AUTOMATED VISION FOR IDENTIFICATION OF SILHOUETTE SHAPES WITH AN APPLICATION TO INDUSTRIAL AUTOMATION

CHRISTOS KOULOPOULOS, B.Sc., M.Sc.

Thesis submitted for the Degree of Doctor of Philosophy

The City University, London
Department of Systems Science

January, 1982

CONTENTS

Page

CHAPTER 4: PROBLEM DEFINITION AND METHODOLOGY

CHAPTER 5: SIMULATION STUDY

Page

CHAPTER 7: PROCESSING ALGORITHMS

CHAPTER 8: EXPERIMENTS AND EVALUATION

Page

CHAPTER 9: OVERALL CONCLUSIONS AND RECOMMENDATIONS

APPENDIX

ACKNOWLEDGEMENTS

The financial support of the British United Shoe Machinery (BUSM) Company, which sponsored the research work reported in this thesis, has been much appreciated by the author, who would like to thank particularly [redacted].

The help and guidance, and in particular the negotiating contacts between The City University and the above Company, of [redacted], who was acting as supervisor, were also much appreciated.

The author would like to thank sincerely Professor L. Finkelstein, who supervised the work in general, for his many valuable criticisms, suggestions and corrections, and for the encouragement and motivation he offered continuously, for which the author is deeply indebted.

The practical suggestions of the senior lecturer, [redacted], in connection with the improvement of the drive mechanism of the scanning table, were very much appreciated.

The author is grateful to [redacted] for undertaking the typing of this thesis.

To his best friends, [redacted], who gave unlimited moral support and encouragement, the author is very grateful indeed.

Finally, the author would like to thank all his family for the unconditional support and love which they offer him continuously, and particularly his mother who made him fortunate enough to have her with him and who kept the kitchen going through the long nights during the writing of this thesis.

ABSTRACT

The automated interpretation of images from the products of a wide range of industries is a problem of enormous interest, and in particular the machine classification and identification of such two-dimensional images.

To achieve this it is necessary to simulate some of the activities of the human eye-brain mechanism so that a system or machine may be able to recognise patterns or shapes automatically.

The work reported in this thesis is concerned with the development of processing methods for examining silhouette images, with a view to exploiting the methods in the automation of industrial on-line inspection.

Other methods are available for recognising shapes, but their low accuracy, high cost to be constructed and their slow performance make them unacceptable for effective industrial applications.

Recent developments and availability of inexpensive and versatile digital processors, and of high resolution image sensors in the form of self-scanned photodiode arrays, have made automation of this kind of inspection possible.

A simulation experiment to evaluate particular methods developed during the present research work of defining features for silhouettes was carried out. Successful operation of automatic search and learning techniques were also demonstrated. A special interface was designed and constructed incorporating hardware boundary detecting, which speeds the scanning operation of the developed scanning rig from over 25 minutes (as originally) to only 4 seconds per sample under test on average.

A series of experiments and evaluation was performed in which the test scanning rig, which was finally developed and constructed and also interfaced to a minicomputer, was trained to recognise automatically on-line real patterns.

Features of an unknown pattern can be compared with the features of a set of known patterns, named as classes, in order to be identified if they belong to any of these classes. With the "minimum distance"

concept, only as many features as are necessary to separate all the classes can be used to identify an unknown pattern.

The system can distinguish between shapes which are larger or smaller by about 4% in size. Also, it distinguishes between left- and right-hand versions of a pattern and at several different orientations.

Further, the system can be evaluated with hundreds of different shapes, something which makes it industrially acceptable.

The above research work was supported by an industrial contract until its completion. Patent applications have been made in connection with several aspects of the present work.

CHAPTER 1: INTRODUCTION

1.0 GENERAL CONSIDERATIONS

The present research work was undertaken with the object of developing and constructing a trainable pattern classififer, which is required to learn and then identify sheet material workpieces to be used in an industrial assembly operation.

Many industries require fast, accurate and cheap ways of automatic inspection for their products. The automated interpretation of the silhouette images of these products, and in particular the machine classification and identification of such two-dimensional shapes during an inspection operation, are problems of great importance.

Despite the fact that the number of people working in this field has increased considerably over the last few years, it is essential to notice that the problem of achieving all the above three requirements simultaneously is not an easy task. What makes it more difficult is that advanced technology, which may be used to help solve such a problem, is still expensive.

Since the work reported in this thesis was sponsored by an industrial company, it was essential to keep the resulting system inside certain cost limits required by the company. This undoubtedly offered an interesting challenge during the development of the present work.

Automated vision research has mainly concentrated upon the visual perception of light intensity. Usually an analog to digital (A/D) converter transforms data from a television (T.V.) camera, or other visual sensor (e.g. photodiode array), into numbers in the memory of a computer. The major research task is then to develop algorithms that can process those numbers and figure out the object under observation.

To achieve this it is also helpful to develop some of the processes to be performed for automatic inspection in computer simulation before any hardware implementation.

However, the difficulty of simulating the environment and physical characteristics of a real machine accurately, limits the usefulness of simulation. A real machine must be built.

Many methods are available for recognising shapes, but their main disadvantages are the high cost of constructing them or their low accuracy, but mainly the slow speed of their performance. During the development of the present work these factors were considered carefully in order to achieve an acceptable balance and make its use commercially feasible.

1.1 Computer Vision in Industrial Automation

The basic meaning of sight is to perceive by the eye. Several physical properties of any specified point in the field of view can be sensed by seeing: light intensity, colour, distance and motion. Of these, the first - light intensity - is the most important. After all, a black-and-white photograph is generally "seen" as an adequate representation of the scene it depicts, even though it really conveys information only about the relative brightness of points in the scene. Certainly, such a picture has no colour, no depth and no motion. Most of the research thus far conducted on the automation of vision deals with the process of seeing light intensities.

One might think that a computer cannot possibly see, because it does not have physical hardware capable of measuring light intensities. This claim is obviously true if we define computer in its narrowest sense, to be merely a central processing unit with memory but without peripheral devices.

A computer can "see" if it contains all the following: a T.V. camera or other visual sensor, a central processor and memory, appropriate interface for transmitting visual data into memory, and appropriate software for interpreting the data the memory receives.

Although T.V. cameras have been widely used for research into computer vision, a group of photodiodes, for example, can also be used. Whatever visual sensor we select, A/D converters can transform the brightness measured by the sensors into numbers in the memory of a computer. The key problem of whether a computer can see is, therefore, reduced to one of whether software can be devised that makes sense, "attains awareness or understanding", out of the numbers derived from the visual sensors.

When we move to actual factory operations, we should seriously

consider the problem of quality control, perhaps the biggest headache for most manufacturers and distributors. Suppose a computer is programmed to be aware of production, inspection or distribution. Equipped with the pattern classification, perceptions, and decision-making abilities, that computer could look at a car, say, and see whether the hub-caps and radio antenna are missing, or it could check a shipment to see that the proper goods are enclosed, or it could watch the bottles to monitor the clarity of the beer, and it would never get bored, take coffee breaks, or be distracted by people walking past.

Even if the automatic factory is still a dream, the automatic inspection and assembly station is not. Prototype versions exist in laboratories today and are rapidly being reduced in cost. Within a few years we should see computer-operated systems capable of replacing men in a great variety of dull, repetitive, dangerous, or otherwise unpleasant factory occupations.

1.2 The Necessity of Identifying Silhouette Shapes

Silhouette images are very important in intelligent automation, and particularly in automatic inspection, because most objects, the inspection of which is desired, are adequately defined by an outline shape.

A silhouette shape can be identified by any one of the following ways:

(1) A two-dimensional network of samples or cells, each of which may have the value "0" or "1".

(2) A list of co-ordinates specifying the location of the cells on the periphery of the silhouette object.

(3) A chain encoding of the boundary, in which the position of each boundary cell is defined by specifying its relationship with respect to its immediate neighbours (e.g. Freeman 3 bits/cell chain code).

The boundary of a real object must always be closed.

1.3 Objective

The objective of the work presented in this thesis is to produce an instrument for the automatic on-line inspection and identification of flat opaque components which are industrially manufactured.

There are three assumptions implicit in this account:

(a) That we are dealing with discrete objects, i.e. objects which do not overlap.

(b) That the objects are to be described one at a time; and

(c) That the objects are considered to be two-dimensional shapes, and treated, therefore, as silhouette shapes.

The shapes may contain small (less than 3 mm diameter) holes, which must be ignored in the sorting process. The largest sample will fit inside a rectangle of about 200 mm by 450 mm, the smallest within a square of side 25 mm.

The machine must distinguish between components from adjacent sizes, which are roughly similar shapes differing in size by 6 mm on some dimensions, though not necessarily on all.

The instrument must process the patterns at a high speed, and, if possible, of the order of 4 patterns per second.

The instrument is also required to "learn" new patterns as and when required, without the assistance of a skilled programmer.

Finally, by not exceeding the storage capacity of the general computer used during the on-line operation, the resulting system should be able to "remember", i.e. to be able to identify, as many different shapes as possible, when presented on a moving line at random orientation.

1.4 Proposed Approach

Many methods are available for recognising objects, and it was necessary to select one which could meet our requirements. The selection was made chiefly from theory and applied to the problem of identifying silhouette shapes. To test that the method was worth considering for further development, a simulation experiment was introduced.

The simulation had to show that the measures to be used for

distinguishing silhouette patterns could be determined accurately enough.

Two simple and mathematically defined shapes were chosen for the simulation, an ellipse and a cardioid. They were generated by a computer program and plotted on a microfilm. To be processed by the computer, the above shapes were imposed on to a binary grid of cells.

The size, orientation and form (of the ellipse by its eccentricity) were varied systematically. The accuracy with which parameters could be determined, such as area, perimeter, orientation and radii from the centroid of the shape to its boundary, was measured.

The above area and radii lengths could be measured accurately enough, something which made their use as characterising features practically feasible, but perimeter was not a stable parameter. It was necessary, however, to determine the boundary in order to extract the other parameters. It was found also that the boundary points should be determined by looking in two perpendicular directions otherwise spaces would appear in the outlines, which would cause errors.

After the simulation and its assessment, the involvement in experimenting and evaluating with real patterns at an on-line inspection operation should be the next stage.

The instrument would take the following form: The components would be placed on a transparent table and illuminated from underneath. The table would move by a stepping motor driven either by computer or manually. The outlines of the components would be sensed by some form of opto-electronic sensing device. A T.V. camera, such as a vidicon, could be used, but a line of array of photodiodes would probably be better since precise dimensional measurement would then be much simpler. The shapes would be illuminated from the side opposite the camera; the component edges would be detected simply by noting the light level at each point in the field of view; whenever this crossed a non-critical threshold in passing two such consecutive points, an edge point would be indicated. The object boundary would therefore be constructed by storing all such points.

The instrument would be required to operate basically in two modes: a "learn" mode and an "inspect" mode. In the learn mode samples of all patterns to be identified would be presented in turn to the machine and features would be extracted to describe each shape economically, and a

combination of features selected would be sufficient to separate the patterns as different classes with the minimum of computation . The tolerances involved in each feature measure would be supplied to the machine to assist in the selection of the feature set per pattern. The feature sets of all pattern classes would remain stored in the machine, so that the new patterns could be added subsequently to the inspection list without the need for repeated presentation of shapes already scanned and stored.

In the inspect mode, the instrument would identify patterns passing through the camera field, one at a time, by first extracting feature measures and comparing these with the stored sets of features until a fit was found. The method of search would be optimised so as to minimise the mean time taken to identify a pattern under test.

The only effective way of speeding up the overall performance of the system is to perform boundary location and possibly other operations in hardware, which could extract the boundary points, as well as indicate their nature (i.e. "up", "down", "left", or "right" points around the boundary), which would give a very desirable scanning speed to the system.

Human interaction in the feature selection and learning phase, other than to supply the name of pattern classes, is not acceptable in a factory environment, although it is essential for experimental research.

1.5 Conclusions

There are many reasons for looking towards silhouette image processing in industrial applications. Mainly these are:

(i) As an essential component in flexible automation,
(ii) Improvement in the reliability of inspection process, and
(iii) Overall reduction in cost, improvement in quality control, productivity and profitability.

Methods exist for precise manual inspection of manufactured components and for measuring them. These methods are generally slow, work only on samples of product, and require accurate position of parts in highly specific jigs.

By contrast, image methods can check parts for "shape" in addition

to dimension, detect, assess and identify them, at rates faster than operatives, without fatigue. True, 100% inspection is now desirable and also possible.

CHAPTER 2: IMAGE DATA ACQUISITION

2.0 INTRODUCTION

The present chapter is a survey of the principles of image data acquisition and of the principles of equipment available, in order to explain the background of the choice of the equipment for image acquisition in the present system.

Data acquisition of an image is the process of the formation of the image and its conversion into a signal for further processing.

A survey of candidate image data gathering devices, including television cameras, photodiode arrays and laser scanners, was conducted at an early stage of this work and is reported briefly in this chapter, so that a comparison of different types of image acquisition systems, which could be suitable for an industrial implementation of the inspection system, the development of which was undertaken in this work, may be obtained.

Whatever methods of processing are developed, they must be implementable at a cost which is tolerable industrially, and must be fast enough in operation in order to be compatible with an automated production line.

Another factor to be considered is the noise, since the descriptive information extracted from an image is invariably accompanied by an unwanted noise signal, the influence of which on subsequent processing operation must be minimised.

Further, the problem of distortion, resolution, depth of focus, spectral response and even bandwidth should be considered, and make the final choice of optical scanner as suitable as possible for the specific application in which it is to be used.

2.1 Image Acquisition System

The image acquisition system obtains visual data from the part under test. The acquisition method, data content and form, and scanning schemes are determined by the inspection problem. Generally, problems fall into two broad overlapping categories: dimensional and surface feature measurements.

Strictly dimensional problems focus on measuring the geometry of the part with little regard for surface condition, while surface feature problems concentrate on the condition, texture, colour, etc., of the surface with little regard for location. Coupled problems, such as measuring the dimension of a coloured area or the size and location of surface defects, fall in the largely unexplored middle ground.

The implementation of dimensional measurement systems is generally straightforward. Figure 2.1 below shows schematically how an image acquisition system constitutes a sub-system of a complete inspection system.

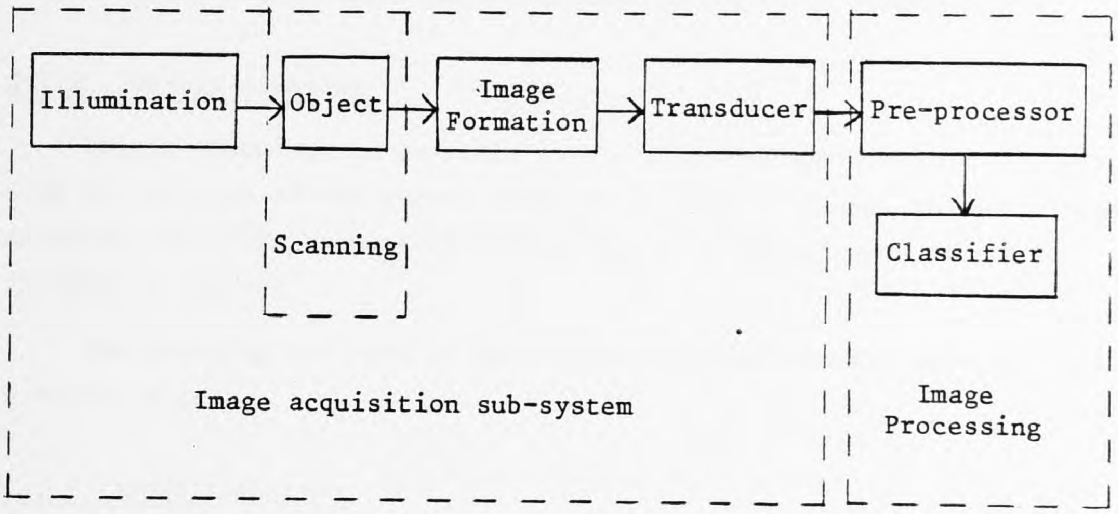


Figure 2.1: Complete inspection system

2.1.1 Illumination

The numerous electro-optic techniques employ various forms of light sources. For example, coherent sources of light such as He - Ne lasers, as well as simple tungsten lamps, are used.

