
	Standard deviation of the grey levels on the surface	0.0205	concave			
		0.0168	convex			
	Cosine term (concave face)	11.03°	10.30°	9.14°	10.67°	11.40°
	% change	3.2%	9.6%	19.8%	6.4%	0
	Cosine term (convex face)	11.38°	10.67°	12.06°	11.03°	10.30°
	% change	5.6%	11.5%	0	8.5%	14.6%
'm' = 20	Mean value of the grey levels on the surface	0.8460	concave			
		0.8679	convex			
	Standard deviation of the grey levels on the surface	0.03	concave			
		0.03	convex			
	Cosine term (concave face)	8.54°	7.55°	5.88°	7.03°	7.81°
	% change	0	11.6%	31.1%	17.7%	8.5%
	Cosine term (convex face)	7.29°	6.47°	5.23°	7.29°	7.55°
	% change	3.4%	14.3%	30.7%	3.4%	0
'm' = 30	Mean value of the grey levels on the surface	0.9100	concave			
		0.9040	convex			
	Standard deviation of the grey levels on the surface	0.01	concave			
		0.01	convex			
	Cosine term (concave face)	4.80°	5.05°	3.98°	4.27°	4.79°
	% change	0	5.2%	17.1%	11.0%	0.2%
	Cosine term (convex face)	4.54°	5.28°	4.27°	4.54°	4.80°
	% change	14.0%	0	19.1%	14.0%	9.1%

Table 7.5 Examining the changes in the cosine term of the specular model for estimating the normal vectors of the opposite sides of the 4 bladed fan, as the shininess is altered.

7.5 DISCUSSION AND FINAL REMARKS

Two different models for surface shading are presented that are easy to apply experimentally. A list of criteria for the selection of the most appropriate model, however, has not yet been established in the literature.

. For the trials conducted on opposite sides of the fan blades, the variations in direction of the estimated surface normals have been shown to be not significantly different from one side to another when low values of shading, necessary for good object recognition, exist on the surface. These differences are not sufficient to provide a basis for discrimination between the two sides, further, they do not follow the actual trend of the measured normals, given in Table 7.2. This is due to the shape of the blades and the lighting conditions chosen. The impact of the lighting variations and the presence of shadows, first on the recognition processes and then on the surface normal estimations, have also been discussed. A further application of the specular model, has been evaluated which considers the percentage changes in the cosine term only of the model across the blade surface. The results obtained, Table 7.5, could provide a basis for distinguishing the opposite sides of the fan considered, even though the intensity variations are small.

A two-stage recognition process is suggested. In the first, the shading of the surface and the presence of shadows must be kept to a minimum for good recognition. For the second stage, the lighting conditions must be arranged to provide maximum shading models. However, to establish a general model for the type of surfaces examined, and to arrive at an optimum value for the surface shading models to provide sensible estimation for the direction of the normal vectors, requires further investigation.

C H A P T E R E I G H T

DISCUSSION

The most important observations made in the forgoing chapters are summarised and discussed.

8.1 LITERATURE REVIEW

The goals of the machine vision system, namely the effective interpretation of visual data for component inspection and robot control, are reported in the literature, as having been achieved in limited task capabilities with operational environment restrictions.

The initial published work, (Chapter 1, references [6-9]), concentrates on quantitative studies of shape and pattern perception, by adopting a psychological approach and human perceptual models. This work does not provide a completely acceptable solution to the industrial applications identified. Based on these initial observations a number of workers, (Chapter 1, references [10-13]), have detailed algorithms and schemes which could be incorporated into basic industrial vision tasks.

Artificial Intelligence concepts and the knowledge-base approach have been used in information representation and processing. Support for the inclusion of the rule-based approach in the design of industrial vision systems to overcome some of the observed shortcomings in performance has been suggested (Chapter 1, references [21-24]). The complexity of the computational methods and the non-versatility of the computer

languages employed, together with the relatively large data-base requirement and high processing power needed, would not, in this author's opinion, make them suitable for immediate industrial application.

A number of researchers, (Chapter 2, references [52-59]), have designed and developed industrial vision systems for inspection and robot control, performing tasks of various degrees of complexity. These examples provide a good basis for comparison between systems incorporating specialised hardware, typically stereo vision and range finder sensors, and systems using only a single camera for image data acquisition.

8.2 PRE-PROCESSING AND CONTOUR TRACING

Illumination plays a critical part in most image processing applications. It is, indeed, for this reason that industrial vision systems have been defined as 'computer vision with controlled lighting'. Digital image processing methods are at the heart of any computer vision system, that improves and processes pictorial data for use in automatic machine perception. A basic part of the image information is in its edge or boundaries. It has been shown by experiments on human vision systems that the image boundary is of major importance in recognition, (Chapter 3, reference [2]). For the object recognition algorithms, where the boundary representation has

been incorporated into, image enhancement and techniques for detecting the subtle edges are of principal importance. Chapter 3 documents the principle edge detection techniques and compares the performance of different methods. It has been shown in this chapter, that no one technique was significantly superior to the others and the choice would be dictated by the image processing situation.

Techniques have been developed whereby the outer contour of the object image is traced and coded. The codes described are independent of object orientation. One such technique, which has potential applications in an engineering environment, is the Freeman's code, (Chapter 4, reference [1]). Codes which are spatially invariant are used for both object recognition, and the estimation of object orientation

.

Image enhancement together with the edge detection routine will ensure that the boundaries are well defined and distinct from the background. These preprocessing procedures are performed in addition to thresholding, to minimise the effects of shadows and to give a better defined contour for recognition purposes.

8.3 FOURIER ANALYSIS IN OBJECT RECOGNITION

The use of Freeman's code to represent the contour of a shape results in the generation of a numerical series which

significantly reduces the number of data points to be processed compared with the original pixel array, whilst still retaining the contour properties of the shape. The resulting series were used, Chapter 4, for subsequent identification of an object and its orientation estimation. Further additional data reduction is achieved by Fourier analysis of this chain code data.

In the work presented in Chapter 5, features which are invariant with respect to translation, rotation and object size were obtained from the use of normalised Fourier Descriptors (FD). Theoretical and experimental evidence is available, (Chapter 5, reference [3, 4]), which indicates that the FD is more powerful than other boundary representation techniques, such as the polygon approximation. Two methods were investigated for the evaluation of Fourier Coefficients. The coordinate values of the contour pixels were used in the first, and the actual orientation vectors of the Freeman's chain code were used in the second method. The inverse Fourier transforms were developed and the reconstruction of the shapes confirmed that only a small number of Fourier Coefficients were needed to reproduce the shape for reliable recognition. Methods were developed whereby the error due to the approximation of the boundary by a small number of FD could be predicted.

The proposed procedure for the selection of the sub-optimum number of FD necessary for reliable recognition, will be particularly advantageous in applications where a large number of objects are to be identified and the images monitored are to be checked against a reference library of shapes.

8.4 SURFACE TEXTURE ANALYSIS AND IMAGE SYNTHESIS

Contour tracing and coding of the boundary to represent shapes are shown to be useful for identification of some objects. It is generally believed, however, that additional features will, in many circumstances, be necessary in a real engineering environment to allow for the differences in front and back faces of an object to be recognised.

As a step forward, surface features and properties relating to the surfaces were investigated. Texture analysis for surface definition has been used by a number of researchers, (Chapter 6, references [1-3]). In most of the published literature, range data measurements have been employed to aid the segmentation of the textured image and to detect surface orientation, using data from specially designed hardware in the analysis.

The technique adopted by this author, Chapter 6, was based on the analysis of a perceived structure and was built by

extending the methods described earlier. An object's surface was studied and useful features were extracted from the obtained power spectra. The impact of shadows, the application of enhancement and segmentation were demonstrated. For the recognition process itself, two different methods for the comparison of the feature vectors were investigated.

Textural analysis, useful as it is for the identification of simple engineering components, has a weakness in that it could not detect relatively major surface features. It could not for example distinguish between the opposite sides of the fans used in Chapter 5. Attributes which provide extra information that could be used to discriminate between surfaces of different properties, can be found by using image synthesis techniques. Several models for the prediction of surface shading were examined, Chapter 7, and the process of applying these models for surface feature detection was presented. For the fans employed in the study, the large radius of curvature prevented the concavity and convexity of the opposite sides being detected, using the two shading models discussed, unless high density shadows were created on the surface. It was established for the recognition processes developed earlier, high density shadows, i.e. intensity variations of over 10% on the surface being considered, could not be tolerated. As a result of this, a two stage recognition process would be required if both methods are necessary in component identification.

In a final attempt to extract usable information from the small changes in the light intensity levels over the surface, the variant term in the models, namely the 'cosine term' was monitored. The Tables obtained, Chapter 7, suggest that the percentage changes in the cosine term could be adopted to provide a basis for distinguishing the opposite sides of the components considered.

CHAPTER NINE

CONCLUSIONS

9.1 CONCLUSIONS

The major conclusions that can be drawn from this work are:-

(1) Image edge detection should normally proceed the boundary following algorithm.

(2) Freeman's chain code provides useful spatially invariant sequences for use in object recognition and orientation estimation.

(3) Fourier analysis allows Descriptors to be derived which are independent of an object's silhouette size and orientation. The contour data is significantly condensed and aids multiple object recognition.

(4) Surface textural features will generate extra data applicable to the object recognition process.

(5) Shapes from shadings and the models utilised will distinguish between various surfaces. The lighting conditions required for the shading models will, however, differ from those needed for object recognition; hence, multi-stage strategies may have to be employed.

REFERENCES

CHAPTER 1

[1] KITTLER, J.V., "PRIP review", Technology Division, SERC Rutherford Appleton Laboratory report, May 1982.

[2] LEVIALDI, S., "Digital image analysis", Pitman Publishing Ltd, London, 1984.

[3] GONZALEZ, R.C. and WINTZ, P., "Digital image processings", Addison-Wesley Co., Reading, MA, 1977.

[4] NEVATIA, R., "Machine perception", Prentice-Hall, New York, 1982.

[5] KELLEY, R.B., MARTINS, H.A.S., BIRK, J.R. and DESSIMOZ, J.D., "Three vision algorithms for acquiring workpieces from Bins", Proc. of the IEEE, Vol. 71, No. 7, July 1983, pp. 803-820.

[6] SELFRIDGE, O.R. and NEISSER, U., "Pattern recognition by machine", Sci. Am., 203, August 1960, pp. 60-68.

[7] ATTNEAVE, F. and ARNOULT, M.D., "The quantitative study of shape and pattern perception", Psychol. Bull., 53, 452 (1956).

[8] LANGDON, J., "The perception of 3-D solids", Quart. J. Exptl. Psychol., 7, 1955, pp. 19-36.

[9] STEVENS, S.S., "The psychophysiology of vision", Sensory Communication, W. Rosenbith (ed.), M.I.T. Press, and John Wilet and Sons Inc., New York, 1961.

[10] ROBERTS, L.G., "Machine perception of three-dimensional solids", M.I.T. Press, Cambridge, MA, 1965.

[11] FISCHER, G.L., POLLOCK, D.Y., RADAK, B. and STEVENS, M.E., "Optical character recognition", Spartan, Baltimore, 1962.

[12] ROSENFELD, A. and KAK, A.C., "Digital picture processing", second edition, Academic Press, New York, 1982.

[13] SKLANSKY, J. and WASSEL, G.N., "Pattern classifiers and trainable machines", Springer-Verlag, New York, 1981.

[14] YACHIDA, M. and TSUJI, S., "Industrial computer vision in Japan", IEEE Computer, May 1980, Vol. 13, No. 5, pp. 50-63.

[15] YACHIDA, M. and TSUJI, S., "A versatile machine vision system for complex industrial parts", IEEE Trans. Computers, Vol. C-26, No. 9, pp. 882-894.

[16] EJIRI. M., LINO, T., MESE, M. and IKEDA, S., "A process for detecting defects in complicated patterns", Computer Graphics and Image Processing, Vol. 4, June 1973, pp. 326-339.

[17] KASHIOKA, S. EJIRI, M. and SAKAMOTO, Y., "A transistor wire-bonding system utilising multiple local pattern matching techniques", IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-6, No. 8, 1976, pp. 562-569.

[18] AGIN, G.L., "Computer vision system for industrial inspection and assembly", IEEE Computer, Vol. 13, No. 5, pp. 11-20.

[19] GONZALEZ, R.C. and SAFABAKSH, R., "Computer vision techniques for industrial inspection and robot control: A tutorial overview", Tutorial on Robotics, IEEE Computer Society, EH0207, 1984, pp. 300-324.

[20] KASSLER, M., "Robot Vision", Inst. of Radio and Electr. Eng., Australia, (IREE Aust) Vol. 1, No. 2, June 1981, pp. 105-107.

[21] NEUMANN, B., "Vision Systems: state-of-the-art and prospects", Artificial Intelligence: Toward Practical Applications, T. Bernold and G. Albers (eds.), Elsevier Science Publishers, 1985, pp. 51 -61.

[22] McDERMOTT, J., "A rule-based configurer of computer systems", Artificial Intelligence, Vol. 19, 1982, pp. 39-88.

[23] MAGEE, M. and AGGARWAL, J.K., "Intensity guided range sensing recognition of three-dimensional objects", Proceedings of the Conference of Computer Vision and Pattern Recognition, Washington D.C., June 1983, pp. 550-552.

[24] MAGEE, M. and NATHAN, M., "A rule based system for pattern recognition that exploits topological

constraints", Proceedings of the Conference on Computer Vision and Pattern Recognition, San Fransisco, June 1985, pp. 62-67.

[25] FREEMAN, H., "Computer processing of line-drawing images", Computer Surveys, Vol. 6, No. 1, March 1974, pp. 57-97.

CHAPTER 2

[1] PETAJAN, E.D., "Automatic lipreading to enhance speech recognition", Proceedings of Computer Vision and Pattern Recognition Conference, San Francisco, June 1985, pp. 40-47.

[2] BRUSSEL, H.V. and BELIEN, H., "A high resolution on tactile sensor for part recognition". Proceedings of the 6th International Conference on Robot Vision and Sensory Controls, June 1986, Paris, pp. 49-59.

[3] FAVRE, J.M, and YONG, Y., "A visual servicing system for a coachwork assembly robot", Proceedings of AFCET Conference, Toulouse, October 1985, pp. 221-232.

[4] ROBERTS, L.G., "Machine perception of three-dimensional solids", Optical Electro-Optical Processing of Information, MIT Press, 1965.

[5] DUDA, R.O. and HART, P.E., "Pattern classification and scene analysis", Wiley, New York, 1971.

[6] PREWITT, J.M.S., "Object enhancement and extraction", Picture Processing and Psychopictorics, Academic Press, New York, 1970.

[7] FREEMAN, H., "Computer processing of line-drawing images", Computer Surveys, Vol. 6, No. 1, March 1974, pp. 57-97.

[8] GONZALEZ, R.C. and WINTZ, P., "Digital image processing", Addison-Wesley Co., Reading, MA, 1977.

[9] NIKRAVAN, B., BAUL, R.M. and GILL, K.F., "The Fourier Descriptor in closed curve representation for simple engineering object recognition", Paper submitted to the I.Mech.E. for publication.

[10] ARKING, A.A., LO, R.C.and ROSENFELD, A., "An evaluation of Fourier Transform Techniques for cloud motion estimation", Computer Science Technical Report Series, No. TR-351, University of Maryland, MD, USA, 1975.

[11] BADRELDIN, A., PRASAD, T., ISMAIL, M. and WONG, A.K.C., "Shape description for N-dimensional curves and trajectories", IEEE Proceedings on Cybernetics and Society, 1980, pp. 713-717.

[12] WALLACE, T.P. and WINTZ, P.A., "An efficient three-dimensional aircraft recognition algorithm using normalised Fourier Descriptors", Computer Graphics and Image Processing, Vol. 13, 1980, pp. 99-126.

[13] ROSENFELD, A., "Introduction to machine vision", IEEE Control Systems Magazine, Vol. 5, No. 3, August 1985, pp. 14-17.

[14] ROSENFELD, A., "Computer vision research for industrial applications", IEEE Proceedings of trends and applications, May 1983, pp. 43-49.

[15] LIN, J.C. and CHI, C.Z., "Accuracy analysis of a laser/camera based 3-D measurement system", Proceedings of the 3rd International Conference on Robot Vision and Sensory Controls (ROVSEC3), Novembver 1983, Cambridge MA., pp. 158-170.

[16] LEWIS, R.A. and JOHNSON, A.R., "A scanning laser rangefinder for a robotic vehicle", Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1977, pp. 120-138.

[17] PUGH, A., "Robot Vision", IFS (Publications) Ltd, U.K., 1983.

[18] JARVIS, R.A., "A computer vision and robotics laboratory", IEEE Computer, Vol. 15, No. 6, June 1982, pp. 8-22.

[19] HALL, E.L., TIO, J.B.K., McPHERSON, C.A., DRAPER, C.S. and SADJADI, F.A., "Measuring curved surfaces for robot vision", IEEE Computer Dec, 1982, pp. 42-54.

[20] McPHERSON, C.A., "Three-dimensional robot vision", Proceedings of the 3rd International Conference on Robot Vision and Sensory Controls (ROVSEC3), November 1983, Cambridge MA., pp. 116-126.

[21] HORN, B.K.P., "Understanding image intensities", Artificial Intelligence, Vol. 8, 1977, pp. 201-231.