Basically, incident light may be reflected, scattered or absorbed. A perfect reflector would reflect all the incident light without any absorption or scattering, and a perfect absorber would absorb all the light without any reflection or scattering. However, for most practical considerations there will be a combination of all modes and selective wavelengths will be absorbed, scattered or reflected, depending on the nature of the surface.

With a perfect surface, the reflected light would have the same angle of light incidence.

In practice, however, the reflector would not be perfectly smooth and the light is scattered to varying degrees, dependent on the surface roughness.

The main reflected beam is reflected in the "specular" direction and the scattered beams are said to be "off-specular".

The light source may be either monochromatic (a single wavelength only present) or polychromatic (two or more wavelengths, or bands, present).

Some objects it is necessary to inspect by illuminating from above, others to be illuminated from underneath during their scanning.

2.1.2 Object scanning

Object scanning, in general, is the repeated and controlled traversing of any area of the object under test, or its volume, with a moving detector, in order to measure some quantity or detect some defect on the object surface.

The scanning can also be performed with a stationary detector and a moving object.

2.1.3 Image formation

The principal elements in image formation are represented in a schematic form in Figure 2.2. The "box" is a device that is capable of acting upon a radiant energy component of the object under test. This does not rule out the possibility that the "box" itself emits some type of energy and then acts upon the interaction of this energy with the object.

Real image formation systems possess "neighbourhood" properties in the processes by which images are formed. This means that a point (x, y) in the image plane (see Figure 2.2) is the image not only of a corresponding point (p, q) in the object plane, but also may be a function of radiant energy contributions in a (possibly infinite) neighbourhood of p, q. The image, f(p, q), is the accumulation of those infinitesimal contributions.

The techniques employing optics to form the image normally require fairly sophisticated optics, and often involve specially-designed difraction gratings or sensor arrays (Kasdan and Mead, 1978).

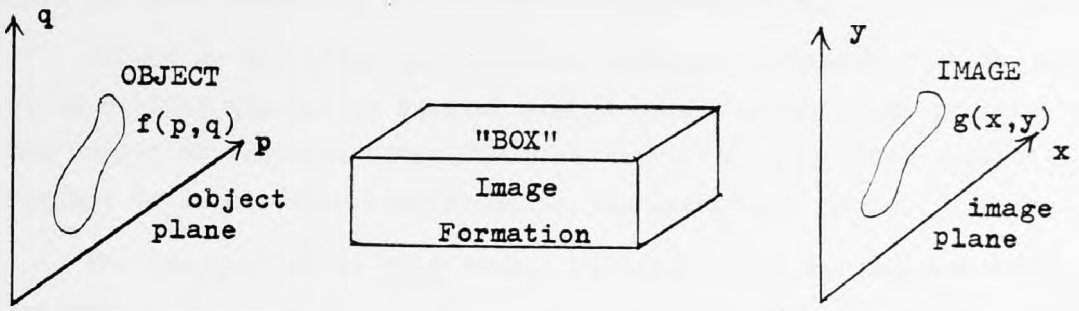


Figure 2.2: Schematic of image formation system

The direct image scanning techniques use simpler optics, such as a camera lens.

2.1.4 Transducer

The function of the transducer, which is the main part in an optical scanning device, is to generate an electronic signal out of an optical one.

Its principles and uses will be examined in some detail in paragraph 2.2 of this chapter, and in particular there will be a discussion on what form of optical image acquisition scanning system is presently available for different industrial applications.

2.1.5 Image processing

The image processor has two main functions: pre-processing and classification.

The function of the pre-processor, which follows the transducer, is to reduce the rate at which information is presented to the classifier, by removing as much unwanted data as possible at an early stage. The classifier then identifies the object under consideration.

The pre-processor may be a feature extractor (in feature space analysis), or a non-terminal string generator (as in linguistic classification).

2.2 Optical Image Acquisition Systems

This section outlines the operating principles of the most commonly used optical image acquisition scanning systems, for image data acquisition,

such as laser scanners, television (T.V.) cameras and photodiode arrays.

Viewed as a system, the input or stimulus variable of the transducer of an optical scanner is radiant energy (e.g. light, X-ray photons, etc.). The output or response variable is an electrical quantity, typically current in an electron beam scanning the electronic image.

For the purpose of this study, the aspects to be compared will include:

(i) Distortion,
(ii) Resolution,
(iii) Depth of focus,
(iv) Spectral response,
(v) Bandwidth,
(vi) Noise,
(vii) Cost, maintenance and practical considerations.

The choice of sensors and the measurement or inspection techniques is also dependent upon the size and shape of the components, as well as the desired speed of operation.

2.2.1 Laser scanner

The laser scanner is essentially a sequential scanning device. A spot of light is moved along the line of scan across the surface undergoing inspection. Light reflected from the surface may then be sensed by a photosensitive transducer, such as a photomultiplier tube. Depending on the particular application, either the specular or the off-specular reflection may be sensed.

The laser spot may be scanned across the inspection surface, as it progresses along the scanning line, by means of a rotating mirror polygon.

2.2.2 Television camera tubes

In the camera tube the elements of a picture are scanned serially at a rate sufficiently high that they appear to the eye to be simultaneous. Camera tubes can be very useful for pattern recognition, provided the compatibility of the scanning type, speed and input/output signal characteristics have been carefully evaluated.

Modern camera tubes, such as the orthicon and the vidicon, accumulate the information from all of the light all the time, and hence have

a sampling ratio of unity compared with the 10^{-5} of the early dissector-type tubes. The orthicon introduces the principle of low velocity beam scanning as well as the vidicon. The target in the vidicon is a photoconductive rather than a photoemissive insulator.

2.2.3 Photodiode arrays

In the camera tube an electron beam is used to scan a charge pattern on a target and there a vacuum tube is essential. However, in the last decade or more it has been possible to scan elements of a photoconductive target by circuit means only. As early as 1969, Weimer et al showed that a 256-line (6×10^4 picture elements) can be addressed or scanned at television rates by a matrix of leads so arranged that each picture element lies at the interaction of a horizontal and vertical lead. The 6×10^4 picture elements were accessed by 5×10^2 leads, 256 for the vertical and 256 for the horizontal lines. The total number of leads to the device was reduced from 512 to 64 by the use of integrated decoder circuits.

The whole system is based on the principle of operation of a photodiode, almost invariably silicon.

An important improvement in the method of scanning was suggested in 1969 by Sangter and Teer, using a method called "a bucket brigade", and by Boyle and Smith in 1970, using a method called "charge coupling". In both cases, the charge pattern, stored in a line of picture elements, can be stepped across to one edge of the target at video rates and fed into the first stage of a video amplifier built into the same silicon chip on which the picture elements are located. Successive lines of a two-dimensional array can be scanned off in this way by using Charge Coupled Devices (CCD).

More recently, a new device, called "charge injection device" (CID), has been introduced. In this case line arrays are formed of MOS capacitors. The top electrode of this capacitor is normally held at a potential -ve enough to form an inversion layer on the silicon surface below it. To read out this charge, the voltage of the top electrode is reduced to zero so that the collected charge is injected into the silicon substrate as a signal.

In contrast to the CCD sensors where the signal charge is transferred through the array to an output circuit, the charge in the CID device is confined in the CID structure to an imaging site during sensing. Site addressing is accomplished by using an X-Y coincident voltage technique

not unlike that used in digital memory designs.

Information on commercially available CID arrays is limited. Moreover, such devices are produced mainly as two-dimensional matrix arrays. CID arrays have not as yet found wide acceptance in industrial application, although many believe that CID arrays are potential solid-state image scanners of the future.

2.2.4 Distortion

The distortion associated with scanning devices can be geometrical and/or electronic distortion. As sources of distortion are the non-linear relation between input and output of a component; the non-uniform transmission at different frequencies; the phase shift which is not proportional to frequency. The linearity of a scanner depends upon any mechanical gears in the system, which can introduce at most jitter or linear error. The scanners must contain provisions for pin-cushioning correction, spot size correction and grey level correction in order to compensate for the difference between the behaviour of the system near the edge and at the centre of the frame.

2.2.4.1 Laser scanner

One source of distortion in laser scanners is jitter (i.e. instability either in amplitude or phase of a signal). This may be a combination of both lateral and longitudinal jitter.

2.2.4.2 T.V. camera tubes

One source of distortion in cameras results from the normal types of distortion found in an optical system using lenses, such as spherical and chromatic observation, stigmatism and coma. However, high quality lenses are readily available and such distortion would not be present to the extent that it would be a problem.

The main form of distortion is non-linearity within the camera scanning electronics. This results in an output video signal that represents image elements improperly with respect to time. This is conceptually similar to longitudinal jitter in laser scanners. Non-linear scan of evenly spaced elements would result in frequency distortion of the video output signal waveform.

2.2.4.3 Photodiode arrays

The geometrical precision of manufacturing photodiode arrays is extremely good. For example, an array presently available can have 1024 photodiodes on 25.4 µm ± 2 µm (non-cumulative) centres, each having dimensions of 25 x 25 µm resulting in an array approximately 25 mm long. This is contained in a 24-lead dual-in-line package with a glass window.

More recently, 13 µm pitch of diodes are also available. Most of the distortion, therefore, is derived in the imaging lenses, as described in the T.V. camera section above.

2.2.5 Resolution

Resolution refers to the number of distinguishable points (spots) which the scanner is capable of sampling in the scanning field and make the analysis of the sample possible.

2.2.5.1 Laser scanner

For a laser scanner the smallest practicable spot size attainable is about 10 µm (0.5 thousandths of an inch). The width of scan obtainable, for reasons of distortion effects, however, decreases with decreasing spot size. Thus, for example, for 10*1/1000" spot size the maximum scan width is about 24 inches, and for 1 to 2*1/1000" spot size this coverage is reduced to only about 4 to 6 inches.

2.2.5.2 T.V. camera tubes

The number of lines of resolution may be defined as the number of black and white lines of equal width that could be contained in the vertical dimension of the picture.

Vidicon resolution capabilities may vary over a wide range. Some tubes can display well over 1000 lines of horizontal resolution. As the resolution elements become smaller, however, the tube output signal decreases in amplitude caused by the finite size of the beam spot and the granular structure of the target material.

The horizontal resolution capability is also dependent upon the bandwidth of the video amplifier in the camera system, and is thus said to be "bandwidth limited".

The vertical resolution of a T.V. camera in its ability to resolve horizontal lines in a field depends on the number of scans that are used

per frame. The standard system used in the U.K. for broadcasting is 625 lines, or a scanning ratio of 625:1. It is possible, however, to achieve ratios of 1200:1 and higher.

There is a close relationship between vertical and horizontal resolution and bandwidth, and this is discussed in some detail in the "bandwidth" section of this chapter.

2.2.5.3 Photodiode arrays

The resolution of a photodiode array system is governed largely by the magnification of the image falling on a diode.

One-dimensional arrays of up to 256 diodes at 100 μm pitch and up to 1024 diodes on 25 μm pitch, and two-dimensional arrays of up to 1728 diodes, are commercially available; thus, with 1:1 imaging resolutions of 13 μm pitch may be attained.

2.2.6 Depth of focus

Once it is determined that a self-scanned array will perform the function required, the next problem is to form an image on the array of the object to be inspected.

As lens distortion and errors are small compared with the system resolution, simple photographic lenses may be employed.

These are defined by working distance, magnification and field of view. Also, the focal length of the lens and the space required are given by the following equation:

Focal Length (L) + Camera Housing = Space Required.

In Figure 2.3 below it is shown how the optical system is represented.

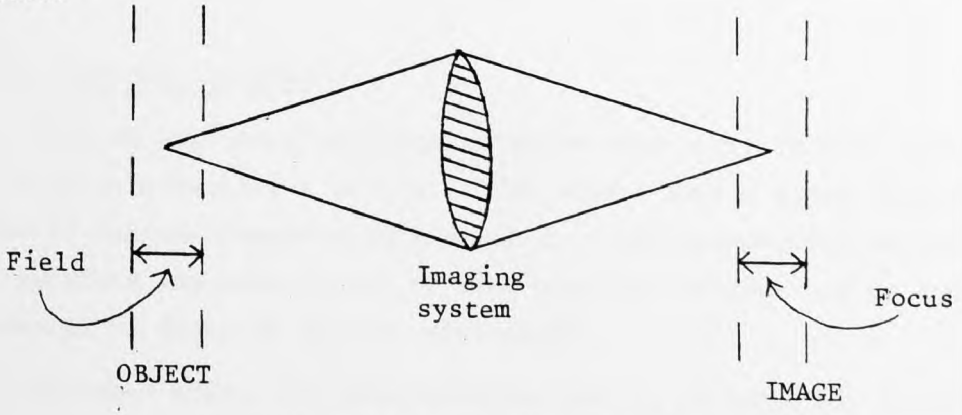


Figure 2.3: Optical system

The depth of focus of an imaging system is the difference between the maximum and minimum distance between image and lens, and it is improved by reducing the aperture of the lens, but is extremely costly in the use of photons. We can also improve depth of focus by controlling position of objects.

2.2.6.1 Laser scanner

Providing the scanning width is not too great, the amount of defocusing is small. For greater widths, some form of curved specimen holder or correction of the optical field is required.

2.2.6.2 T.V. camera tubes

In T.V. camera tubes the depth of focus depends on various factors, the most important being lens-to-object distance, focal length and aperture.

2.2.6.3 Photodiode arrays

The depth of focus for photodiode array scanners is very similar to that of the T.V. camera, since a similar type of optical system would be used. The resolution is a function of the diode area and spacing.

The starting point of the optical design is usually the resolution requirement, which defines the magnification (or reduction) of the system:

$$\frac{\text{Resolution Requirement}}{\text{Diode Pitch}} = \frac{1}{\text{Magnification}} = \frac{\text{Object Distance}}{\text{Image Distance}}$$

A fair amount of defocusing would not have any effect on the output for objects below a certain size.

2.2.7 Spectral response

When we are using photoconductors we must bear in mind that these are frequency-selective devices. This means that a given intensity of light of one wavelength when dropped on a photoconducting material will not generate the same number of free electric carriers as an equal intensity of light of another wavelength.

In other words, the photoelectric yield, or spectral response, depends upon the frequency of the incident radiation.

2.2.7.1 Laser scanner

Here the majority of photoelectric materials used operate in the visible or near infrared wavelength regions. However, tubes operating in the far-ultraviolet and solar-blind (far-ultraviolet but insensitive to solar radiation) regions are available.

So, by the colour of a laser beam we can determine approximately the spectral response of the photomultiplier tube, since a specific material used will give a specific colour.

At short wavelengths, the spectral response is a function solely of the enhance window material characteristics; at longer wavelengths it is a function of the intrinsic property of the photocathode used in the photomultiplier tube.

The response for various different types of window material is shown in Figure 2.4, and for photocathodes in Figure 2.5.

These are only schematic graphs.

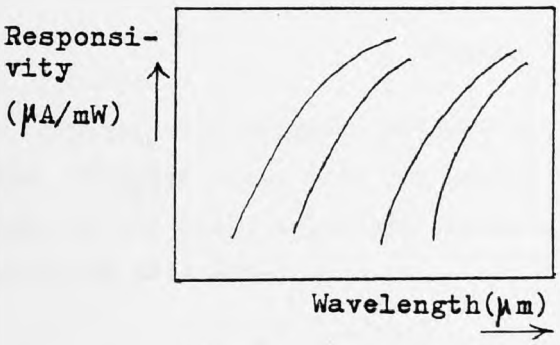


Figure 2.4

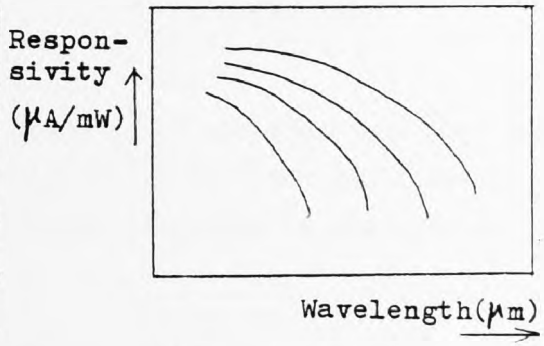


Figure 2.5

2.2.7.2 T.V. camera tubes

The spectral response of most of the T.V. cameras, and especially of vidicons, is similar to that of the human eye. The response of a typical vidicon is shown in Figure 2.6. However, vidicons working in the infrared or ultraviolet regions can be produced by using suitable target materials.

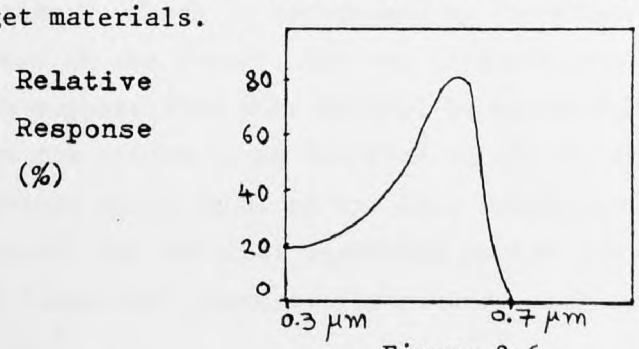


Figure 2.6

2.2.7.3 Photodiode arrays

The spectral response of a silicon diode is basically the same shape for all devices, but varies in minor detail, depending on such factors as the silicon resistivity, diode junction depth and silicon dioxide thickness over the surface. The response for a typical photodiode is shown in Figure 2.7 and it may be seen that the visible region and a substantial portion of the near infrared region are covered.

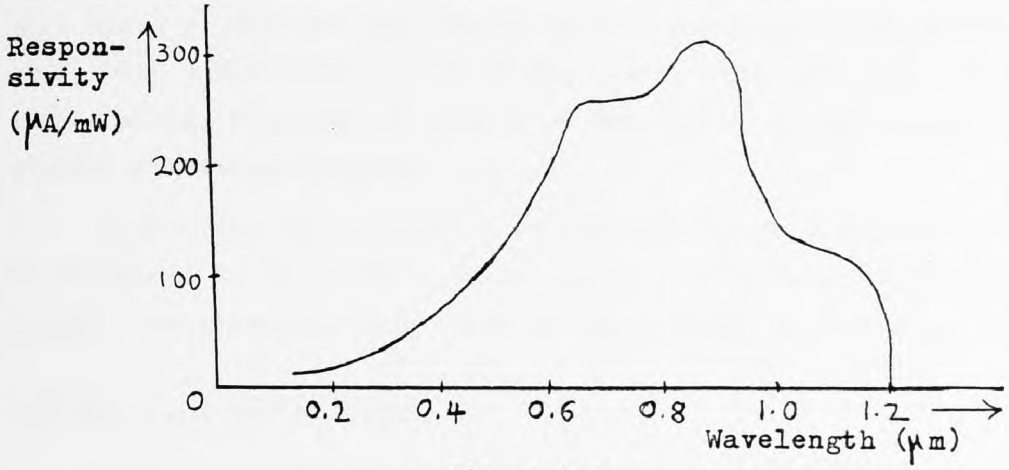


Figure 2.7

There is a mismatch between the response of the diode and the human eye. However, this does not necessarily matter for many applications and, in any case, a certain amount of light filtering is possible to overcome this effect.

2.2.8 Bandwidth

To illustrate the significance of the bandwidth, and more precisely of the video bandwidth, let us consider the following example:

Suppose that a television camera is recording a scene which is totally dark except for two points of light which lie in the same plane as a scanning line. Assuming a perfect camera system (including perfect optics), the video signal will contain two electrical impulses separated by a time T, which is determined by the frame repetition rate, the number of lines in the raster, and the distance between the points in the image. Let us suppose also that we will be satisfied if the transmission system allows the points to be resolved in the received picture when they are a distance apart equal to the line separation. That is to say, the horizontal and vertical resolving powers are to be comparable. If there are N (complete) pictures per second, each composed of L lines, then $T = 1/NL^2$.

The video bandwidth is inversely proportional to T and therefore directly proportional to the number of resolved "points" in the picture. The greater the picture detail, and thus the greater the rate of information transmission, the wider the required bandwidth.

2.2.8.1 Laser scanner

There are mechanical limitations to operating speed of a laser scanner. It is possible to achieve a mirror polygon of 12 sides rotating at a speed of 400 revs/sec, which gives a scanning frequency of 4800 scans/sec. With a spot size of 0.5 μm and a scan width of 1.5 m, this would give a signal frequency of nearly 30 MHz, which is high even for analog signal processing equipment.

In practice it is possible to achieve a practicable and analog bandwidth of up to 10 MHz. It is possible to attain such bandwidth because the photomultiplier tube in the scanner is virtually noise-free.

2.2.8.2 T.V. camera tubes

Since the bandwidth is closely related to resolution, in fact it limits the resolution of a T.V. camera. This relationship can be shown with an example:

Suppose a 1000-line horizontal resolution with a scan ratio of 625:1 is required. The frame rate is normally 50 Hz (the mains frequency in the U.K.). The number of cycles on a horizontal line is half the horizontal resolution since an adjoining pair of black and white bands constitute a single cycle. The horizontal rate is therefore given by:

$$\text{Horizontal rate} = \frac{1000}{2} \times \frac{50}{1} \text{ Hz} = 25 \text{ KHz.}$$

Thus, the time for one horizontal vidicon scan is 1/25 KHz = 40 μS.

This time, however, includes flyback blanking time, which is about 15% of the total (i.e. 6 μS). The active time is then 34 μS.

And the bandwidth is given by:

$$\text{Bandwidth} = 500 \times 1/34 \times 10^6 \approx 15 \text{ MHz.}$$

It can be seen, therefore, that large bandwidths are required for high resolution in T.V. camera scanners.

2.2.8.3 Photodiode arrays

The maximum diode-to-diode scan rate is normally set by shift register performance and is 3 - 5 MHz typically. However, rates of between 10 to 35 MHz have been quoted using a simple chain of logic inverters in place of a shift register.

The practical line scan or frame minimum rate is limited by the integration time of the individual diodes. This integration time is the time between the recharge (scan) pulses to a diode. It is during this time that the charge leaks away, and at the end of the period the total charge lost is proportional to the integration illuminance.

The line scan or frame rate is given by:

(Diode-to-diode scan rate)/(number of diodes)

providing the integration time limit is not violated.

Thus, for a diode-to-diode scan rate of 1 MHz, with a two-dimensional array of 64 x 64, the frame rate would be nearly 250 Hz.

As well as being a limit to the minimum integration time, there is also a limit to the maximum integration time, which varies exponentially with temperature.

2.2.9 Noise

The significance of considering the noise, and especially the signal-to-noise ratio, in a picture, is that, in general, this ratio is proportional to the diameter of the picture element. Also, the noise-limited resolution of a picture is proportional to the contrast of the test pattern. In addition, higher presentation brightness requires pictures of higher signal-to-noise ratios.

Other sources of noise can arise from subsequent signal treatment (secondary emission multiplication noise, output-resistor noise and amplifier noise).

Also, other noise is due to the characteristics of the associated equipments.

2.2.9.1 Laser scanner

The noise in laser scanners can be considered to be purely instrumental and a signal-to-noise ratio of better than 60 dB would be expected.

However, the existence of dark current in the photomultiplier tube may be due to field emission, gas ionisation, interelectrode leakage, residual radio-activity or cathode thermionic emission, and hence the noise is concerned with the characteristics of this dark current. Virtually all sources of dark current may be eliminated by careful tube design, except photocathode thermionic emission.

The amount of dark current varies from less than 1 electron/cm^2/sec to 10 electrons/cm^2/sec, which is less than 10^{-17} Amps for a 1-inch diameter photocathode.

2.2.9.2 T.V. camera tubes

The dark current of a camera tube is the current generated by scanning the target area whose face is in complete darkness. The current is present because the photoconductor is not a perfect insulator in darkness, and small amounts of positive charge (in vidicon) are built up between successive scans. Also, the dark current increases with target voltage.

In the vidicon, for example, other noise is dependent upon the output level, since the lower the output, the more amplification is required in later stages, hence amplifying the dark current noise. Thus, noise is more troublesome at low lighting levels since the vidicon output increases with increasing illumination.

In Figure 2.8 below the variation of output against illumination for different dark currents is shown only schematically.

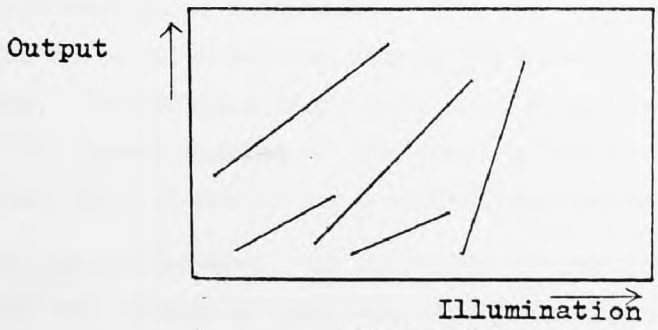


Figure 2.8

2.2.9.3 Photodiode arrays

Three components contribute basically to noise in photodiodes. These are: clock pattern breakthrough from the rest of the circuitry included on the chip; summed dark leakage current; and thermal noise.

A typical maximum figure quoted for noise is ± 2% of signal at 5% discharge per integration time (at 200 KHz scanning rate).

Also, a typical figure for dark leakage current of 500 pA mm^{-2} defines the maximum integration time which can be used in a given application, and hence the minimum light level that can be detected.

2.2.10 Cost, maintenance and practical considerations

Current and much future automatic inspection equipment is mainly based on flying-spot laser scanners, where resolution of 10,000 picture points across each scan and throughput speeds of several thousand metres/minute are possible. Also, because of the limited presence of noise, the laser scanners at present available for automatic inspection are more expensive, with a range of prices around £10,000.

Although T.V. systems are well developed, resulting in low capital cost (around £800), the high noise levels and the cost and inconvenience of replacing tubes at regular intervals, as well as the consequent re-calibration of the instrument, are great disadvantages.

This is in contrast to the absolute spacial stability of diode arrays, which are also more easily interfaced to modern electronic processing circuitry. A fully-engineered sensing system based on a 1728 elements of linear CCD array would cost about £4,000.

Since CCD arrays can be fabricated with closer centre-to-centre spacing of the sensing elements, they offer better resolution. In order to improve the image definition, however, it is often necessary to introduce image reduction techniques. Moreover, with small arrays such image reduction is often essential simply to accommodate the image pattern on the array. The reduced image size then effectively cancels the advantages offered by closer spacing of the sensing elements. Thus, doubling the array size may not result in two-fold improvement in resolution.

Unpredicted movement or vibration of the component in relation to the array can introduce some inaccuracy in measurement. Sideways movement across the optical axis causes uncertainty in detecting the transition edges and introduces extraneous noise in the video signal. Errors caused by such noise can be minimised, and in some cases the overall accuracy improved by statistical averaging of a large number of readings (Fry, 1976).

Any movement of the component in the direction of the optical axis

can, under certain conditions, give rise to changes in optical magnification of the image, effectively introducing further noise into the video signal.

An optical imaging system which uses only the axial beam for imaging, often referred to as a telecentric focusing system (Purll, 1978), can eliminate the effects of axial movement.

In many applications, the use of an external binarising threshold masks any discrepancy in the response of individual elements in the array. Variation in responsivity of the order of 10% on its own may not cause any serious problem. However, in more demanding applications, where finer details of component geometry have to be inspected, the cumulative effect of all possible forms of error (including responsivity error) may add up to a value which may cause concern. Methods have been devised (Fry, 1979), whereby the correction factors for individual sensing elements can be found experimentally. The digital codes can be used to control the current output of an A/D converter and thus define the required gain of individual elements within prescribed limits.

2.2.11 Comparison chart

	Laser scanners	T.V. cameras	Photodiode arrays
Distortion	Geometric distortion due to circular scan, easy to eliminate	Non-linear scan, geometric distortion, astigmatism, coma	Due mainly to imaging lenses
Resolution points/scan	≈ 10,000	≈ 1,000	≈ 2,000
Depth of focus	Very low	Good	Good
Spectral response	Depends on type of photocathode and window materials	Similar to the human eye	Covers wider range than in T.V. cameras
Bandwidth	By response of photomultipliers it gets up to 85 MHz	20 MHz	10 - 35 MHz
Noise	Mainly due to the dark current (Low)	Depends on the output level. So higher noise with increase of illumination (Medium)	Due mainly to dark current and thermal noise (Medium)
Cost (1980)	≈ £10,000	≈ £800	≈ £4,000 (for CCD)

2.2.12 Discussion and conclusions

Although linear solid-state sensors still cannot compete with electron beam scanned tubes in terms of uniformity and area resolving power, linear solid-state sensors in the form of self-scanned photodiode arrays are now widely used.

Self-scanned imaging arrays combine, as the name suggests, the dual functions of detecting the incident radiation and of reading out this information periodically. This is because silicon, which is the pre-eminent material for the fabrication of integrated circuits, is, fortunately, also a very suitable material for the detection of visible radiation; having a bandgap of 1.1 eV it has a useful working range of 0.4 to 1.1 μm which covers the whole visible spectrum plus the near infrared. Moreover, an attribute for solid-state sensors which is now beginning to be seriously exploited is the precision with which individual sensor elements can be accessed by virtue of the digital nature of the addressing pulses. These features, and the ease with which they enable the output of a self-scanned array to be coupled to digital signal processing systems, are likely to lead to widespread use of such arrays in more applications.

The principal rivals of the photodiode arrays are the conventional T.V. cameras and flying-spot laser scanners, which have differences mainly practical rather than fundamental. Environmentally, they are larger, less robust and require less convenient power supplies. Also, the high cost of laser scanners and continuous cost for maintenance of T.V. systems make the use of the solid-state optical scanners even more attractive. However, the appropriateness of using self-scanned arrays and the design limitations imposed by the properties of such devices must be considered in a given situation.

In addition to its obvious practical advantages, such as small size and robustness, a solid-state array offers the very important and fundamental advantage for measurement systems (as opposed to other imaging systems) that the division of space into resolution picture elements (pixels) is defined stably and linearly by the geometry of the chip itself. It has, however, two major performance limitations which cannot be ignored.

The first limitation is the element count. Although the measurement accuracy can sometimes be increased by interpolation, there is still no alternative to laser scanners when intrinsic resolutions of greater than

1 in 2000 are required, for instance in high-resolution defect and shape inspection.

The above limitation occurs because optical edges are not perfectly sharp, and the edges in the image will not necessarily be spatially synchronised with the element spacing, so that one or more elements in the vicinity of each image edge will give a signal level part way between black and white. The measurement accuracy is therefore not absolute, but is determined by the optical magnification and the number of elements which can be disposed across the width of the image; at present up to around 2,000 elements can be obtained in linear arrays.

The second limitation is non-uniformity .

Although there are many industrial applications in which this is not a problem, in low-light-level applications, or those requiring faithful grey-level representation, the device has to be calibrated in the signal processing system. The systems designer will have to trade-off the disadvantages of such a calibration requirement against the problems to be found with alternative imaging systems, such as T.V. cameras.

Unfortunately, both element count and uniformity are limited by semiconductor process technology, which may not improve as rapidly as the advances in integrated circuit design itself.

However, for many non-critical applications almost any type of self-scanned array will suffice, and most choices will therefore be made on grounds such as price and availability.

CCD arrays development has attracted so much publicity in recent years that people tend to assume that CCD arrays are the best solid-state solution to any imaging problem. This is not necessarily so. The major advantage of CCD arrays is their low read-out noise, something which also led to their being considered for use in the present system, despite some non-uniformity which they introduce and which was not considered critical for our application.