[22] AUGUSTEIJN, M.F. and DYER, C.R., "Model-based shape from contour and point patterns", Proceedings of Computer Vision and Pattern Recognition Conference, San Francisco, June 1985, pp. 100-105.

[23] KANADE, T., KENDER, J., "Shape from texture paradigm", Technical report, Carnegie-Mellon University, Computer Science Department, Pittsburgh, USA, July 1980.

[24] ROSENFELD, A. and KAK, A.C., "Digital picture processing", second edition, Academic Press, New York, 1982.

[25] BROOKS, R.A., "Symbolic reasoning among 3-D models and 2-D images", Ph.D. Thesis, Department of Computer Science, Stanford University, USA, 1981.

[26] WINSTON, P.H., "Learning structural descriptions from examples", in the Psychology of Computer Vision, P.H. Winston (ed.), McGraw-Hill, 1975.

[27] GÄSSLER, J. and MASSEN, R., "A hierarchical classifier for non-coded photo-cassettes based on visual lable identification", Proceedings of the 6th International Conference on Robot Vision and Sensory Controls, June 1986, pp. 107-116.

[28] FU, K.S., "Syntactic Methods in Pattern Recognition", Academic Press, New York, 1974.

[29] **SHAFER, S.A. and KANADE, T.,** "Using shadows in finding surface orientation", Technical Report, Carnegie-Mellon University, Department of Computer Science, Pittsburgh, USA, January 1982.

[30] **BARROW, H.G. and TENENBAUM,** "Reconstructing smooth surfaces from partial, noisy information", ARPAIUS Workshop, L.S. Baumann (ed.) November 1979, pp. 76-86.

[31] **LOWE, D.G. and BINFORD, T.O.,** "The interpretation of geometric structure from image boundaries", ARPAIUS Workshop, L.S. Baumann (ed.), April 1981, pp. 39-46.

[32] **WITKIN, A.,** "Shape from contour", Technical report, AI-TR 589, MIT, USA, November 1980.

[33] **WALTZ, D.,** "Understanding line drawings of scenes with shadows", The Psychology of Computer Vision, P.H. Winston (ed.), McGraw-Hill, 1975, pp. 19-91.

[34] **O'GORMAN, F.,** "Light lines and shadows", Technical Report, School of Social Sciences, University of Sussex, December 1976.

[35] **QUAM, L. and HANNAH, M.J.,** "Stanford automated photogrammetry research", Technical Report,AIM-254, Stanford AI Lab., USA, November, 1974.

[36] **BALLARD, D.H. and BROWN, C.M.,** "Computer vision", Prentice-Hall Inc, New Jersey, 1982.

[37] BAJCSY, R. and LIBERMAN, L., "Texture gradient as a depth cue", Computer Graphics and Image Processing, Vol. 5, No. 1., March 1976, pp. 52-67.

[38] LU, S.Y. and FU, K.S., " A syntactic approach to texture analysis", Computer Graphics and Image Processing, Vol. 7, No. 3, June 1978, pp. 303-330.

[39] TAMURA, H., MORI, S. and YAMAWAKI,, "Textural features corresponding to visula perception", IEEE Trans. Vol. SMC-8, 1978, pp. 460-473.

[40] WESZKA, J.S., DYER, C.R. and ROSENFELD, A., "A comparative study of texture measures for terrain classification", IEEE Trans. Vol. SMC-6, 1976, pp. 269-285.

[41] TOMITA, F., "Description of textures by a structural analysis, "6IJCAI, 1979, pp. 884-889.

[42] DEGUCHI, K. and MORISHITA, I., "Texture characterisation and texture-based partitioning using two dimensional linear estimation", IEEE Trans, Vol. C-27, August 1978, pp. 739-747.

[43] HARALICK, R.M., SHANMUGAM, K. and DINSTEIN, I., "Texture features for image classification", IEEE Trans., Vol. SMC-3, November 1973, pp. 610-621.

[44] GALLOWAY, M.M., "Texture analysis using grey level run length", Computer Graphics and Image Processing, Vol. 4., No. 3, June 1975, pp. 172-179.

[45] HAYES, K.C., SHAH, A.N. and ROSENFELD, A., "Texture coarseness: Further experiments", IEEE Trans., Vol. SMC-4, September 1974, pp. 467-472.

[46] ROSENFELD, A., "A note on automatic detection of texture gradient", IEEE Trans., Vol. C-24, October 1975, pp. 988-991.

[47] MITCHELL, O.R., MYERS, C.R. and BOYNE, W., "A max-min measure for image texture analysis", IEEE Trans., Vol. C-26, April 1977, pp. 408-414.

[48] ROSENFELD, A. and THURSTON, M., "Edge and curve detection for visual scene analysis", IEEE Trans., Vol. C-20, May 1971, pp. 562-569.

[49] HEFFERNAN, P.B. and ROBB, R.A., "A new method for shaded surface display of biological and medical images" IEEE Trans., on medical imaging, Vol. MI-4, No. 1, March 1985, pp. 26-38.

[50] UDUPA, J.K., "Display of 3D information in discrete 3D scenes produced by computerised tomography", Proceedings of the IEEE, Vol. 71, No. 3, March 1983, pp. 420-431.

[51] HERMAN, G.T. and UDUPA, J.K., "Display of three-dimensional discrete surface", SPIE Vol, 283, 1981, pp. 90-97.

[52] ROSSOL, L., "Computer vision in industry", Robot Vision, A. Pugh (ed.), IFS (Publications) Ltd. U.K. 1983, pp. 11-42.

[53] WARD, M.R., ROSSOL, L. and HOLLAND, S.W., "CONSIGHT: An adaptive Robot with vision", Robotics Today, Summer 1979, pp. 26-32.

[54] ROSSOL, L., "Vision and adaptive robotics in general motors", Proceedings of the 1st International Conference on Robot Vision and Sensory Controls, Stratford-upon-Avon, U.K., IFS (Publications), April 1981, pp. 277-287.

[55] BAIRD, M.L., "SIGHT-I: a computer vision system for automated IC Chip manufacture", IEEE Trans., Vol. SMC-8, No. 2, February 1978, pp. 133-139.

[56] AYRES, R.A., BREWER, E.G. and HOLLAND, S.W., "Grid circle analyser computer-aided measurement of strain", SAE Trans., 790741, 1979, pp. 2630-2634.

[57] KANADE, T., THORPE, C. and WHITTAKER, W., "Autonomous Land Vehicle at CMU", Proc. of the 14th Annual Computer Science Conference, 1986 (CSC '86 Proceedings), pp. 71-80.

[58] WHITTAKER, W., "Terregator-Terrestrial Navigator", Technical Report, CMU Robotics Institute, 1985.

[59] MATSUSHIMA, T., SONOMOTO, I., and HARADA, T., "Automated high speed recognition of printed music, (WABOT-2 vision system)", Proceedings of '85 International Conference on Advanced Robots, Japan, September 1985, pp. 477-482.

[60] YACHIDA, M. and TSUJI, S., "Industrial computer vision in Japan", IEEE Computer, May 1980, Vol. 13, No. 5 pp. 50-63.

[61] HOLLINGUM, J., "Machine vision: The eye of automation", IFS (Publications) Ltd, London, 1984.

CHAPTER 3

[1] ICF SYSTEMS DIVISION, "Description of the Image Processing System developed in the Mechanical Engineering Department", Internal Report, Mechanical Engineering Department, University of Leeds, 1983.

[2] NEVATIA, R., "Machine Perception", Prentice Hall, 1982.

[3] DAVIS, L.S., "A survey of edge detection techniques", Computer Graphics and Image Processing (CGIP), Vol. 4, 1975, pp. 248-270.

[4] ROBERTS, L.G., "Machine perception of three dimensional solids ", Optical Electro-optical Processing of Information, J.T. Tippett (ed.), MIT Press, 1965, pp. 159-197.

[5] DUDA, R.O. and HART, P.E., "Pattern classification and scene analysis", Wiley, 1971, pp. 267-272.

[6] PREWITT, J.M.S., "Object enhancement and extraction", Picture processing and psychopictorics, B.S. Lipkin, A. Rosenfeld (eds.), Academic Press, 1970, pp. 75-149.

[7] ROSENFELD, A. and KAK, A.C., "Digital picture processing", Academic Press, 1976.

[8] KIRSCH, R., "Computer determination of the constituent structure of biological images", Computer Biomedical Research, Vol. 4, No. 3, 1971, pp. 315-328.

[9] ROBINSON, G.S., "Edge detection by compass gradient masks", Computer Graphics and Image Processing (CGIP), Vol. 6, 1977, pp. 492-501.

[10] FREI, W. and CHEN, C.C., "Fast boundary detection: A generalisation and a new algorithm", IEEE Trans., Vol. C-26, No. 10, October 1977, pp. 988-998.