A detailed account of the type of CCD self-scanned array used in the present system and its performance will be given in chapter 6 of this thesis.

Some estimates can be made as to the future development possibilities for CCD-based television cameras. Medium-resolution monochrome cameras are already available. Prototype colour cameras, possibly with only

moderate sensitivity and resolution for such applications as video recorders, will become available in 1981-82, with higher performance units suitable for Electronic News Gathering a little later, say 1982-83.

Further development over several more years will probably still be necessary before the very demanding requirements of studio-quality cameras and industrial high-quality inspection cameras can be met.

2.2.13 Summary conclusion

A charge coupled device (CCD), in the form of self-scanned photodiode array, was chosen as most appropriate for use in the present system.

This was chosen in preference to a T.V. camera (e.g. vidicon) as the input device because of its high resolution, rapid response, absence of spatial distortion and because its output is a sequence of samples synchronised to an external clock, thus, in keeping with a digital computer is a much easier task.

A laser scanner would have provided still better resolution, but would have been unacceptably expensive. Further, lighting and viewing conditions would have been more difficult to vary.

Special care should, in any case, be taken in providing adequate and effective illumination.

CHAPTER 3: CRITICAL SURVEY OF METHODS FOR THE ANALYSIS OF SILHOUETTE SHAPES

3.0 INTRODUCTION

In this chapter a brief account is given of the methods which are available for the numerical description of two-dimensional silhouette shapes.

The purpose of this study is to understand more clearly the different approaches to solving the problem of identification of silhouette shapes, and to clarify the advantages and disadvantages of the different methods, in order to make sure that the choice of the technique adopted in the present work is in a position to meet the desired requirements in an efficient and economical way.

There are two assumptions implicit in this account:

(1) That we are dealing with discrete objects, i.e. objects which do not overlap, and

(2) That the objects are to be described one at a time.

In choosing the methods to describe here, it is also assumed that the description is required for further analysis by pattern recognition techniques.

There appear to be two main classes of methods, based on whether the data are collected in the form of polygonal outlines, and they are called Vertex methods, or based on whether the data are collected on a grid, and they are called Grid methods.

In order to assess these methods, or any other method, a number of desirable qualities has been suggested, which the ideal numerical description should have. It will be apparent that no real descriptor possesses all these qualities simultaneously.

3.1 Criteria

The following six criteria are suggested as desirable qualities for numerical descriptions:

(a) Uniqueness

(b) Independence among descriptors used as features

(c) Rotation invariance

(d) Scale invariance

(e) Straightforward physical interpretation

(f) Parsimony.

3.1.1 Uniqueness

The first desirable quality is that of uniqueness. This implies that it is necessary that the features selected to describe a given shape should give one, and only one, description for the shape.

Further, two objects should have the same descriptions if, and only if, they are identical in shape. In that case, they should be considered as members of the same class.

3.1.2 Independence among descriptors

It is desirable that independence of the descriptors, which are to be used as features, ensures that each of them is measuring some different quality of the object.

If the descriptors are independent, it is much easier to eliminate those which contribute little, and concentrate on the more important ones.

3.1.3 Rotation invariance

Rotation invariance implies that if the orientation of a given shape varies during different tests, under which the shape is examined, the descriptors of the shape should remain unchanged.

This property is significant in the case of choosing the right datum point of reference with which the selected features are related, so that they can keep the identifying characteristics of the shape.

3.1.4 Scale invariance

The desired quality of scale invariance implies that the features used to characterise an object should be able to be scaled by a factor without changing the shape of the object. A large circle, for example, has the same shape as a small one.

The above quality is desirable particularly for cases where the amount of data is too large to be accommodated in small memory allocations.

So, descriptors, each of two or three digits big, for example, may be divided by a factor of 10 or 100, say, and become one or two digits big instead.

3.1.5 Straightforward physical interpretation

The quality of straightforward physical interpretation helps because it is easier to have faith in descriptors where there is need to understand what is going on.

Here, for example, it is easier to use as measures quantities with which we are very familiar, such as lengths or areas, instead of square roots of numbers, logarithms, etc.

3.1.6 Parsimony

Lastly, it is desirable that the number of descriptors used is the least which is adequate, and that they are such that they can be acquired and processed in the least time and least resource requirement.

3.2 Jordan Theorem

Before describing some of the types of analysis open to us, it is useful to introduce the Jordan theorem, since it makes explicit a fundamental point about shape analysis.

Simply, the theorem states that there are three parts to a single non-intersecting closed curve: namely, an inside, an outside, and an edge or boundary, as is shown in Figure 3.1 below.

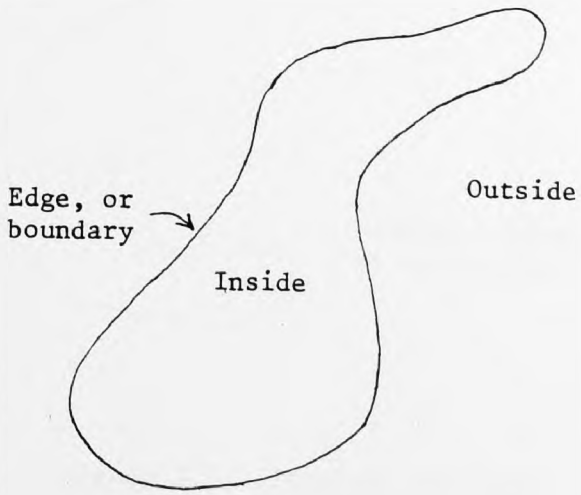


Figure 3.1: Jordan theorem

Some methods of analysis concentrate on the boundary, and some on the inside. An implication of the theorem is that shapes are not always simple, such as circles or rectangles, but they can be much more complex forms, which are not necessarily mathematically easy to define.

In such cases of complicated shapes special handling is required. We say then that such shapes have critical problems.

3.3 Critical Problems

There are two most important problems which affect some methods in handling irregular shapes. These are:

(a) Holes inside the shape, and

(b) Multiple crossing, or re-entrant.

The above problems are shown in Figure 3.2 below:

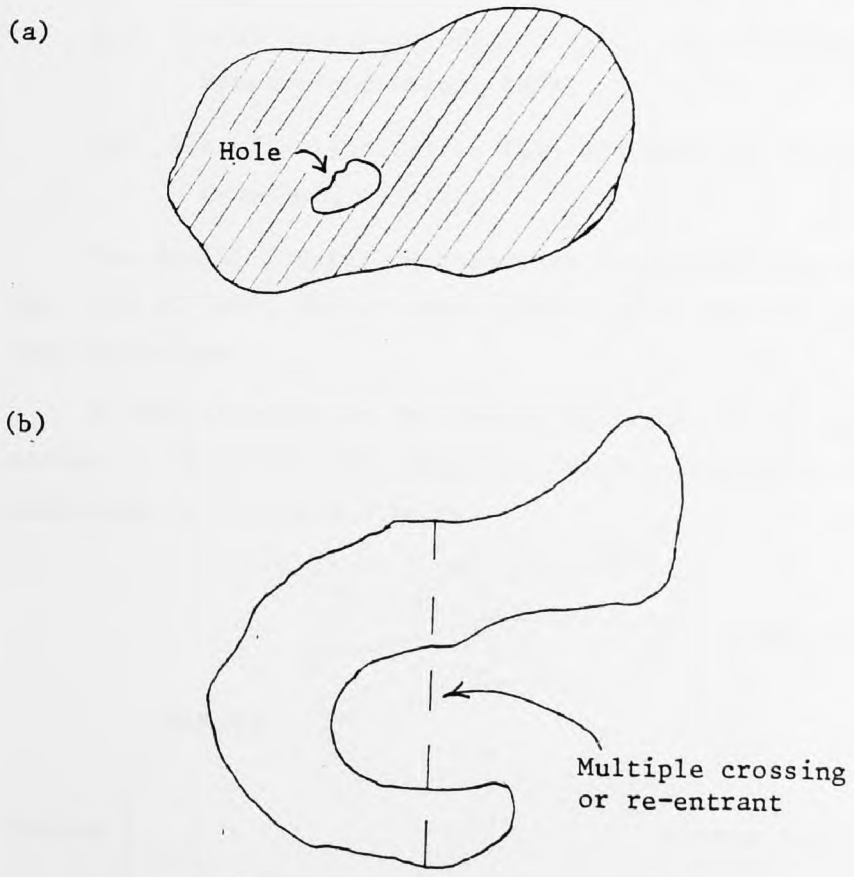


Figure 3.2: Critical problems

If from some chosen reference point inside the object, perhaps the centre of gravity, a radius intersects with the boundary more than once, the curve has a re-entrant.

3.4 Vertex or Polygonal Methods

Vertex methods are those in which the object is approximated by a polygon, the co-ordinates of each vertex being recorded in sequence.

Most of the methods fall into the general category of vertex, and within this group there appear to be two main types: the Fourier, and the Density function methods.

3.4.1 Fourier methods

In this category we have:

(a) Radial function (after Elrlich and Weinberg, 1970, and Schwarz and Shane, 1969)

(b) Bend function (after Zahn and Roskies, 1972, and Granlund, 1972).

The radial Fourier methods were introduced into geology some years ago, and at least six or seven papers have appeared which make use of this technique.

A central point in the object is found, or assigned, and a Fourier series is fitted to the periphery, with reference to this origin, as indicated in Figure 3.3 below.

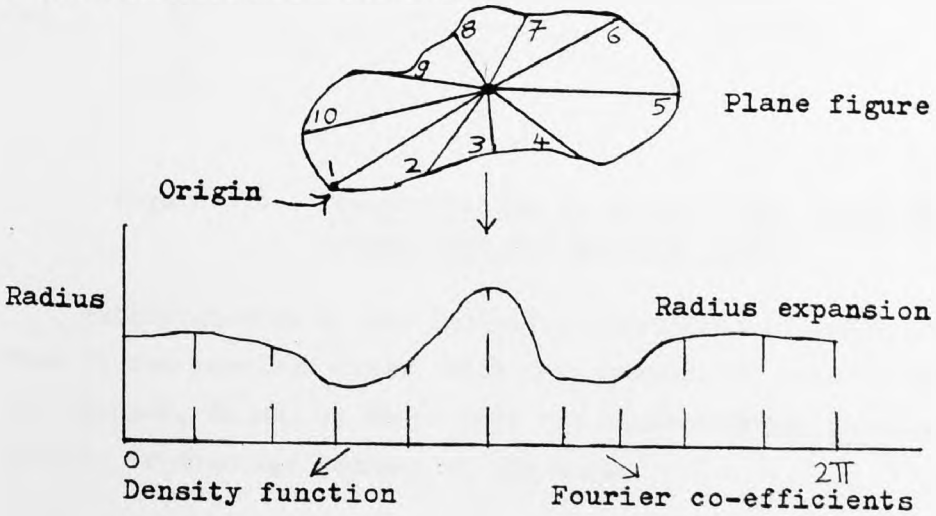


Figure 3.3: Radial method

The above method cannot handle holes or re-entrants. Fourier series have some useful qualities. They can be expressed as the summation of terms made up of amplitudes and phase angles. The phase angles, like all orientation data, are difficult to handle, and usually dropped. The amplitudes, however, appear to contain a great deal of information in themselves since in some shape studies, like grain shape studies, for example, about 90 to 95% of the information appears to reside in the first eight amplitudes. The amplitudes are rotation invariant, and can be made scale invariant with reference to the average radius of the object. The physical interpretation is not very difficult.

The Bend function Fourier method, which has been suggested, does not share the problem of re-entrants, although holes cannot be handled. Here, angular deviations are plotted against length, normalised to fit between zero and 2π, and then a Fourier series fitted to this (see Figure 3.4).

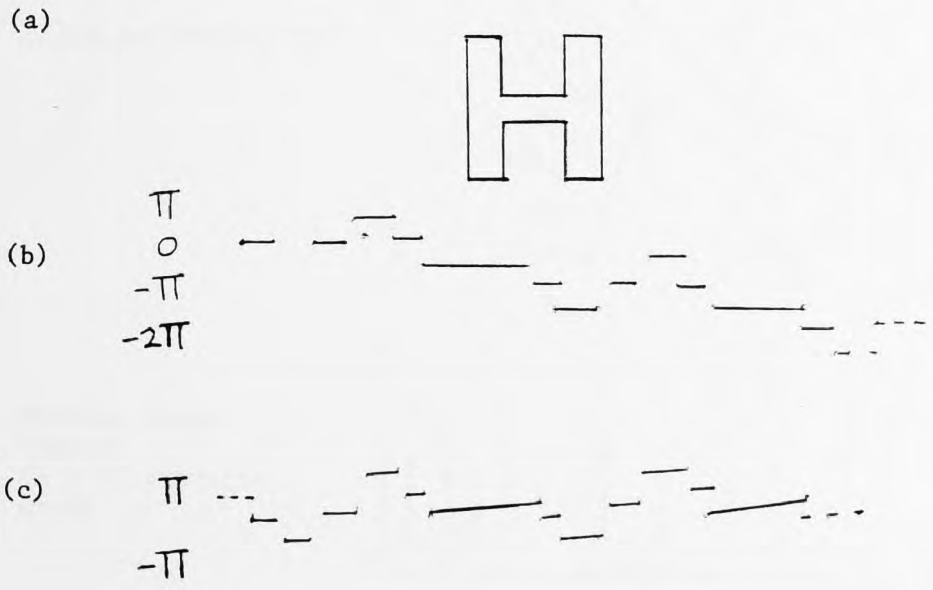


Figure 3.4: Transformation to angular bend function (after Zahn and Roskies, 1972)

Interpretation of the individual coefficients seems more difficult than in the previous case. With this method, if some of the coefficients are dropped, it may be found that the reconstructed boundary intersects itself, or does not connect at the ends.

3.4.2 Density function methods

Density function methods are quite numerous. In this category the following types have been selected:

(a) Slope density (after Nahin, 1972)

(b) Directionality spectrum (after Freeman, 1962 and 1969)

(c) Radius method (after Brezina, 1977)

(d) Angular method (after Piper, 1970).

Nahin's slope density method seems a useful one. The slope density function of a silhouette made up of straight line chords (a polygon), is a sequence of Dirac delta functions, each corresponding to the proportion of the outline which lies at a given angle, as it is shown in the diagram of Figure 3.5 below:

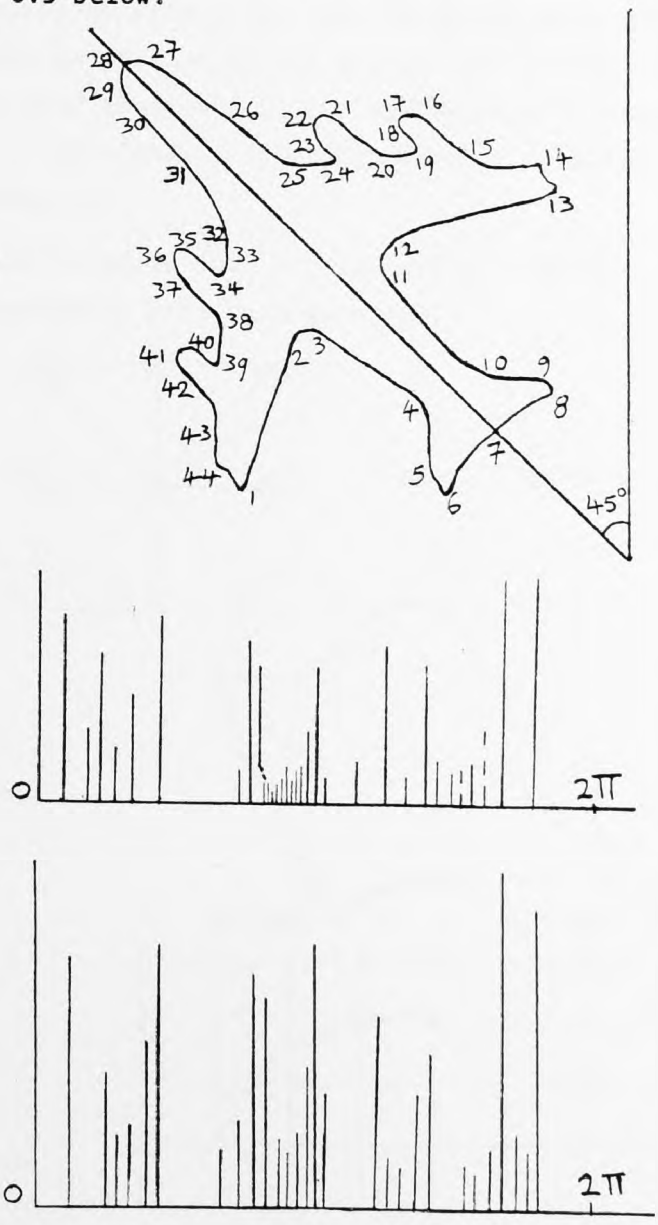


Figure 3.5: Slope density techniques

A rotation of the outline shifts the density function by an easily calculated amount, but to cope with this rotation variance, Nahin suggests that a Fourier series might be fitted to the slope density. The amplitudes of this function would be rotation invariant. However, a slight problem is encountered in approximating any density function by a Fourier series, since eliminating terms, say in an effort to smooth the outline, may lead the function to take on negative values; in other words, it is no longer a true density function. In Figure 3.5 an application of this method is shown. It can be seen that two slope densities are presented, one a machine slope density and the other a true slope density. Nahin originally developed a machine to do the processing, and a computer program to simulate the machine. The machine slope density is based on this simulator. The true slope density, given below the previous one, is the proportion of the Boeing 707 outline at any given angle. It can be seen that there are slight differences between the two densities. Smoothing of the density by application of a moving average would reduce the differences.

A method which bears a great deal of similarity to Nahin's is given by Freeman, and is illustrated here in Figure 3.6.

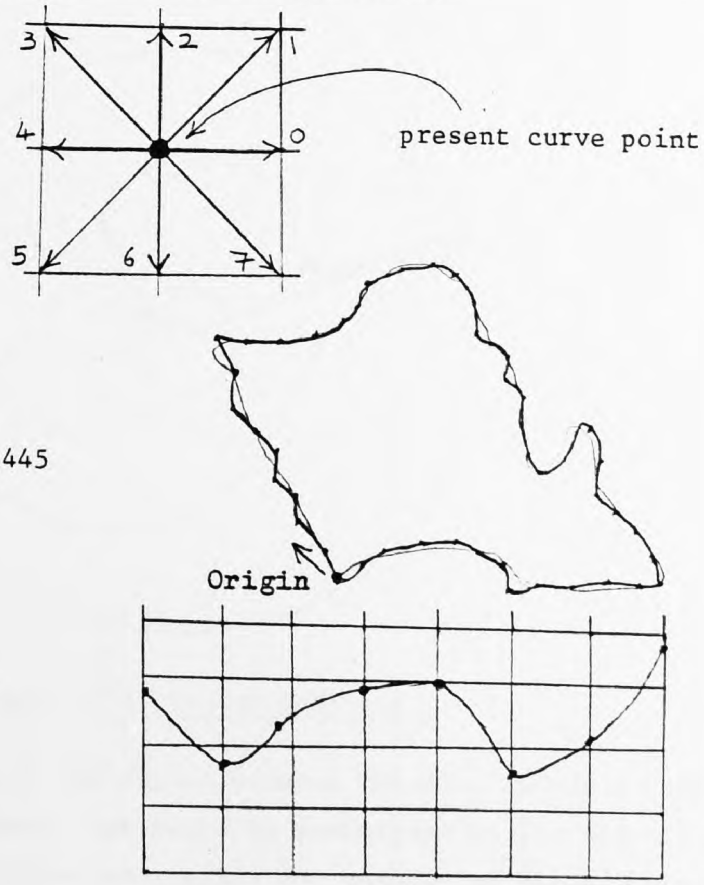


Figure 3.6: Directionality spectrum

There are really two elements to this, firstly, a chain encoding system, where an irregular shape can be simplified into a chain made up of a sequence of eight possible directions. A slight problem with this type of encoding is that reconstruction of the shape may leave a gap at the ends. The normalized directionality spectrum, which is merely the proportion of the outline in any given direction, is just a simplified version of the slope density function given by Nahin. Here there are also problems of rotation invariance.

Another density function method has been suggested by Jiri Brezina, 1977. This depends, like the radial Fourier method, on finding a centre of gravity for the shape, and then considering the lengths of the radii which can be drawn. This type of method should be able to cope with multiple crossings and holes. In Figure 3.7 it is suggested that we are not considering radii between length zero and the maximum possible, but between some minimum value, r_{min} and the r_{max}. There are no problems here of rotation invariance.

A similar method, suggested by Piper, 1970 (see Figure 3.8), takes the angles between successive chords of the polygon, and uses these angles to construct a cumulative frequency curve.

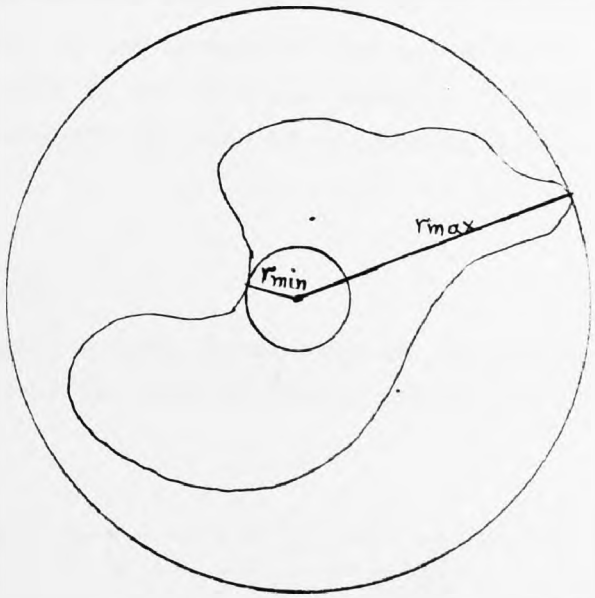


Figure 3.7: Radius analysis

The distribution of the angles between adjacent chords is suggested as a measure of roundness, but might be summarised by its moments, and the moments used in further analysis. The method has the advantage of being rotation invariant, but there is a hidden snag - to obtain a fair

representation the chords ought to be of the same length. This is rather difficult to achieve with digitised shapes.

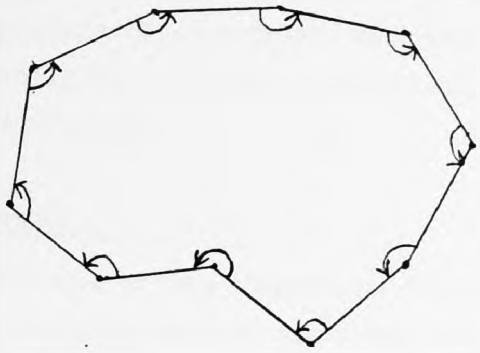


Figure 3.8: Angular change

This suggested use of moments might be applicable in the case of the method suggested by Brezina.

All the density function methods ignore the sequential nature of the polygon outline, and do not permit reconstruction of the outline, and, as we have seen, some of them require further processing before they become rotation invariant. These methods are related to the Fourier methods by more than just the use of similar data. They represent an analysis in the frequency domain, while the Fourier methods represent analysis in the spectral domain. This is a fundamental dichotomy between all analyses of sequential data.

3.5 Grid Methods

If we imagine a grid placed over an outline, a square is shaded in if the majority of its area is within the outline (see Figure 3.9).

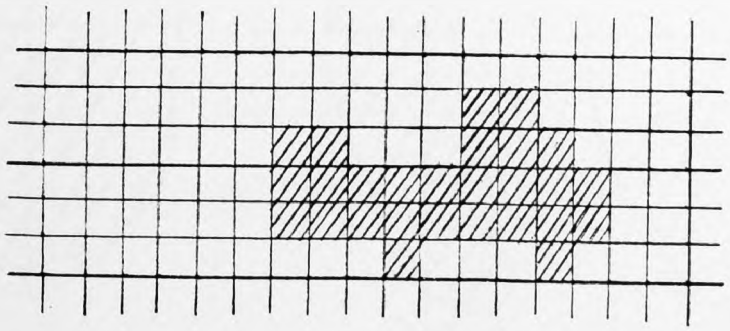


Figure 3.9: Gridded plane figure (after Zahn, 1969)

The example here is presented on a square grid, but a hexagonal grid is an equally viable alternative, and may have some advantages.

Clearly a major advantage of a grid representation is that it may be implemented mechanically. A television raster image, with sufficient contrast, is essentially a binary representation, or similarly by using a photodiode array scanner.

3.5.1 Moments method

Suggesting the use of the moments of the distributions derived from the density function methods provides a way into an account of the one grid method, which also relies on the calculation of moments. The difference lies in the fact that two dimensional moments are used.

In particular, the polygon inside is thought of as a two dimensional body of no thickness, and the mechanical moments associated with such a body are computed. This does not have to be done on a grid basis, but the arithmetic becomes very simple if we are dealing with a binary grid.

The first moment represents the centre of gravity of the body, while the second moments represent the moments of inertia about the axes. Naturally, it is possible to go on to higher moments, but the physical interpretation becomes more difficult.

Medalia, 1970, has used this method in analysis of carbon black aggregates (see Figure 3.10 below), and shows how an ellipse may be derived from the first and second moments.

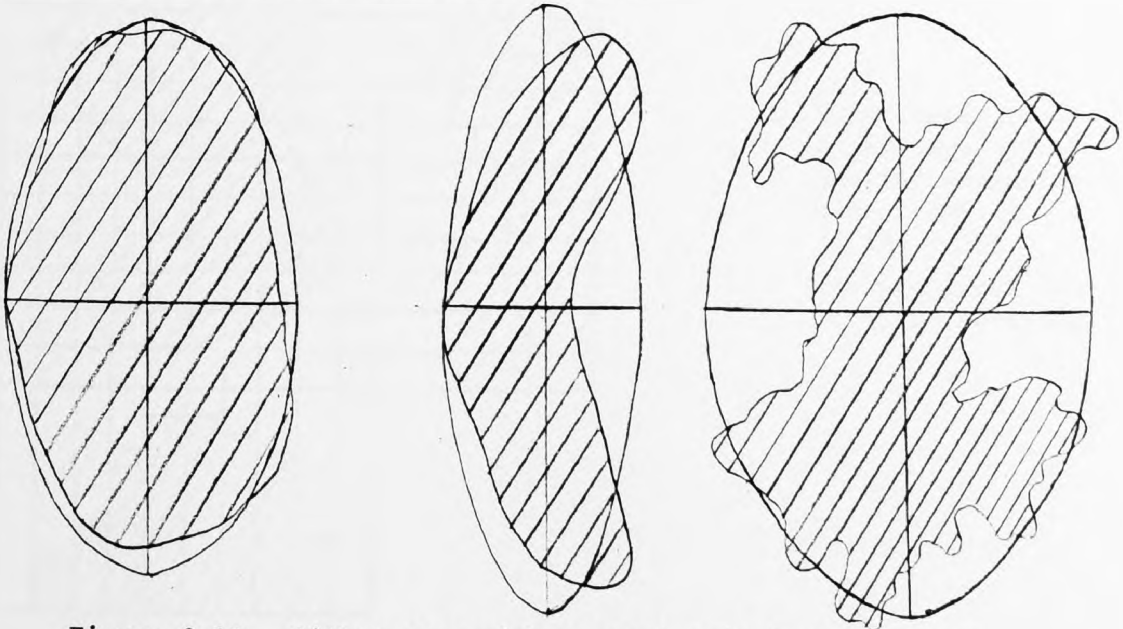


Figure 3.10: Silhouettes with central principal axes and radius equivalent ellipse

This ellipse has the same dynamic properties as the original two dimensional body.

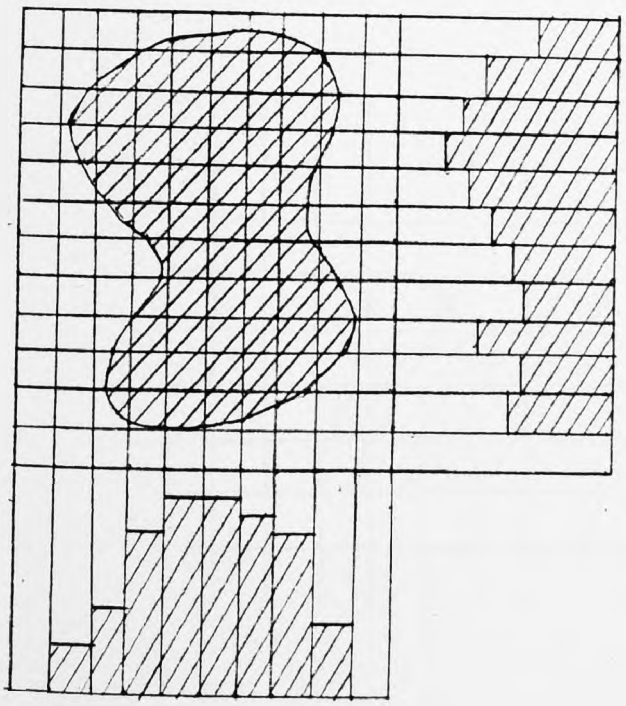
The higher moments have been used in pattern recognition studies, in analysing alphabetic characters. This general field, of handwritten and printed character analysis, is a useful one for the development of shape analysis, and the Fourier Bend function has been used in it.

3.5.2 Longitudinal and transverse slices

The number of filled cells in linear slices, both parallel and perpendicular to the direction of movement of the shapes, are measured and expressed as histograms. The shape of the histograms provides a "signature" which is then compared with a library of stored signatures until a "fit" is found which identifies the unknown pattern. The total number of cells in each histogram is a measure of the area of the shape.

This method offers simple computation. However, it has the following disadvantages. For a given shape, the form of the histogram alters as the shape rotates. Thus, perhaps 30 different pairs of histograms have to be stored (and searched) for each of the several hundred patterns from which a selection must possibly be made.

Further, many shapes may give rise to the same histogram pair; the description is not unique (Figure 3.11).



Characterisation of a shape using two perpendicular histograms of filled cells.

(Longitudinal and transverse slices)

Figure 3.11: Histogram

3.5.3 Heuristic measures

These consist of a variety of measures which may be used to characterise a silhouette pattern which have no systematic relationship with one another. They may be used either together or in combination with other more systematically derived measures to improve overall performance.

The perimeter of a silhouette is measured by counting the number of cells in its boundary. A boundary cell is defined as a cell at the "1" level and hence falling inside the pattern, but having at least one nearest neighbour cell at the "0" level and hence outside (Figure 3.12).

The area of a shape is measured by the number of cells within the shape, i.e. at the "1" level. Unless a shape is very thin, most of its cells do not form part of the boundary and are consequently not a potential cause of error. Thus, area may be measured very accurately in a simple manner and is a good parameter for characterising patterns.

The ratio (PERIMETER**2/AREA) is a dimensionless quantity which is sensitive to pattern shape; this ratio is, for example, 16 for a square, 4π for a circle, and so on. It is unfortunately not unique: many shapes have the same value for this parameter, particularly those with complex outlines. Further, the ratio requires a measurement of perimeter, which is notoriously difficult to measure accurately.

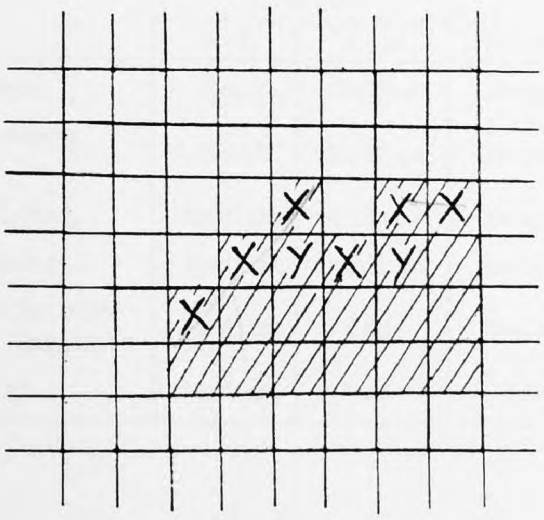


Figure 3.12: Definition of a boundary point

3.5.4 Circle measures

An alternative feature set for identifying silhouette shapes is obtained by noting the number of points (and possibly also the angles of these points) at which a series of circles centred at the centroid intersect the boundary. The radii of the family of circles can be scaled according to the area of the shape, to accommodate variations in size.