CHAPTER 4

[1] FREEMAN, H., "Boundary encoding", Picture Processing and Psychopictoics, B.S. Lipkin and A. Rosenfeld (eds.), Academic Press, 1970, pp. 241-306.

[2] GEISLER, W., "A vision system for shape and position recognition of industrial parts", Proceedings of the 2nd International Conference on Robot Vision and Sensory Controls, IFS (Publications) Ltd, U.K., 1982, pp. 253-262.

[3] KITCHIN, P.W. and PUGH, A., "Processing of binary images", Computer Vision, A. Pugh (ed.), IFS (publications) Ltd, U.K., 1983, pp 21-42.

[4] DESSIMOZ, J.D., "Visual identification and location in a multi-object environment by contour tracking and curvature description", Proc. 10th IPA, Stuttgart 1978.

[5] BATCHELOR, B.G., HILL, D.A. and HODGSON, D.C., "Automated visual inspection", IFS (Publications) Ltd, The Netherlands, 1985.

CHAPTER 5

[1] FREEMAN, H., "Boundary encoding and processing", Picture Processing and Psychopictorics, Lipkin and Rosenfeld (eds.), Academic Press, New York 1970, pp. 241-306.

[2] PAVLIDIS, T. and ALI, F., "Computer recognition of handwritten numerals by polygonal approximations", IEEE Trans. Syst. Man Cybernet. SMC-6, 1975, pp. 610-614.

[3] ZAHN, C.T. and ROSKIES, R.Z., "Fourier descriptors for plane closed curves", IEEE Trans. Computers, Vol. C-21, 1972, pp.269-281.

[4] GRANLUND, G.H., "Fourier preprocessing for hand print character recognition", IEEE Trans. Computers, Vol. C-21, 1972, pp. 195-201.

[5] FREEMAN, H., "Computer processing of line-drawings images", Computer Survey No. 6, 1974, pp. 57-97.

[6] KUHL, F.P. and GIARDINA, C.R., "Elliptic Fourier features of a closed contour", Computer Graphics Image Processing, Vol. 18, No. 3, 1982, pp. 236-258.

[7] COOLEY, J.W. and TUKEY, J.W., "An algorithm for the machine calculation of complex Fourier series", Math Comp., Vol. 19, No. 90, April 1965, pp. 297-301.

CHAPTER 6

[1] BAJCSY, R. and LIBERMAN, L., "Texture gradient as a depth cue", Computer Graphics and Image Processing (CGIP), Vol. 5, No. 1, March 1976, pp. 52-67.

[2] BAJCSY, R., "Computer description of textured surfaces", 3IJCAI, 1973, pp. 572-579.

[3] BRADLEY, P.K. and DYER, R., "Edge separation and orientation texture measures", Proceedings of Computer Vision and Pattern Recognition Conference, IEEE Computer Society, San Francisco, June 1985, pp. 306-311.

[4] GALLOWAY, M.M., "Texture analysis using grey level run length", Computer Graphics and Image Processing, Vol. 4, June, 1975, pp. 172-179.

[5] MITCHELL, O.R., MYERS, C.R. and BOYNE, W., "A max-min measure for image texture analysis", IEEE Trans., Vol. C-26, April 1977, pp. 408-414.

[6] ROSENFELD, A. and THURSTON, M., "Edge and curve detection for visual scene analysis", IEEE Trans., Vol. C-20, May 1971, pp. 562-569.

[7] WESZKA, J.S., DYER, C.R. and ROSENFELD, A., "A comparative study of texture measures for terrain classification", IEEE Trans., Vol. SMC-6, April 1976, pp. 269-285.

[8] BEAUCHAMP, K. and YUEN, C., "Digital methods for signal analysis", George Allen and Unwin Ltd., 1979.

[9] BALLARD, D.H. and BROWN, C.M., "Computer Vision", Prentice-Hall, New Jersey, 1982.

[10] STREMLER, F.G., "Introduction to communication systems", second edition, Addison-Wesly, 1982.

CHAPTER 7

[1] McPHERSON, C.A., "Three-dimensional robot vision", Proceedings of the 3rd International Conference on Robot Vision and Sensory Controls (ROVSEC3), November 1983, Cambridge MA , pp. 116-126.

[2] MACKWORTH, A.K., "Interpreting pictures of polyhedral scenes", AI., Vol. 4, 1973, pp. 121-137.

[3] HORN, B.K.P., "Image intensity understanding", AI Memo 335, MIT Laboratory, August 1975.

[4] BLINN, J.F., "Models of light reflection for computer sythesised pictures", Computer Graphics, Vol. 11, No.2, 1977.

[5] PHONG, B.T., "Illumination for computer generated images", Comm, ACM 18, No. 6, June 1975, pp. 311-317.

[6] McPHERSON, C.A., TIO, J.B.K., HALL, E.L. and SADJADI, F.A., "Curved surface representation for image recognition", Proc. Pattern Recognition and Image Processing Conference, June 1982, pp. 370-378.

[7] COOK, R.L. and TORRANCE, K.E., "A reflectance model for computer graphics", Computer Graphics, Vol. 15, No. 3, August 1981, pp. 307-316.

[8] LANING, J.H. and MADDEN, S.J., "Capabilities of the SHAPES system for computer-aided Mechanical Design", Proc. of the First Annual Conference on Computer Graphics in CAD/CAM Systems, MIT, Cambridge, MA, April 1979, pp. 223-231.

A P P E N D I X A 1

Computer program listing for edge detection, using the Kirsch template matching method, and edge thinning.

```
(* ************************************************** *)
(*    This program detects the edges of an input image    *)
(*    using the KIRSCH operator . It also thins the edges *)
(*    detected by substituting the edge pixels by one     *)
(*    pixel value (the brightest edge pixel detected) .   *)
(* ************************************************** *)
PROGRAM EDGESEAR( INPUT,OUTPUT );
CONST MAX = 256;
      THRESH = 134;
TYPE LINE = ARRAY[1..MAX] OF CHAR;
     INT = INTEGER;
     POINTS = ARRAY[1..200] OF INTEGER;
     THIN = ARRAY[1..20000,1..3] OF INTEGER;
     NUMOFDOTS = INTEGER;
      EDGEOP = ARRAY[1..3,1..24] OF INTEGER;
      ACC = ARRAY[0..179,0..363] OF INTEGER;
      BLOCK=ARRAY[1..MAX,1..MAX] OF INTEGER;
      PICT=ARRAY[1..MAX,1..MAX] OF CHAR;
VAR RMIN,RMAX,EDGE,TTH,I,J,K,L,X,Y,R:INTEGER;
    N:NUMOFDOTS;
    STORE:THIN;
    A :EDGEOP;
    IX:LINE;
    IY2,RESULT:BLOCK;
    IY:ARRAY[1..MAX,1..MAX] OF CHAR;
(* The image is input by calling the external FORTRAN programs *)
(* developed for loading an image from the VAX  aux. discs.    *)
PROCEDURE FOPEN(%REF CH:INTEGER);FORTRAN;
PROCEDURE FCLOSE(%REF CH:INTEGER);FORTRAN;
(* CLOSES FILE ON CHANNEL 1 *)
PROCEDURE GETDAT(%REF CH,NUM:INTEGER;%REF IL:LINE);FORTRAN;
(* READS NUM VALUES FROM CHANNEL CH INTO ARRAY IL *)
PROCEDURE IPSAVE( IP:PICT );FORTRAN;
PROCEDURE FINDEDGES(I,J:INTEGER; VAR IY2:BLOCK );
VAR X,Y,LAR,H:INTEGER;
    SUM:ARRAY[1..8] OF INTEGER;
BEGIN
        FOR N:=1 TO 8 DO
        SUM[N]:=0;
        FOR N:=1 TO 8 DO
        FOR X:=1 TO 3 DO
        FOR Y:=1 TO 3 DO
        SUM[N]:=SUM[N] + IY2[I+X-1,J+Y-1]*A[ X,Y + (N-1)*3 ];
(* Multiplies array element by corresponding mask element and sums *)
        LAR:=SUM[1];
        FOR H:=2 TO 8 DO
        IF SUM[H] >= LAR THEN LAR:=SUM[H];
        IY2[I,J]:= LAR ;
        (* Max. response of applying all the masks *)
        IY2[I,J]:=( IY2[I,J] + 3825 )DIV 30;
        (* Normalise to range 0-255 *)
END;(* PROC Findedges *)

PROCEDURE THINEDGES(VAR IY2:BLOCK;STORE:THIN;N:NUMOFDOTS);
VAR X,Y,I,J,OPTX,OPTY,E:INTEGER;

BEGIN
   (* Initialise array *)
   FOR I:=1 TO 20000 DO
   FOR J:=1 TO 3 DO
   STORE[I,J]:=0;
   WRITELN('Thinning edges.....');WRITELN;
   N:=0;
   (* Scanning horizontally *)
   FOR Y:=6 TO 250 DO
   BEGIN
      X:=6;
      REPEAT
         X:=X+1;
         IF (IY2[X,Y]>=THRESH) AND (IY2[X-1,Y]<THRESH) THEN
         BEGIN
```

```
            (* START OF EDGE *)
            E:=X;
            OPTX:=X;
            N:=N+1;
            REPEAT
               X:=X+1;
               IF X>250 THEN X:=250;
            UNTIL (X=250) OR (IY2[X,Y]<THRESH);
            (* WIDTH OF EDGE = X-E *)
            FOR J:= (E+1) TO X DO
            IF IY2[J-1,Y]<IY2[J,Y] THEN OPTX:=J;
            STORE[N,1]:=OPTX;STORE[N,2]:=Y;STORE[N,3]:=IY2[OPTX,Y];
         END; (* IF *)
      UNTIL (X=250);
   END; (* FOR *)
   (* Scanning vertically *)
   FOR X:=6 TO 250 DO
   BEGIN
      Y:=6;
      REPEAT
         Y:=Y+1;
         IF (IY2[X,Y]>=THRESH) AND (IY2[X,Y-1]<THRESH) THEN
         BEGIN
            E:=Y;
            OPTY:=Y;
            N:=N+1;
            REPEAT
               Y:=Y+1;
               IF Y>250 THEN Y:=250;
            UNTIL (Y=250) OR (IY2[X,Y]<THRESH);
            FOR J:= (E+1) TO Y DO
            IF IY2[X,J-1]<IY2[X,J] THEN OPTY:=J;
            STORE[N,1]:=X;STORE[N,2]:=OPTY;STORE[N,3]:=IY2[X,OPTY];
         END; (* IF *)
      UNTIL (Y=250);
   END; (* FOR *)
      FOR I:=1 TO 256 DO
      FOR J:=1 TO 256 DO
      IY2[I,J]:=0;
      WRITELN('No. of edge pixels found =',N:5);WRITELN;
      FOR X:=1 TO N DO
      IY2[( STORE[X,1] ),( STORE[X,2] )]:=STORE[X,3];
      FOR X:=1 TO 256 DO
      FOR Y:=1 TO 256 DO
      IY2[X,Y]:=255;
      FOR X:=1 TO N DO
      IY2[ ( STORE[X,1] ),( STORE[X,2] ) ]:=0;
END; (* EDGETHIN *)

BEGIN
   (* INTIALISE VARIABLES *)
    A[1,1]:=-5; A[1,2]:=3; A[1,3]:=3;
    A[2,1]:=-5; A[2,2]:=0; A[2,3]:=3;
    A[3,1]:=-5; A[3,2]:=3; A[3,3]:=3;


A[1,4]:=3; A[1,5]:=3; A[1,6]:=3;
A[2,4]:=3; A[2,5]:=0; A[2,6]:=3;
A[3,4]:=-5; A[3,5]:=-5; A[3,6]:=-5;



A[1,7]:=3; A[1,8]:=3; A[1,9]:=-5;
A[2,7]:=3; A[2,8]:=0; A[2,9]:=-5;
A[3,7]:=3; A[3,8]:=3; A[3,9]:=-5;


A[1,10]:=-5; A[1,11]:=-5; A[1,12]:=-5;
A[2,10]:=3; A[2,11]:=0; A[2,12]:=3;
A[3,10]:=3; A[3,11]:=3; A[3,12]:=3;
```

```
A[1,13]:=-5;A[1,14]:=-5;A[1,15]:=3;
A[2,13]:=-5;A[2,14]:=0;A[2,15]:=3;
A[3,13]:=3;A[3,14]:=3;A[3,15]:=3;


A[1,16]:=3;A[1,17]:=3;A[1,18]:=3;
A[2,16]:=3;A[2,17]:=0;A[2,18]:=-5;
A[3,16]:=3;A[3,17]:=-5;A[3,18]:=-5;


A[1,19]:=3;A[1,20]:=3;A[1,21]:=3;
A[2,19]:=-5;A[2,20]:=0;A[2,21]:=3;
A[3,19]:=-5;A[3,20]:=-5;A[3,21]:=3;


A[1,22]:=3;A[1,23]:=-5;A[1,24]:=-5;
A[2,22]:=3;A[2,23]:=0;A[2,24]:=-5;
A[3,22]:=3;A[3,23]:=3;A[3,24]:=3;


I:=1;
L:=MAX;
FOPEN(I);
FOR J:=1 TO MAX DO
BEGIN
     GETDAT(I,L,IX);
     FOR K:=1 TO MAX DO IY[J,K]:=IX[K];
     (* STORES IMAGE IN ARRAY IY *)
END;
FCLOSE(I);
(* CONVERT CHAR ARRAY TO INTEGER ARRAY *)
   FOR I:=1 TO 256 DO
   FOR J:=1 TO 256 DO
   IY2[I,J]:=ORD( IY[I,J] );
   WRITELN;
   WRITELN('Processing picture using edge operator.....');WRITELN;
   FOR I:=1 TO 253 DO
   FOR J:=1 TO 253 DO
      FINDEDGES(I,J,IY2);
   THINEDGES(IY2,STORE,N);
   FOR Y:=1 TO 256 DO
   FOR X:=1 TO 256 DO
   IY[X,Y]:=CHR(IY2[Y,X] );
   IPSAVE(IY);
END.
```