Like the method adopted in the present work, which involves a family of radii, this approach can be developed virtually to guarantee recognition; all that is needed is a large enough number of circles!

The circle method does not easily give orientation, but this is not an explicit requirement for every application involving pattern sorting problems.

The circle method would be of interest if it were to provide some definite advantage, for example, simple computation. One possible advantage is that although the radii of all boundary points must be computed, the angles of only those points at which the circles intersect the boundary need be determined.

3.6 Comparison Chart

(a)

	Vertex Methods					
	Fourier		Density			
	Rad.	Bend	Slope	Dir.	Rad.	Angle
Uniqueness	Good	Medium	Medium	Medium	Good	Bad
Indep. among Descr.	Good	Medium	Medium	Medium	Medium	Medium
Rotat. Invar.	Medium	Medium	Medium	Bad	Good	Good
Scale Invar.	Medium	Medium	Medium	Medium	Medium	Medium
Straightforward Physic. Interpr.	Medium	Bad	Medium	Good	Medium	Bad
Parsimony	Medium	Bad	Bad	Bad	Medium	Medium

(b)

	Grid Methods			
	Moment	Long. and Trans. Slices	Heuristic	Circles
Uniqueness	Medium	Bad	Bad	Good
Indep. among Descr.	Medium	Medium	Good	Good
Rotat. Invar.	Medium	Bad	Medium	Bad
Scale Invar.	Medium	Medium	Medium	Good
Straightforward Physic. Interpr.	Medium	Medium	Good	Good
Parsimony	Medium	Medium	Bad	Medium

3.7 Conclusions

We can see from the comparison chart of the previous paragraph that the "circle measures" method is the most promising method, because it gives the best results for most of the desired qualities which the other methods have for the numerical description of a silhouette shape.

However, in the description of the "circle" method it has been indicated that we can have problems so far as orientation is concerned, and that a large number of circles is required in order to have increased accuracy which will guarantee recognition. This means excessive computations which take a long time.

Since orientation as well as accuracy were also important requirements in the development of the present system, and since none of the other methods was in a position to fulfil all the requirements, an alternative method had to be devised with a better performance.

It was thought that if the grid technique was adopted and the radius approach was incorporated into it, the result should bring the best possible solution. This is also because the radial analysis in both Fourier and Density methods indicated good uniqueness and acceptable rotation invariance, both important qualities.

The biggest problem, which remained, was the amount of computation involved, and in particular during the initial measurements for collecting shape information, such as boundary points from a silhouette shape imposed on a grid.

This problem was thought to be solved in the best possible way, i.e. to achieve boundary determination by designing and building a special-purpose hardware.

Considerations relevant in deciding upon the best methodology include processing time minimisation, ease of incorporation of redundancy for error correction, and trade-off of possibility of rejection with substitution in the final match.

Both the methodology and the hardware developed for the present system, as well as the experimental results, are reported in detail in the rest of the chapters of this thesis.

The most challenging remaining problem in devising a special-purpose architecture lies in finding efficient methods for computing radius lengths and angles, something which becomes increasingly demanding, and as yet no methods exist to satisfy this demand.

CHAPTER 4: PROBLEM DEFINITION AND METHODOLOGY

4.0 INTRODUCTION

The present system is required to identify shapes supplied by the industrial company sponsoring this work, which shapes are flat leather components from a selection of several hundred. They are to be presented on a moving line, one at a time and at random orientation.

The shapes range in size from about 1" square to about 6" by 8" and should eventually be recognised at about 4 per second. Shapes scaled in size by 4% must be distinguishable, as must left- and right-handed versions of the same pattern. An overall error rate of less than 1 in 1000 is required. It must also be possible to train the machine to recognise new shapes without having to re-scan those already stored.

Although the shapes are flexible and may have fuzzy edges, they are nevertheless silhouettes, defined completely by their boundaries. However, the full specification of a silhouette still contains far more information than is really needed for identification. Thus, to process this redundant information is unnecessarily costly in computation. It is, therefore, necessary to devise an optimal method for boundary extraction and then to use a processing technique which retains for each pattern only sufficient information to distinguish it from other patterns being considered.

In the present work where the radius comparison method approach is used, by imposing the pattern on a grid, the specification is carried out by radii measured from the centroid of the shape at given angles. The longest radius is used as a datum for orientation. The choice of this method was made in January, 1977, since which time other investigators have announced the use of similar methods.

Area is used as an additional recognition parameter, because it is easy to measure accurately. Boundary length seems a bad measure since its apparent value varies, for example, with orientation, due to quantisation effects. Although there are ways of reducing this error, the additional complexity they introduce is considered unacceptable (Kammenos, 1978).

Although the recognition method adopted here is essentially deterministic, patterns may vary in shape due to knife wear and warping, and there may be mechanical imperfections in the scanning system.

Here, an unknown shape from those stored as classes is identified by comparing its vector of feature lengths, measured in cells, with those for each of the stored prototypes in turn, until a fit is found.

Protection against error is provided by inserting redundancy into each feature vector, i.e. by selecting feature vectors such that each differs from the feature vector of each of all other classes in at least D entries. The classes are then separated mutually by a "minimum distance" D, by analogy with error protection in information encoding.

The error correction precautions also permit the use of lower sampling resolution, and hence of a smaller and less expensive sensing array than would otherwise be required. Also, the introduction of a three-scan technique provides an optimal method for edge detection in a two-dimensional binary image.

Since the work was carried out in the laboratory, where a small computer, such as a PDP 11/10, was used for the on-line operation, with a limited memory capacity, and a scanning table whose moving speed was too slow for the speed capability of the system developed, it was only possible to identify shapes from a selection of one hundred samples, which on average were recognised at a speed of about 4 seconds per shape.

Inevitably, the system will make some mistakes. The possible mistakes during the identification process are of two kinds: it may reject a shape which it cannot identify or, alternatively, it may assign a shape erroneously to the wrong class. This second kind of failure is called substitution. Substitution errors cause much more trouble in production than rejection errors.

4.1 Properties of a Silhouette Shape

A natural image is a map of the distribution of light energy over a two-dimensional plane surface, leading to its mathematical definition as a function $f(x, y)$ of the two orthogonal spatial variables x and y. For monochromatic images, $f(x, y)$ is a scalar which is continuous, real, single-valued, positive and well-behaved mathematically. The value of the function at any point is termed the grey level at that point.

Before being processed on a digital computer, an image must first be converted into an array of numbers. This is achieved by sampling it

over a two-dimensional rectangular grid of points normally separated by equal distances d in the x-direction and d in the y-direction. The effect of this operation is to convert the continuous image into a sampled function f'(nd, md), specifying the amplitude f' measured at the point defined by the integers n, m. The samples are termed "picture cells", often abbreviated into "pixels".

The size of the sampling interval d is a compromise between two conflicting considerations. If d is too small, an excessive quantity of data will be required to represent the image. On the other hand, if d is too large, information present in the original image will be lost in the sampling, and a degradation introduced. If the sampling interval is larger than $1/2f_{max}$, where f_{max} is the highest spatial frequency present in the image, then energy at spatial frequencies higher than $f_s = 1/2d$ will re-appear at lower spatial frequencies "reflected about" f_s. This phenomenon is termed "aliasing"; more than 5% of aliased energy within an image causes a degradation which is instantly perceivable visually.

For many digital processing applications, particularly when special-purpose hardware is to be used, the grey level of each pixel is quantised so that the full range of brightness occupies one of only a limited number of discrete levels. Since the human eye can distinguish no more than 128 levels in an image, this number is often used. Each pixel may then be stored and processed as a 7-bit word. Unfortunately, the process of quantisation introduces an additive noise which is random and cannot be eliminated completely by subsequent processing; it causes a further irretrievable loss of image information.

For monochromatic images, object boundaries are manifest as a connected discontinuity in tone (grey level) or texture. Discontinuity in tone can be detected by thresholding, or by sensing of a gradient. This latter method provides some immunity to variation of background intensity and of ambient illumination, but is susceptible to noise.

Outlines of real objects possess certain properties which are important and greatly simplify their description, and provide immunity to noise. For an image defined on a grid of cells, these are as follows:

(a) The boundary cells form a closed chain, such that each boundary cell has two, and only two, boundary cells as neighbours. A neighbour cell is defined as one which is

either directly or diagonally adjacent to the cell in question.

(b) If a cell is part of the boundary, then at least one neighbour cell must lie outside the silhouette. The converse, incidentally, is not true.

(c) Moving around the boundary in a given direction, only seven relationships between successive cells are possible. These are the relationships forming the Freeman chain code.

These properties may be used to find and eliminate errors in the sensing of the boundary cells arising, for example, from noise.

Similar rules are available for associating possible boundary points located in successive line scans. For example, if cells x_1 and x_2 in a particular scan are boundary cells, then in the following scan there must be boundaries at positions $(x_1 - 1)$, x_1, or $(x_1 + 1)$, and at $(x_2 - 1)$, x_2, or $(x_2 + 1)$. Moreover, the sign of the edge, increasing or decreasing in intensity, must be the same in the neighbourhood of x_1 and x_2 in successive scans.

Further, there must be an even number of boundary crossings within each scan, with each positive transition being followed by a related negative going transition, when the boundary is of a real object lying completely within the scanned field.

4.2 Information Reduction

For an image N by M pixels, and for which each pixel may take one of K different values, the entropy H (which is the average amount of information communicated per pixel) is given by:

$$H = - \sum_{i=1}^{K} p_i \log_2 p_i \quad \text{bits/pixel}$$

and the total information within the image obtained by summing H over all pixels to yield:

$$I = - MN \sum_{i=1}^{K} p_i \log_2 p_i \quad \text{bits}$$

where p_i is the probability of occurrence of the i^{th} level.

The value of p_i may be obtained simply by computing the histograms of amplitude levels, and dividing each bar by the number of pixels.

The above analysis has assumed that the sample amplitudes are mutually independent, but examination of any image representing a real scene shows that this assumption is very untrue. In fact, samples close together are found to be highly correlated, so the image contains much less information (by possibly two orders of magnitude) than the computation above would suggest. But, when the image is stored or processed, a quantity of data $MN \log_2 K$ bits has to be handled. If the appalling redundancy resulting from the excess of data over information in the images can be reduced, it is possible to improve the efficiency of image handling systems significantly.

A surprising amount of useful information can be conveyed even by very simple images. Although such images contain much less data than an original grey scale picture, they can nevertheless retain all information necessary for industrial useful tasks.

The silhouette image is of this kind. A silhouette is produced by imposing a threshold on the grey level values and setting those pixels above this threshold to the "1" level, those below to "0" level.

A silhouette image is, in fact, completely specified by its outline; by coding this outline efficiently, enormous economies can be obtained in both processing and storage.

4.3 Recognition Measures of Silhouette Images

In practical automatic inspection, when the silhouette is acquired as an electric signal by a transducer, which encodes the silhouette on to a rectangular grid of cells, if the outline of the silhouette is close to the boundary between cells, it may spill over into cells on both sides of the cell boundary. Thus, the condition that the cells defining the boundary form a continuous chain, in which only two cells are adjacent to each cell in the boundary, may be violated.

The chain encoded specification of the shape outline is not in itself a good description to use for recognition purposes, since the quantity of information in a chain encoding is also much larger than is, in practice, necessary for identifying shapes.

Further, no simple algorithm is available for rotating a chain encoding, so that variation of orientation is difficult to accommodate.

Also, the number of cells filled by a boundary of given length depends on the orientation of the boundary with respect to the grid of cells.

A linear boundary of length L cells width occupies $L(\pm 1)$ cells when parallel to the cell axis, but only $L/\sqrt{2}$ when at 45 degrees to the axis. The boundary will thus appear to be shortened by a factor $1/\sqrt{2}$, which in this case is a reduction of more than 40% (Clarke and Bedford, 1976), if the count of cells occupied by the boundary is used as the measure of boundary length directly.

Moreover, this effect is not ameliorated by decreasing the size of the cells, so as to obtain improved resolution .

To remove the inaccuracy caused by the cellular structure, it is necessary to find measures which are invariant to rotation with respect to the cell structure.

A definition of a silhouette shape which is far more economical even than chain encoding, though nevertheless efficient for on-line recognition, is obtainable by taking a number of measures on the shape and regarding these as features. Possible measures include the area of the shape, its perimeter and the lengths of a set of radii measured from one unique and easily computed datum point, such as the centroid of the shape.

The centroid is the point having co-ordinates Xo, Yo from which the sum of squared distances of all other points within the silhouette shape is a minimum.

Let the silhouette have N cells. Then, the sum S is given by:

$$S = \sum_{i=1}^{N} \left((Xi - Xo)^2 + (Yi - Yo)^2\right) \qquad (4.E1)$$

By differentiating the expression for S we obtain the following formulae for computing the centroid co-ordinates:

$$Xo = 1/N \sum_{i=1}^{N} Xi, \quad \text{and} \quad Yo = 1/N \sum_{i=1}^{N} Yi \qquad (4.E2)$$

Computation of the centroid does not require that the cells in the

boundary be ordered. But, if a chain encoding is used, we do require ordering of the boundary cells.

Thus, the co-ordinates of the centroid may be determined simply by summing and averaging the location of the cells; its position is clearly unique.

For most silhouette images the number of cells within the image is subsequently greater than the number of cells in the boundary.

Since errors in deciding which cells should be incorporated into the silhouette occur only at the boundary, the centroid is relatively insensitive to this kind of error. However, this insensitivity is not maintained for silhouettes which are long and thin in shape, but again the centroid is not the only datum possible.

As features, one can, for example, take radii measured from the datum (centroid) to the boundary, perhaps at a fixed angular interval of, say, 20 degrees. The longest radius can be used as another datum reference; the angle between this radius and the grid axis would then specify the orientation of the silhouette.

The radius lengths constitute the feature set.

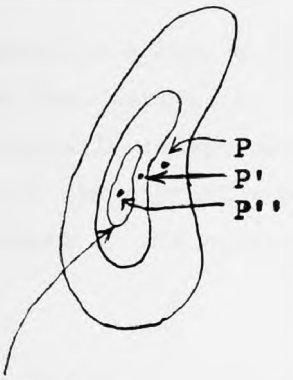
Alternatively, the area of a silhouette image, as measured by the number of cells inside the boundaries of the image, is virtually invariant to orientation provided that the number of cells is large. This is in complete contrast to the boundary length, whose variation with orientation remains even by altering the size of the cells.

Thus, use of the centroid computed by including each filled cell inside the object boundary could be a better method.

Will the centroid thus obtained be the same as that obtained using boundary cells alone?

That the centroid of a silhouette computed using the cells only of the outline is not, in general, the same as that obtained when all cells within the outline are considered, may be seen from the following argument:

Consider a body not symmetric about a point within the body, as shown in the diagram below. Then consider the body to be made up of a series of concentric laminae, each of the same shape as the outline of the body, by decreasing in size successively towards the centre of the body. The centroid of the outline will occur at a point P. The centroid



i^{th} lamina

of a lamina well inside the body will occur at a point P'. The centroid P'' of the complete body is the resultant of the centroids P' due to the various laminae, computed according to the formula:

$$P''_{(x,\,y)} = A \cdot P'_i$$

Since $P''_x = 1/N \sum_{i=1}^{N} x_i$ and $P''_y = 1/N \sum_{i=1}^{N} y_i$ and A is the area coefficient proportional to the number N of cells within the lamina, the centroid P'' will thus move as P'_i moves.