A P P E N D I X A 2

Computer program listing for contour tracing, chain encoding, Fourier Descriptors development and similarity measurement.

```
(*    ************************************************    *)
(*    This program traces the contour of input shapes , chain *)
(*    encodes the boundary , estimates the orientation when   *)
(*    compared to another version of the shape , calculates   *)
(*    the Fourier Descriptors of the traced contour , using   *)
(*    the boundary pixel coordinates , and measures the simi- *)
(*    larity between two shapes being considers .             *)
(*    ************************************************    *)
PROGRAM SEARFOUR (INPUT,INFILE,FCHAIN,OUTPUT,OTFILE);
CONST MAX=256 ;
      PI=3.141593 ;
      %INCLUDE 'PASGKS_PAR/LIST'
TYPE VECTOR=ARRAY[1..200] OF REAL;
     INTVECTOR = ARRAY[1..13] OF INTEGER ;
      CHTABLE = PACKED ARRAY[1..20,1..12] OF CHAR ;
      VICTORY=ARRAY[1..10000] OF INTEGER;
      VIC=ARRAY[1..10000] OF REAL;
      MATRIX=ARRAY[1..MAX,1..MAX] OF CHAR;
      CHAINMAT=ARRAY[-1..1,-1..1] OF INTEGER;
      STRING20=PACKED ARRAY [1..20] OF CHAR;
      STRING=VARYING[40] OF CHAR;
      LINE=ARRAY[1..MAX] OF CHAR;
      %INCLUDE 'PASGKS_INC/LIST'
VAR   INFILE,FCHAIN,OTFILE  :TEXT;
      FILE1 : PACKED ARRAY[1..20] OF CHAR ;
      I,J,COUNT1,CA,CB,MK,Z,X,II,JJ,KK,LL,Y,RT,SHIFT,K,N,L,M,STARTER:INTEGER;
      XCG,YCG,RMIN,RMAX,CMIN,CMAX,XR,YR,COUNT2,MF,COUNT,COUNT3 :INTEGER;
      XMIN,XMAX,YMIN,YMAX,CE,DMIN,A,B,AS,BS,CS,ANG,ANGD,DSUM,RTHRS :REAL;
      CR,SS,AT,COA,SIA,TAA,DMAX,AM,AMMAX,CXTEMP,SXTEMP,CYTEMP,SYTEMP :REAL;
      XAV,YAV,XAVTEMP,YAVTEMP,EUCDISTMP,EUCTMP,EUCDIS     :REAL ;
      ANS  :CHAR;
      IX:LINE ;
      R,C,R1,C1,NR1,NRT,NC1,NCT,PX,PY,CHAIN     :VICTORY ;
      CEI,SEC,D,RXSUM,IXSUM,RYSUM,IYSUM  :VIC;
      XORD,YORD,RFINAL,SFINAL,TDRXY,TDIXY,ANGRADF,ANGDEGF,SFINAL2,RFINAL2:VIC;
      CHAINNUM:CHAINMAT;
      FILENAME:STRING20;
      LAB1: STRING;
      FILNM: STRING;
      PIXEL:ARRAY[1..MAX,1..MAX] OF CHAR;
      XO,DX,XSP,YO,DY,YSP :REAL ;
      IXW,IYW,IXD,IYD,IXF,IYF,TNR,HARMONICS,COUNTMIN :INTEGER ;
      XVEC,YVEC : ARRAY[1..10000] OF REAL ;
      ZVEC,WVEC : ARRAY[1..10000] OF REAL ;
(* ------------------------------------------------*)
%INCLUDE 'SYS$LIBRARY:TEKINT.PAS/LIST'
(* ------------------------------------------------ *)
(* The following Procedure traces the outer contour of a pattern *)
PROCEDURE BOUNDARYSEARCH( VAR RP,CP,CODE:VICTORY;
                          VAR CAP,CBP,MKP:INTEGER;
                          A:MATRIX;CODENUM:CHAINMAT;
                          VAR COUNTER:INTEGER);
VAR N,K,H :INTEGER;
(* RP[N] & CP[N] are coordinates of centre point,N no. of last pixel *)
BEGIN  (* Column search,CA always on left *)
N:=COUNTER;
H := CAP-1 ;
K := RP[N]-2 ;
REPEAT
     H := H+1 ;
     REPEAT
          K := K+1 ;
          IF (ORD(A[K,H])>(ORD(A[(K+1),H]))) THEN
           BEGIN
            IF (K<>RP[N]) OR (H<>CP[N]) THEN
             BEGIN
              IF (K<>RP[N-1]) OR (H<>CP[N-1]) THEN
               BEGIN RP[N+1]:=K;
               CP[N+1]:=H;
               MKP:=1;
               CODE[N+1]:=CODENUM[RP[N+1]-RP[N],CP[N+1]-CP[N]];
```

```
            COUNTER:=COUNTER+1;
            END;
          END;
        END
ELSE IF (ORD(A[(K+1),H])>(ORD(A[K,H]))) THEN
        BEGIN
       IF ((K+1)<>RP[N]) OR (H<>CP[N]) THEN
          BEGIN
           IF ((K+1)<>RP[N-1]) OR (H<>CP[N-1]) THEN
           BEGIN RP[N+1]:=K+1;
           CP[N+1]:=H;
           MKP :=1;
           COUNTER:=COUNTER+1;
           CODE[N+1]:=CODENUM[RP[N+1]-RP[N],CP[N+1]-CP[N]];
             END
            END;
          END;
      UNTIL (K = RP[N]) OR (MKP = 1) ;
         IF (MKP<>1) THEN
          K := RP[N]-2 ;
         UNTIL (H=CBP) OR (MKP=1);
(*-- Now Row search i.e. if no boundary found in Column search --*)
         IF(MKP<>1)THEN
          BEGIN
           K:=RP[N]-2;
           REPEAT
           K:=K+1;
           IF(ORD(A[K,CBP])>(ORD(A[K,CAP]))) THEN
            BEGIN
             IF (K<>RP[N]) OR (CBP<>CP[N]) THEN
              BEGIN
               IF (K<>RP[N-1]) OR (CBP<>CP[N-1]) THEN
                BEGIN
                 RP[N+1]:=K;  MKP:=1;  CP[N+1]:=CBP;
                 CODE[N+1]:=CODENUM[RP[N+1]-RP[N],CP[N+1]-CP[N]];
                 COUNTER:=COUNTER+1;
                END;
              END;
            END
           ELSE IF (ORD(A[K,CAP])>(ORD(A[K,CBP]))) THEN
            BEGIN
             IF (K<>RP[N]) OR (CAP<>CP[N]) THEN
              BEGIN
               IF (K<>RP[N-1]) OR (CAP<>CP[N-1]) THEN
                BEGIN
                 RP[N+1]:=K;  MKP:=1;  CP[N+1]:=CAP;
                 CODE[N+1]:=CODENUM[RP[N+1]-RP[N],CP[N+1]-CP[N]];
                 COUNTER:=COUNTER+1;
                END
              END;
            END;
          UNTIL (K = RP[N]+1) OR (MKP = 1) ;
       END;
END;
(* ------------------------------------------------------------ *)
(* This Procedure uses the developed DFT routine to evaluate the   *)
(* Fourier Coefficients, the inverse of Fourier analysis is also   *)
(* performed to get back to the original contour points .          *)
(* ------------------------------------------------------------ *)
PROCEDURE BNDFT(COUNTERS :INTEGER;
                VAR RSP,CSP :VICTORY;
                VAR RXSUMS,IXSUMS,RYSUMS,IYSUMS  :VIC ;
                VAR XORDS,YORDS     :VIC) ;
VAR N,M :INTEGER ;
    ALPHA :REAL ;
    RXSTEMP,IXSTEMP,RYSTEMP,IYSTEMP :VIC ;
BEGIN
L:=COUNT-2  ;
FOR N:= 1 TO L DO
   BEGIN
      R1[N] := RSP[N+2] ;
      C1[N] := CSP[N+2] ;
```

```
    END ;
FOR N:=1 TO (L-1) DO
    BEGIN
       RXSUMS[N] := 0 ;
       IXSUMS[N] := 0 ;
       RYSUMS[N] := 0 ;
       IYSUMS[N] := 0 ;
       ALPHA := N*2*PI/L ;
       FOR M := 1 TO (L-1) DO
         BEGIN
            CXTEMP := COS(ALPHA * M) ;
            SXTEMP := SIN(ALPHA * M) ;
            CYTEMP := COS(ALPHA * M) ;
            SYTEMP := SIN(ALPHA * M) ;
            RXSUMS[N] := RXSUMS[N] + R1[M] * CXTEMP ;
            IXSUMS[N] := IXSUMS[N] + R1[M] * SXTEMP ;
            RYSUMS[N] := RYSUMS[N] + C1[M] * CYTEMP ;
            IYSUMS[N] := IYSUMS[N] + C1[M] * SYTEMP ;
         END ;
       TDRXY[N] := (RXSUMS[N] + IYSUMS[N])/(L-1) ;
       TDIXY[N] := (RYSUMS[N] - IXSUMS[N])/(L-1) ;
       ANGRADF[N] := ARCTAN((TDIXY[N])/(TDRXY[N])) ;
       ANGDEGF[N] := (ANGRADF[N] *180)/PI ;
       RXSUMS[N] := RXSUMS[N]/(L-1) ;
       IXSUMS[N] := -IXSUMS[N]/(L-1) ;
       RYSUMS[N] := RYSUMS[N]/(L-1) ;
       IYSUMS[N] := -IYSUMS[N]/(L-1) ;
    END ;          (* End of implementing the Fourier Analysis *)
(* ************************************************************ *)
(* The following routine is to test the accuracy of the Fourier      *)
(* Descriptors , starting by finding the averages of the Boundary    *)
(* coordinates.The first few (6 - 10) descriptors are sufficient     *)
(* to obtain the original coordinates to reconstruct the boundaries  *)
(* ************************************************************ *)
XAVTEMP := 0 ;
YAVTEMP := 0 ;
FOR M:= 1 TO L-1 DO
   BEGIN
      XAVTEMP := XAVTEMP + R1[M] ;
      YAVTEMP := YAVTEMP + C1[M] ;
   END ;
XAV := XAVTEMP/(L-1) ;
YAV := YAVTEMP/(L-1) ;
WRITELN('X AV. is  ',XAV:10:2,'Y AV. is   ',YAV:10:2) ;
(* Now find the X & Y coordinates          ******    *)
WRITELN ;
WRITELN('  ENTER THE NUMBER OF HARMONICS  ') ;
READLN(HARMONICS) ;
MF := HARMONICS ;
FOR M:= 1 TO L DO
   BEGIN
      XORDS[M] := 0 ;
      YORDS[M] := 0 ;
      RXSTEMP[M] := 0 ;
      IXSTEMP[M] := 0 ;
      RYSTEMP[M] := 0 ;
      IYSTEMP[M] := 0 ;
      ALPHA := (M*2*PI)/L ;
      FOR N:= 1 TO MF DO
        BEGIN
           CXTEMP := COS(ALPHA * N) ;
           SXTEMP := SIN(ALPHA * N) ;
           RXSTEMP[M] := RXSTEMP[M] + RXSUMS[N] * CXTEMP ;
           IXSTEMP[M] := IXSTEMP[M] + IXSUMS[N] * SXTEMP ;
           RYSTEMP[M] := RYSTEMP[M] + RYSUMS[N] * CXTEMP ;
         IYSTEMP[M] := IYSTEMP[M] + IYSUMS[N] * SXTEMP ;
        END ;
      XORDS[M] := (XAV) + 2 * (RXSTEMP[M]) - 2 * (IXSTEMP[M]) ;
      YORDS[M] := (YAV) + 2 * (RYSTEMP[M]) - 2 * (IYSTEMP[M]) ;
   END  ;
END ;
(* ------------------------------------------------------------ *)
```

```
(* The images are input to the program using the developed FORTRAN *)
(* programs , called here as external fiels .                       *)
PROCEDURE FOPEN(%REF CH:INTEGER);FORTRAN;
PROCEDURE FCLOSE(%REF CH:INTEGER);FORTRAN;
PROCEDURE GETDAT(%REF CH,NUM:INTEGER;%REF IL:LINE);FORTRAN;
BEGIN (* MAIN PROGRAM *)
(* ------------------------------------------------------------- *)
(* Read in from FCHAIN.DAT Code elements (Freeman's Codes)       *)
(*   FCHAIN.DAT holds the octal codes correspondind to Freeman's *)
      OPEN (INFILE,'FCHAIN.DAT',OLD);
      RESET (INFILE);
      FOR I:=1 TO 3 DO
      BEGIN
          FOR J:=1 TO 3 DO
          READ(INFILE,CHAINNUM[I-2,J-2]);
          READLN(INFILE);
      END;
      CLOSE (INFILE);
(* Read FCHAIN.DAT completed  *)
(* The following routine prints the Freeman's code on the screen  *)
(*  to check if the codes have been read correctly                *)
      FOR I:=1 TO 3 DO
      BEGIN
          FOR J:=1 TO 3 DO
          WRITE(CHAINNUM[I-2,J-2]);
          WRITELN;
      END;
(* ------------------------------------------------------------- *)
(*                 Read in picture file(s) required              *)
(*              RT represents the no. of files read in.          *)
 RT:=1 ;
 WHILE RT<=2 DO
   BEGIN
       II:=1;
       LL:=MAX;
       FOPEN(II);
       FOR JJ:=1 TO MAX DO
       BEGIN
           GETDAT(II,LL,IX);
           FOR KK:=1 TO MAX DO PIXEL[JJ,KK]:=IX[KK];
       END;
       FCLOSE(II);
(* Picture file has now been input *)
(* ------------------------------------------------------------- *)
(* The main program for detecting the boundary and Chain_Encoding *)
(* the Contour starts from here.                                  *)

(*        Search for the first pixel on Boundary starts          *)
(*    The search starts from the 10th line and the 10th column   *)
(*  The corners of the frame ignored as invalid data ,Chapter 4  *)
 I:=10;
 COUNT:=2 ;
  REPEAT
      J:=10;
      WHILE(J<MAX) AND (ORD(PIXEL[I,J])=0) DO
          J:=J+1;
          I:=I+1;
   UNTIL ORD(PIXEL[(I-1),J])=1;
 R[COUNT]:=I-1;  C[COUNT]:=J;
(*                     End of first pixel search.                 *)
(*   First step in search,initially to left around boundary.       *)
   MK:=0;  CB:=C[COUNT];  CA:=CB-1;
   BOUNDARYSEARCH(R,C,CHAIN,CA,CB,MK,PIXEL,CHAINNUM,COUNT);
   IF MK<>1 THEN
    BEGIN CA:=CA+1;  CB:=CB+1;
        BOUNDARYSEARCH(R,C,CHAIN,CA,CB,MK,PIXEL,CHAINNUM,COUNT)
    END;
(*                End of first boundary search                    *)
(*  ---------------------------------------------------------     *)
(*       At this point it has already been decided which way      *)
(*  The search is to be continued in , i.e. left or right.        *)
REPEAT
```

```
(* Complete boundary search *)
     MK:=0;
     IF (R[COUNT]-R[COUNT-1])<=0 THEN
      BEGIN
           MK:=0; CA:=C[COUNT]; CB:=CA+1;
           BOUNDARYSEARCH(R,C,CHAIN,CA,CB,MK,PIXEL,CHAINNUM,COUNT)
      END
     ELSE BEGIN
             MK:=0; CB:=C[COUNT];CA:=CB-1;
             BOUNDARYSEARCH(R,C,CHAIN,CA,CB,MK,PIXEL,CHAINNUM,COUNT)
           END;
     IF ((R[COUNT]-R[COUNT-1]) <=0) AND (MK<>1) THEN
      BEGIN CB:=C[COUNT];CA:=CB-1;
          BOUNDARYSEARCH(R,C,CHAIN,CA,CB,MK,PIXEL,CHAINNUM,COUNT)
      END;
     IF (MK<>1) THEN
      BEGIN
          CA:=CA+1; CB:=CB+1;
           BOUNDARYSEARCH(R,C,CHAIN,CA,CB,MK,PIXEL,CHAINNUM,COUNT)
      END;
   UNTIL (C[COUNT]=C[2])AND(R[COUNT]=R[2]);
   WRITELN('***SEARCH COMPLETED***':40) ;
(* ------------------------------------------------------------------ *)
(*  The following section calculates the curvature elements, by using   *)
(* the Table developed in Chapter 4, and plote the developed series.    *)
 XMIN:=3 ; XMAX:=COUNT ;
 YMIN:=-180 ; YMAX:=180 ;
 GRAT(XMIN,XMAX,YMIN,YMAX);
 XLABEL(34,25,'CONTOUR POINTS');
 YLABEL(12, 4,'CONTOUR ELEMENTS');
 PLOT(XMIN,YMIN,0);
 FOR X:=3 TO COUNT-1 DO
   BEGIN
       Y := ABS (CHAIN[X+1] - CHAIN[X]) ;
       CASE Y OF
       0 :                    CE:=0 ;
       1 :                    CE:=135 ;
       2 :                    CE:=90 ;
       3 :                    CE:=45 ;
       4 :                    CE:=180 ;
       5 :                    CE:=45 ;
       6 :                    CE:=90 ;
       7 :                    CE:=135 ;
       END (* CASE *) ;
PLOT(X,CE,1);
IF RT=1 THEN
    BEGIN
       CEI[X]:=CE ;
       NR1[X]:=R[X];
       NC1[X]:=C[X];
       COUNT1:=COUNT;
    END
  ELSE BEGIN
          SEC[X]:=CE ;
          NRT[X]:=R[X];
          NCT[X]:=C[X];
          COUNT2:=COUNT;
        END;
   END;
 DEND;
(* By this point the contour points have been stored in separate  *)
(* arrays ,  and the curvature elements also stored and plotted   *)
WRITELN('* COUNT IS ... *',COUNT);
  RT:=RT+1;
  END (* WHILE *);
(*  By this point two shapes have been contour traced , and the curvature*)
(* elements are also developed for the two shapes.The following section  *)
(* developes the correlelograms to compare the shapes for recognition.   *)
 FOR J:=1 TO COUNT DO
    D[J]:=0 ;
    FOR J:=0 TO COUNT-3 DO
     BEGIN
```

```
        FOR I:=5 TO COUNT-2 DO
         BEGIN
            K:=I+J ;
            IF (K)>(COUNT-2) THEN BEGIN
             K:=K-(COUNT-2);
             D[J+1]:=D[J+1]+ABS(CEI[I]-SEC[K]);