Thus, the centroid determined from the silhouette boundary cells alone would not seem to be useful for object designation. The centroid determined by consideration of all cells within the silhouette is therefore more acceptable in consideration if it is immune to change in the orientation of the body with respect to the cell axes.

We can explore the likely extent of the variations which can occur, as follows:

Consider a rectangular body whose sides are of length N and M cells, respectively. The dimensions measured in cells can vary from N - 1 to N and M - 1 to M. When both changes occur simultaneously, this causes the area to vary from (N - 1)(M - 1) to NM. For N = 50 and M = 100, the area changes from 4851 to 5000 units, a change of 2.98%, which is larger than the change in each individual dimension. This approach enables an order to be determined for the size of cell grid which is necessary for accuracy of a measurement to be retained within given bounds.

The effect of change in orientation is as follows:

When the rectangle being considered is oriented at 45 degrees to the axes (the "worst case" condition), the side lengths reduce to $N/\sqrt{2}$ and $M/\sqrt{2}$, respectively. But the area (measured in number of cells filled) is not $N/\sqrt{2} \times M/\sqrt{2}$, but still N x M. However, the accuracy with which the boundary lengths may be measured (as a count of cells) has degraded by a factor of $\sqrt{2}$.

An important conclusion which obviously follows is that once the

centroid has been established from the area cells, then any distance from the centroid to the boundary of the silhouette shape should remain practically independent of the orientation of the body. And these distances can correctly be used as characteristic properties and recognition measures of the silhouette image.

4.4 Errors in Measurement of Features

Ambiguities resulting from variation in orientation may be removed by selecting properly the feature set for the identification.

As shown below, the longest radius from the centroid to the boundary is not necessarily a unique datum (i.e. there may exist other radii of the same length). However, this does not invalidate the uniqueness of a feature set of radii around the boundary, starting, say, in an anticlockwise sense from a reference longest radius, providing the radii are measured in a specified sense and can be "rotated" during the identification stage until a fit is found.

As we have seen, the centroid may be used as a datum point from which the radii to the boundary can be extracted. The x and y coordinates of the centroid are given by equations (4.E2).

In general, the centroid will lie somewhere within the body. Suppose we then remove n_1 cells from a particular quadrant of the shape. For simplicity, let these form a contiguous cluster, although this assumption is not necessary. The centroid will migrate to a new location Xo', Yo', given by:

$$Xo' = 1/(N - n_1) \sum_{i=1}^{N-n_1} X_i, \text{ and } Yo' = 1/(N - n_1) \sum_{i=1}^{N-n_1} Y_i \qquad (4.E3)$$

However, we can now remove a second cluster counting n_2 cells from the opposite quadrant such that the centroid returns to its original location Xo, Yo. This is expressed by the following equations:

$$\left.\begin{aligned} XO &= 1/\left(N - (n_1 + n_2)\right) \sum_{i=1}^{N-(n_1+n_2)} X_i, \text{ and} \\ YO &= 1/\left(N - (n_1 + n_2)\right) \sum_{i=1}^{N-(n_1+n_2)} Y_i \end{aligned}\right\} \qquad (4.E4)$$

The relationships which must be obeyed by the two cell clusters if the two identical longest radii are to be maintained are shown below:

$$\left.\begin{aligned} 1/n_1 \sum_{i=1}^{n_1} X_i &= 1/n_2 \sum_{k=1}^{n_2} X_k, \text{ and} \\ 1/n_1 \sum_{i=1}^{n_1} Y_i &= 1/n_2 \sum_{k=1}^{n_2} Y_k \end{aligned}\right\} \qquad (4.E5)$$

The above two clusters can be different in form even if they contain the same number of cells.

As an example, consider an ellipse, as shown in Figure 4.1 below, which has two diametrically opposite longest radii. By removing a circular portion from the upper right-hand quadrant, the centroid moves into the lower left-hand quadrant and there are no longer two identical longest radii. Removal of the same number of cells from the lower left-hand quadrant returns the centroid to its original location; there are two identical longest radii once more, although these no longer provide identical data.

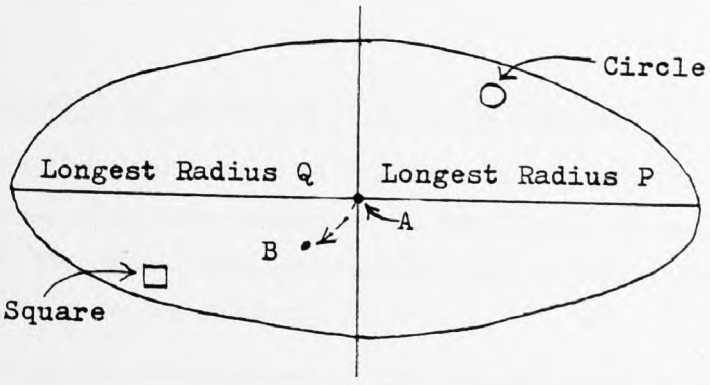


Figure 4.1: Ambiguity of longest radii

The sequence of operations in connection with Figure 4.1 is as follows:

(1) For the original ellipse, centroid is at point A and there are two longest radii, P and Q, which are indistinguishable.

(2) Remove circle from lamina in upper right quadrant; centroid migrates to point B, and there is only one longest radius.

(3) Then, remove square from lower left quadrant. Providing equations (4.E5) are satisfied, centroid will return to point A, and there will once more be two longest radii.

However, these are no longer indistinguishable; if features are examined in a particular sense (clockwise or anti-clockwise), the feature set measured from P differs from that measured from Q only by a cyclic shift, and unique determination of orientation is again possible.

Since sampled images comprise effectively a grid of cells, it is tempting to measure dimensions within the image as separations measured in cells. This kind of measure is, further, easy and efficient to compute in electronic hardware. However, severe errors may appear in the measurements when the distance being measured is not parallel to one of the grid axes. As has already been indicated in the previous paragraph, the apparent length of a straight line decreases by about 40% on being rotated 45 degrees to the axes. To reduce this error, therefore, the square root of the sum of the x-distance squared plus the y-distance squared must be obtained.

The effect would seem to be even more severe (and, moreover, less easy to predict quantitatively) when the shape boundary is curved. It was decided that the only way to assess the effects of quantisation on general shapes was by a simulation.

4.5 Optimal Edge Detection in a Two-Dimensional Silhouette Binary Image

Once the edge points of a silhouette binary image have been located and stored, then other measures of the image, which can be used as features of the shape, can be computed and processed accordingly to enable the image to be identified.

During the development of the present work an optimal method for extracting the edge points of a two-dimensional shape was introduced for a sequential processing which is generally desirable, particularly when an on-line application is contemplated with a real time scanning system as sensor.

Most systems proposed for extracting boundary points from binary images use two consecutive scans, and some use one scan of the image. It is shown here that we can have an optimal method by using three consecutive scans instead, where no boundary points are recorded more

than once and no extra distortion is introduced.

4.5.1 The problem of edge detection

The edge points of a binary image are sensed by locating 0 - 1 and 1 - 0 level transitions on the image.

We can have five alternative boundary conditions, namely:

(a) "No" edge point, when no boundary point is present.

(b) "Left" edge point, when 0 - 1 transition along the scan direction exists.

(c) "Right" edge point, when 1 - 0 transition along the scan direction exists.

(d) "Up" edge point, when 1 - 0 transition between the previous and present scan, respectively, is present.

(e) "Down" edge point, when 0 - 1 transition between the previous and present scan, respectively, is present.

Figure 4.2 illustrates the order with which the edge points of an image are normally extracted by a "real time" line scanning system, starting from edge point 1 and finishing at edge point 26 during 10 consecutive scans.

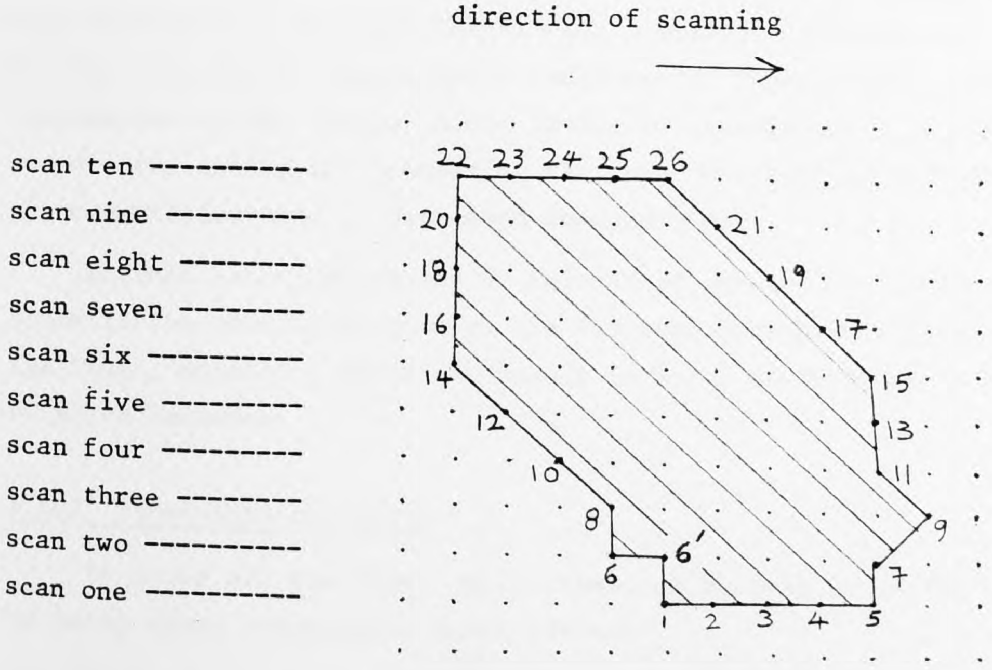


Figure 4.2: Edge detection in binary image

If we consider only the first three possibilities available, i.e. (a), (b) or (c), along the line of a scan, it can be seen from Figure 4.2 that by taking the gradient in only one direction results in gaps appearing in the boundary when this is parallel to the direction of scanning (i.e. points 2, 3, 4, 23, 24 and 25 are not detected), which leads, in turn, to errors in the computation of other measures on the shape, which are to be used as its features.

If, instead, we have all five possibilities available, and using a two-scan approach, by taking two consecutive scans of binarised signal and storing it, updated as each new scan becomes available, we can see from Figure 4.2 that edge points 1, 6, 10, 12 and 14 are recorded twice as "left" and "down" points, respectively, as well as points 15, 17, 19, 21 and 26 as "right" and "up" points. Point 5 is recorded twice as "down" and "right" point, and point 22 as a "left" and "up" point, respectively. Point 9 is recorded three times as a "right", a "down" and an "up" point!

Also, it should be noted that points like point 6' are not considered as boundary points, since they would have required to be compared with neighbouring points situated diagonally to them, something which is not allowed here since it produces mainly boundary points not on the boundary of the image, as can easily be seen from Figure 4.2.

So, although there are normally 26 boundary points in the shape, by using the two-scan method, we extract in effect 40 boundary points. This results in a boundary length much different from the real one, and in this case 53.8% bigger, which could become even more by changing the orientation of the shape. Also, having to use many more points in the computation during the processing for feature selection and extraction, the computation time is increased considerably.

Another error, which may be introduced, is to have extra approximations in the calculations which are involved during the processing of the image, something which can result in small distortions in the selected features.

4.5.2 Three-scan technique

To avoid all the above ambiguities, an optimal method is proposed by using three consecutive scans instead.

In Figure 4.3, A, B, C, D and E represent the values (0 or 1) of 5 cell points taken from a group of three scans. This kind of grouping

is repeated along the direction of scanning until the end of each scanning line. At every grouping the cell of interest, denoted by C, is compared with its neighbouring cells A, B, D and E simultaneously, in order to define its true identity as an "up", "down", "right", "left" or a "no" point on the boundary of an image being scanned.

Comparing Figures 4.2 and 4.3, it can easily be seen that now, with the three-scan method, only 26 boundary points are detected on the shape, as they really happen to be at the present angle of orientation.

With this way it is not possible to record boundary points more than once at any angle of orientation.

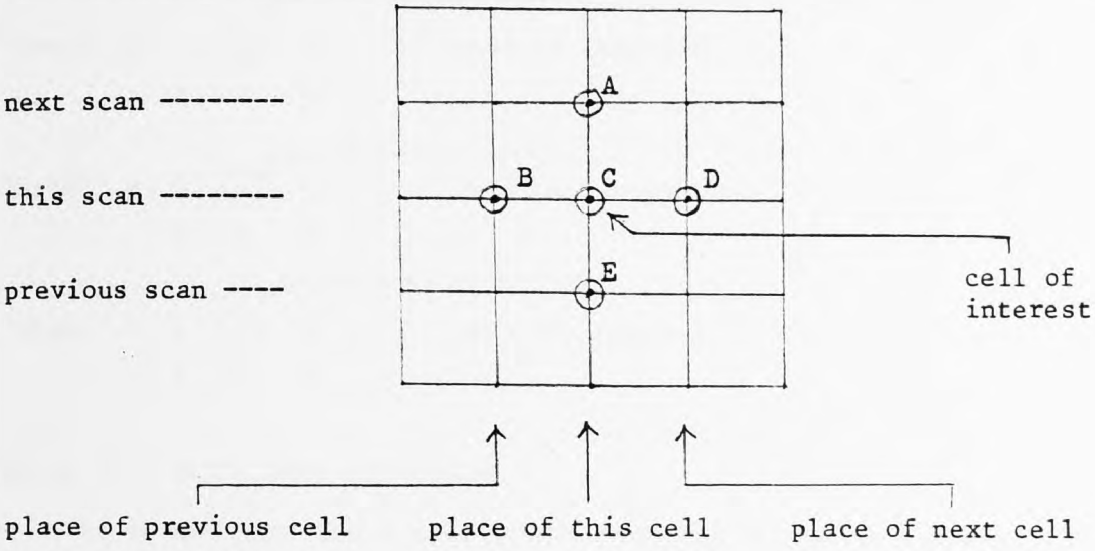


Figure 4.3: Three scan and 5 cell points technique

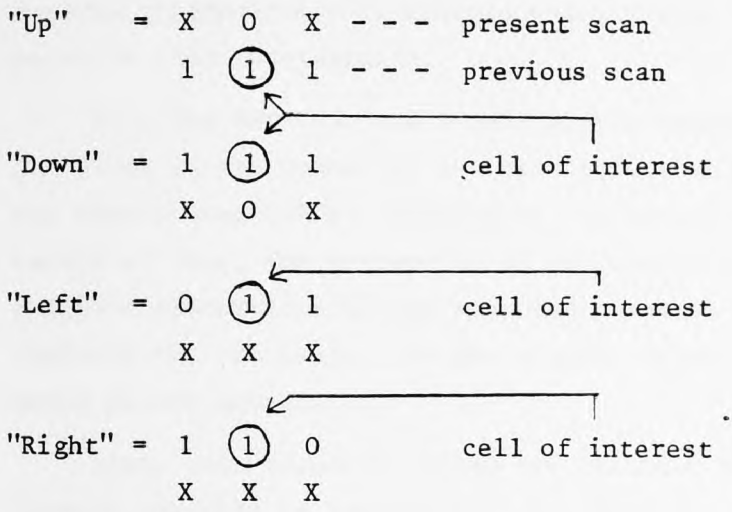
		A			0		
For an "Up point:	B	C	D	=	1	1	1
		E			1		
		A			1		
For a "Down" point:	B	C	D	=	1	1	1
		E			0		
		A			X		
For a "Left" point:	B	C	D	=	0	1	1
		E			X		
		A			X		
For a "Right" point:	B	C	D	=	1	1	0
		E			X		

where: X = don't care condition, i.e. "0" or "1".

Now, in order to obtain the same results (which we get with 5 cell points of the three-scan method), by using two consecutive scans we should need to select 6 cell points at a time, instead of 3 as it is commonly used and shown below:

"Up" = 0 0, "Down" = 1 1, "Left" = 0 1, "Right" = 1 0
1 0 0 1

and for the 6 cell points:



where X = don't care condition.