             END
            ELSE D[J+1]:=D[J+1]+ABS(CEI[I]-SEC[I+J]);
         END;
     END;
DMIN:=D[1];
SHIFT:=0;
FOR J:=2 TO COUNT-2 DO
  BEGIN
   IF (D[J])<(DMIN) THEN
      BEGIN
         DMIN:=D[J];
         SHIFT:=J;
      END;
   END;
  DMAX:=D[1] ;
  FOR J:=2 TO COUNT-2 DO
  BEGIN
      IF (D[J])>(DMAX) THEN
      DMAX:=D[J] ;
  END;
(* ********************************************************** *)
(* The following section plots the correllelograms between the *)
(* two contour curvature element series developped .          *)
(* ********************************************************** *)
XMIN := 0 ;
XMAX := COUNT1 ;
YMIN := 0 ;
YMAX := DMAX + (DMAX/10) ;
WRITELN ('Picture file required ? y/n          ') ;
READLN (ANS) ;
GKS_OPKS (6) ;
GKS_OPWK (1,0,201) ;
GKS_ACWK (1) ;
(* THE FOLLOWING THREE COMMANDS CLEAR THE SCREEN *)
WRITE (CHR(29)) ;
WRITELN (CHR(27),CHR(12)) ;
GKS_CLRWK (1,GALWAY) ;
IF (ANS = 'Y') OR (ANS = 'y') THEN
BEGIN
     WRITELN ('INPUT PICTURE FILENAME') ;
     READLN (FILENAME) ;
     GKM_OPEN (12,FILENAME,TRUE) ;
     GKS_OPWK (2,12,50) ;
     GKS_ACWK (2) ;
END ;
(* THE FOLLOWING INSTRUCTION IS TO SET THE VIEW PORT *)
GKS_SVP (1,0,1,0,1) ;
TEK_SET (XMIN,XMAX,XO,DX,IXF,XSP,IXD) ;
TEK_SET (YMIN,YMAX,YO,DY,IYF,YSP,IYD) ;
IXW := 3 ;
IYW := 4 ;
IXD := 0 ;
IYD := 0 ;
XO := 0 ;
TNR := 1 ;
TEK_AXIS(TNR,XO,YO,XMIN,XMAX,YMIN,YMAX,XSP,YSP,IXW,IXD,IYW,IYD,DX,IXF,DY,IYF);
FOR J := 1 TO COUNT1 DO
BEGIN
      XVEC[J] := J ;
      YVEC[J] := D[J] ;
      ZVEC[J] := J ;
      WVEC[J] := D[J] ;
END ;
COUNT3 := COUNT1 - 25 ;
GKS_SPLI (5) ;
GKS_PL (COUNT3,XVEC,YVEC) ;
```

```
TEK_YLABEL (-60,500,'CORRELATED VALUES') ;
TEK_XLABEL (150,-500,'CONTOUR PIXELS') ;
GKS_UWK (1,GPERFO) ;
IF (ANS = 'Y') OR (ANS = 'y') THEN
BEGIN
       GKS_DAWK (2) ;
       GKS_CLWK (2) ;
       GKM_CLOSE (12) ;
END ;
GKS_DAWK (1) ;
GKS_CLWK (1) ;
GKS_CLKS ;
(* Calculate AMmax .   To be used in rejection decision    *)
DSUM:=0.0 ;
FOR X:=5 TO COUNT-2 DO
  BEGIN
      AM:=(0.5)*(ABS(CEI[X]-CEI[X+1])) ;
      DSUM:=AM+DSUM ;
  END;
RTHRS := DMAX ;
WRITELN('* DMIN IS   ......        *',DMIN);
WRITELN('*  DSUM is    ......      *',DSUM);
WRITELN('*  RTHRS is     .....      *',RTHRS);
WRITELN('* SHIFT IS *',SHIFT:40);
(* Calculating the angle of rotation,by first obtaining the     *)
(* centre of the area of the patterns .                         *)
RMIN:=NR1[6] ;
RMAX:=NR1[6] ;
FOR I:=7 TO COUNT1-2 DO
   IF NR1[I]>RMAX THEN
    RMAX:=NR1[I];
 CMIN:=NC1[6];
 FOR I:=7 TO COUNT1-2 DO
    IF NC1[I]<CMIN THEN
     CMIN:=NC1[I];
 CMAX:=NC1[6];
 FOR I:=7 TO COUNT1-2 DO
    IF NC1[I]>CMAX THEN
     CMAX:=NC1[I];
Z:=0;
FOR N:=1 TO COUNT1 DO
BEGIN
   PX[N]:=0 ; PY[N]:=0 ;
END ;
FOR N:=6 TO COUNT1-1 DO
BEGIN
   Z:=Z+1;
   PX[N+1]:=NR1[N]+1+PX[N];
   PY[N+1]:=NC1[N]+1+PY[N];
END;
XCG:=TRUNC((PX[COUNT1])/Z) ; YCG:=TRUNC((PY[COUNT1])/Z);
 WRITELN(' X of C of A IS .. ',XCG:40);
 WRITELN(' Y of C of A IS .. ',YCG:40);
(* CALCULATE THE ANGLE OF ROTATION BASED ON THE COSINE RULE *)
WRITELN(' INPUT REF. POINT CO-ORDINATES  (X,Y) ');
READLN (XR) ;
READLN (YR) ;
 AS:=SQR(NR1[XR]-XCG)+SQR(NC1[YR]-YCG) ;
 A:=SQRT(AS) ;
 BS:=SQR(NRT[XR+SHIFT]-XCG)+SQR(NCT[YR+SHIFT]-YCG) ;
 B:=SQRT(BS) ;
 CS:=SQR(NRT[XR+SHIFT]-NR1[XR])+SQR(NCT[YR+SHIFT]-NC1[YR]) ;
 CR:=SQRT(CS) ;
 COA:=(AS+BS-CS)/(2*A*B) ;
 SS:=(A+B+CR)/2 ;
 AT:=SQRT(SS*(SS-A)*(SS-B)*(SS-CR)) ;
 SIA:=(2*AT)/(A*B) ;
 TAA:=SIA/COA ;
 ANG:=ARCTAN(TAA) ;
 ANGD:=(ANG*180)/PI ;
 WRITELN('XR IS ...',XR:40);
 WRITELN('YR IS ...',YR:40);
```

```
 WRITELN('NR1[XR] IS ... ',NR1[XR]:40);
 WRITELN('NC1[YR] IS ..  ',NC1[YR]:40);
 WRITELN('NRT[XR+SHIFT] IS .. ',NRT[XR+SHIFT]:40) ;
 WRITELN('NCT[YR+SHIFT] IS .. ',NCT[YR+SHIFT]:40) ;
 WRITELN('THE ROTATION IS ... ',ANGD:50:2);
(* ************************************************** *)
(* Call thr DFT routine to obtain the Forier Descriptors      *)
(* and also the derived contour points, no. of harmonics to   *)
(* be used is to be inputed by the user.                      *)
(* ************************************************** *)
BNDFT(COUNT1,NR1,NC1,RXSUM,IXSUM,RYSUM,IYSUM,XORD,YORD);

(* Use the derived coordinates to re_construct the boundary   *)
GRAT (1,256,1,256) ;
PLOT (1,1,0) ;
PLOT (XORD[1],YORD[1],0) ;
FOR N := 2 TO L DO
PLOT (XORD[N],YORD[N],1) ;
DEND ;
(* This section is to derive the Descriptors which are not shift *)
(*             or rotation variant                               *)
FOR N := 1 TO (L-1) DO
  RFINAL[N] := 0 ;
FOR N:= 1 TO (L-1) DO
    BEGIN
        RFINAL[N] := SQRT((SQR(SQRT(SQR(RXSUM[N]) + SQR(IXSUM[N])))) +
                     (SQR(SQRT(SQR(RYSUM[N]) + SQR(IYSUM[N]))))) ;
        SFINAL[N] := RFINAL[N] / (RFINAL[1]) ;
    END ;
WRITELN(' Enter file name for ALL invariant descriptors ') ;
READLN (FILE1) ;
OPEN (OTFILE,FILE1,HISTORY := NEW) ;
REWRITE (OTFILE) ;
FOR N:= 1 TO (L-1) DO
WRITELN (OTFILE,RFINAL[N]:15:2,SFINAL[N]:15:2) ;
CLOSE (OTFILE) ;
(*  ************************************************  *)
(*  Perform the same Fourier routine on the second shape   *)
(*  ************************************************  *)

BNDFT(COUNT2,NRT,NCT,RXSUM,IXSUM,RYSUM,IYSUM,XORD,YORD);

(* ************************************************    *)
(* Create new output files for derived co_ordinates   **    *)
(* ************************************************    *)
WRITELN (' ENTER COORDINATES FILE NAME') ;
READLN (FILE1) ;
OPEN (OTFILE,FILE1,HISTORY := NEW) ;
REWRITE (OTFILE) ;
FOR N:= 1 TO 20 DO
WRITELN (OTFILE,XORD[N]:15:3,YORD[N]:15:3) ;
CLOSE (OTFILE) ;
(* Use the derived coordinates to re_construct the boundary **    *)
GRAT (1,256,1,256) ;
PLOT (1,1,0) ;
PLOT (XORD[1],YORD[1],0) ;
FOR N := 2 TO L DO
PLOT (XORD[N],YORD[N],1) ;
DEND ;
(* This section is to derive the Descriptors which are not shift *)
(*             or rotation variant                               *)
FOR N := 1 TO (L-1) DO
  RFINAL[N] := 0 ;
FOR N:= 1 TO (L-1) DO
    BEGIN
       RFINAL2[N] := SQRT((SQR(SQRT(SQR(RXSUM[N]) + SQR(IXSUM[N])))) +
                     (SQR(SQRT(SQR(RYSUM[N]) + SQR(IYSUM[N]))))) ;
       SFINAL2[N] := RFINAL2[N] / (RFINAL2[1]) ;
    END ;
WRITELN(' Enter file name for ALL invariant descriptors ') ;
READLN (FILE1) ;
OPEN (OTFILE,FILE1,HISTORY := NEW) ;
```

```
REWRITE (OTFILE) ;
FOR N:= 1 TO (L-1) DO
WRITELN (OTFILE,RFINAL2[N]:15:2,SFINAL2[N]:15:2) ;
CLOSE (OTFILE) ;
(* FIND THE MIN. DISTANCE BETWEEN THE TWO ALL-INVARIANT COEFFICIENTS *)
(*          USING EUCLIDEAN DISTANCE MEASUREMENT.                    *)
EUCDISTMP := 0 ;
FOR N:= 1 TO 15 DO
    BEGIN
       EUCTMP := SQR(ABS(SFINAL[N] - SFINAL2[N])) ;
       EUCDISTMP := EUCDISTMP + EUCTMP ;
    END ;
EUCDIS := SQRT(EUCDISTMP) ;
WRITELN ('EUCDIS IS .....',EUCDIS:10:2) ;
END.
```