If we had selected 4 cell points instead, in the form shown below, with B the cell of interest, we would have had similar ambiguities as with the 3-cell case,

A B C
D

This may be seen from Figure 4.2, and more easily if we change slightly the angle of orientation of the image.

However, using the two-scan and 6 cell points method, we can see that the amount of computation involved is bigger than with the three-scan and 5 cell points technique. This is because:

(a) We have an extra cell point (6 instead of 5) which has to be compared accordingly and repeatedly for correct extraction of the boundary points, and

(b) If a boundary point is an "up" point, as a cell of interest, it is situated on the previous scan, instead of being on the present scan

like any other type of cell of interest, something which also requires extra computation.

Therefore, looking into the whole problem of edge detection, it is shown that with the one-scan method boundary points lying on lines parallel to the direction of scanning are not recorded, with the exception of the end points of these lines of the shape. This is unacceptable for any further processing of the binary image if accuracy is required. The above error may be reduced only with software by using several interpolation techniques which could, though, increase the computation time considerably.

With the two-scan and 3 cell points method we have normally about 50% error in the number of boundary points which are recorded, and it may exceed even 80% by changing the orientation of the image. As a result of this, the processing of the binary image becomes slower, with possible distortions in the computed measures which are to be used as features for the image. We get similar results also with the two-scan and 4 points method.

Also, when boundary points are recorded more than once, more storage capacity is required by the computer, in order to store all the x and y co-ordinates of the boundary points for the processing to be carried out. This is not always possible with small computer systems.

The methods using three scan and 5 points and using two scan and 6 points are superior to the methods previously summarised, and can give results of comparable precision and accuracy.

However, these two methods have different disadvantages in practical implementations.

For example, if the operation of boundary point extraction is to be implemented by using a specially constructed hardware system, with the three-scan method extra storage is required for the third scan, whereas the two-scan method would require one scan less of storage. This is a small disadvantage since the cost of memory per bit is constantly decreasing.

Conversely, the two-scan method requires extra computation in order to process the sixth point and also to remove the ambiguity introduced when processing "up" points. This would mean either extra computation time in software implementation, or extra hardware in hardware implementation for correcting the positioning of "up" points.

In spite of the overall equivalence of the two preferred methods, it is felt that the three-scan method is superior, because no ambiguities are introduced in acquiring an image, which allows for a more straightforward approach for subsequent processing. This would become particularly important where this processing took place in hardware.

In that case, in order to achieve the appropriate logic conditions with which the presence of edge points can be satisfactorily detected from the three consecutive scans, a bipolar field programmable logic array (FPLA) may be used, which can operate at the high speed of a few nsec, as the rest of the hardware circuitry usually requires.

4.6 Image Feature Extraction

When it is required to distinguish different silhouette shapes, less information than is required even to define boundaries will often suffice in practical applications. To select the reduced information necessary to perform some specific task a process of feature extraction is necessary. It is often important that the feature be invariant with respect to the position and orientation of the silhouette shape.

The feature set used here to characterise the various shapes is shown schematically in Figure 4.4 below:

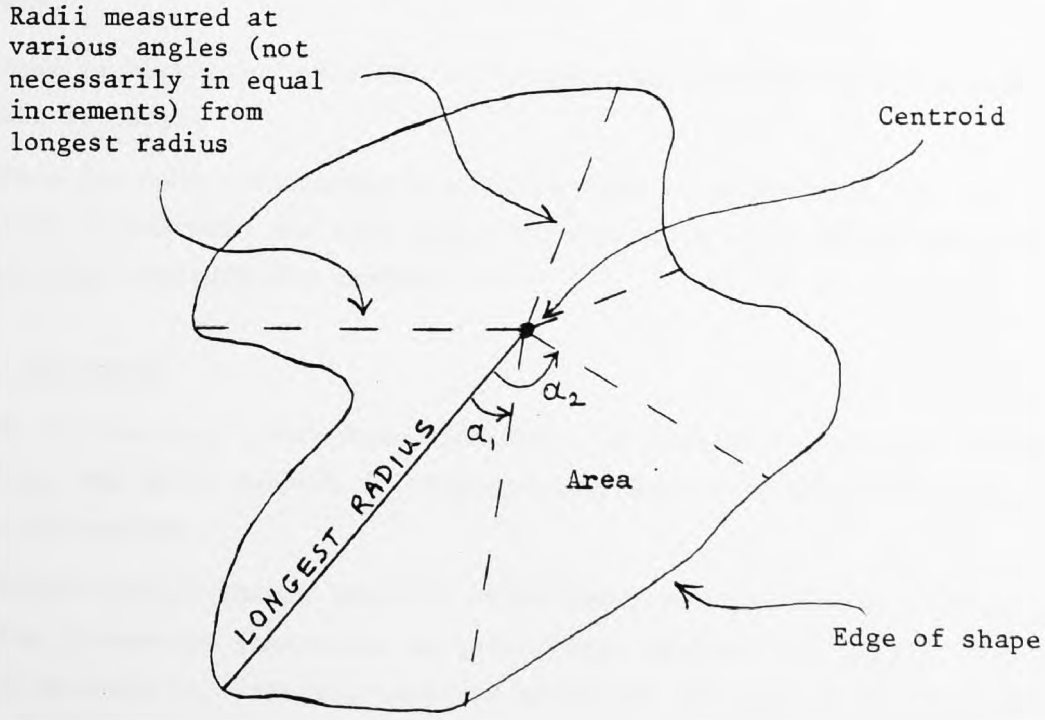


Figure 4.4: Features of a silhouette shape

4.6.1 Centroid

The first stage in the definition of a set of features is the selection of a datum point within the silhouette. This datum should satisfy the following criteria:

(a) Ease of computation,

(b) Uniqueness for a given shape,

(c) Insensitivity to errors in the location of the silhouette outline.

The centroid of the silhouette satisfies these criteria.

The co-ordinates of the centroid may be computed easily by summing and averaging the location of the cells inside the boundary of the silhouette; its position is clearly unique.

Since errors in deciding which cells should be incorporated into the silhouette occur only at the boundary, the centroid is relatively insensitive to this kind of error. This insensitivity is not obviously maintained for very thin silhouettes.

4.6.2 Area

The area of the shape is measured by counting the number of samples above a threshold. In this way it is easy to measure the area with great accuracy.

Area is virtually invariant to orientation provided that the number of cells is large.

From the same information with which area is determined, the co-ordinates of centroid are also computed, something which makes the processing time considerably shorter.

4.6.3 Perimeter

By determining a threshold difference in intensity for cells inside the shape and cells outside it, the boundary points of the silhouette can be determined.

Since sampled images comprise effectively a grid of cells, it is tempting to measure dimensions as separations measured in cells. This kind of measure is, further, easy and efficient to compute in electronic hardware. However, severe errors may appear in the measurements when the distance being measured is not parallel to one of the grid axes.

Thus, as was indicated in section 4.3, the apparent length of a straight line decreases by about 40% on being rotated 45 degrees to the axes. The square root of (separation measured in cells in the x-direction squared plus separation measured in cells in the y-direction squared) as the true dimension must be used; something which is very excessive in computation for every pair of boundary points.

Therefore, the length of the perimeter of a silhouette is not used as member in the feature set, but the location only of all boundary points is extracted (by using the optimal method described in section 4.5), simply to be used for the determination of lengths of radii from the centroid to the boundary of the silhouette shape.

4.6.4 Longest radius

The longest radius, measured from the datum (centroid) to the boundary, can be used as a reference; the angle between this radius and the grid axis would then specify the orientation of the silhouette.

Since the calculation of the longest radius is accurate enough, being independent of orientation to a very high degree, simply because of the unique relation between the centroid and the boundary, it can be used as an important feature of the silhouette shape.

In the case of the existence of more than one equal longest radius in the shape, a selection technique can easily be adopted, such as the one nearer to the smaller radius in the shape, or the middle one if all the longest radii are located on a small arc of the boundary, etc.

4.6.5 Radii at pre-specified angle intervals

As features, one can take, except the area and the longest radius of the shape, radii measured also from the centroid to the boundary, at pre-specified angle intervals, starting from the reference position of the longest radius and proceeding in an anti-clockwise way, say.

The number and spacing appropriate for the radii will evidently depend on the particular problem.

The information required to specify a shape in this way may be estimated as follows:

Considering a shape on a grid 1024 cells square; 20 bits are required to locate the centroid, then 9 to specify the length of each radius. Individual radii can be identified by their order in a list,

and by their known angular separation. Thus, a shape even with a curved boundary could be specified using about 128 bits for a 30-degree, say, radius interval.

This is in contrast to the 2,000 or so bits which would be required if the whole boundary were retained, even if efficiently encoded.

If the shape were complex in that each radius crossed the boundary several times, or there were holes in it, slightly more information would be required.

4.6.6 Orientation

The orientation of a shape is defined as the anti-clockwise angle between its longest radius and the axis of the grid. Two features determine the accuracy with which this may be measured, and both arise essentially because the shape is quantised on to a grid. They are:

(a) The ease with which the longest radius may be identified within a shape.

For a shape which is nearly circular, the longest radius will be very difficult to identify. If, on the other hand, the longest radius encounters the boundary where the radius of curvature is small (e.g. at a sharp point) it may be located with considerable precision.

It is very difficult to calculate the effect of curvature on precision of longest radius identification. Thus, the relationship will be investigated using simulation in the next chapter.

(b) The precision with which the direction of the longest radius may be measured, given that it has been identified correctly. This is easily determined using calculation, as is shown below:

The angle θ between the radius and the x-axis is given by:

$$\theta = \arctan (x/y) \qquad (1)$$

where x and y are the co-ordinates measured in cells of the cell at which the longest radius intersects the boundary, with the centroid as centre of the axes.

Errors in measuring θ arise from errors in measuring both x and y, which may occur simultaneously.

Thus, we have for the total error:

$$d\theta = \partial\theta/\partial x \cdot dx + \partial\theta/\partial y \cdot dy \qquad (2)$$

and on substituting in values of $\partial\theta/\partial x$ and $\partial\theta/\partial y$ obtained by differentiating (1) above, we obtain:

$$d\theta = y/(x^2 + y^2) \cdot dx - x/(x^2 + y^2) \cdot dy \qquad (3)$$

Here, dx and dy are the errors in measuring x and y, respectively, and $d\theta$ is the resulting error in the measurement of θ. How the errors combine evidently depends on the values of x and y, for given dx and dy.

We may investigate extreme cases, for which dx and dy are the same and are equal to M, say.

In the first, x and y are roughly equal and have the same value, denoted K. This implies that the longest radius is at about 45 degrees to the cell axis. On substituting and noting that the direction of errors consequent upon increases in x and y is opposite, we then have:

$$d\theta = M/K.$$

The error now depends only on the relative sizes of θ and M.

As an example, if M = 0.3 mm, to obtain $d\theta$ of one degree, which is 0.0174 rad, we need K = 0.3/0.0174 = 17.2 mm.

In the second, x (or, equivalently, y) is zero; it does not matter which, since the problem is symmetric. Also, y (or x) = K. In this case, after substituting in (3), we again obtain $d\theta = M/K$, so that once again for $d\theta$ of one degree, K = 17.2 mm!

But the situation at intermediate positions remains undetermined, although it is unlikely to be much worse than at the position for which a value has been computed.

It is therefore shown that a minimum longest radius of 17.2 mm is required for one degree angular accuracy and a 0.3 mm resolution cell.

It is further shown that the resolution obtainable is directly proportional to the length of the longest line, as is the angular resolution required.

Any desired degree of precision in orientation measurement may be obtained simply by making the scanning grid finer. However, this makes the processor slower and more expensive. The objective of the processing is to insert a mark at a position specified with respect to the centroid to within about 0.005 inches.

Thus, considerations such as overall shape dimension, in addition to accuracy of angle measurement, must be taken into account.

4.7 Feature Selection

Although the quantity of information required for a silhouette pattern completely is quite large (it can reach tens of thousands of bits), it has heen shown already here that much less information is sufficient to distinguish the shapes from one another. In general, only some hundreds of bits are necessary.

Further, simple measures, such as the area of the shape and the lengths of radii measured at particular angles from the longest radius, were adequate as features for recognition, since they contain the necessary information.

4.7.1 Tolerance

For a silhouette shape characterised by its features, inevitably errors occur in the actual measurement of the features. Further, patterns nominally within a given class tend to differ slightly due, for example, to knife wear and shrinkage.

Thus, tolerances must be imposed on the measurements; a measurement must be regarded as falling into a cell whose size is equal to the tolerance on the measure. Thus, each feature comprises a sequence of similar cells of finite size; it is preferable for each feature to be a small number.

Therefore, the size of the tolerance cells is selected basically to accommodate variations in measured parameters due to cellular quantisation in the scanner and to manufacturing imperfections.

4.7.2 Procedure

As the pattern is scanned, boundary points are located and stored. As soon as scanning is complete, sensed when a scan is encountered containing no boundary points, the centroid is located, and the distances of all boundary points from the centroid and their angle from the longest radius computed.

Then, the lengths of radii at specified angles to the longest radius

are extracted and stored, to form a candidate feature set.

When several radii exist for a particular angle, the longest is selected as the feature to eliminate ambiguities caused, for example, by holes in the pattern or re-entrants, which cause a radius to cut the boundary at several points.

Then, the features chosen, together with auxiliary information such as tolerances and the angles at which radii are to be measured, are stored appropriately, and a new pattern can then be accepted if it is required.

4.8 Learning and Training

After a pattern has been scanned and its features have been selected and stored in the provided memory facilities, a class name is added to it, in order to distinguish it from any other pattern of different shape.

The above procedure can be repeated as many times as all the different patterns have been scanned, so all the different classes are stored in the memory.

4.8.1 Learning

Generally, it is not possible to predict which features will prove useful for separating a particular set of pattern classes. Therefore, in the "learn" stage of the recognition process, many more features must be acquired for each pattern than are ultimately going to be used for recognition.

A sufficient subset can then be selected for use in on-line identification.

The feature data present after the "learn" process has examined all patterns to be identified comprise a table whose I^{th} row contains measured values of a particular feature (e.g. area, or length of longest radius) for all classes, and whose J^{th} column contains all feature measurements for a particular class.

These data are transferred via the backing store to the feature selection procedure; this first cellularises the measures (using tolerances), then selects the sufficient feature set.

4.8.2 Training

To make the on-line inspection as fast as possible, only the minimum number of features necessary should be retained for recognition.

However, procedures for choosing such a minimum subset are cumbersome computationally, and a method which is somewhat sub-optimal has to be used instead.

4.8.2.1 Separation of classes

Because the number of cells per feature can be generally large, and differs from feature to feature, sophisticated mathematical procedures (e.g. those obtained from algebraic coding theory) are unprofitable in this application. Instead, a heuristic approach can be used.

The features are examined in the order in which they appear in the feature table, i.e. the order in which they were gathered.

A feature set provides unique identification only if all columns differ mutually in at least one entry, so that checks can be made of the columns against one another to determine if they do indeed differ in at least one entry.

If not, the next feature in the table is added and so on, until a usable feature set is obtained.

4.8.2.2 Minimum distance

In practice, erroneous measurement may occur, e.g. for small patterns, for which at certain orientations particular features may simply not exist, because the grid of cells is not fine enough.

The solution in this case is to incorporate some controlled redundancy into the feature set, i.e. to require that all columns in the feature set chosen differ mutually in at least, say, two entries instead of one.

In the terminology of coding theory, the features would be separated mutually by a minimum distance of 2.

This modification allows the recogniser to tolerate all occurrences of one error occurring in a feature.

In general, the distance between two vectors containing the same number of entries is the number of positions in which corresponding entries differ. For example, the vectors:

12, 17, 39, 4 and
12, 39, 33, 4

differ in positions (2) and (3), and are therefore separated by a distance 2.

In this shape recognition problem, vectors of numbers representing measurements are used to characterise the various shapes. An unknown shape is then recognised by comparing its vector of measurements with those stored for each class of shape, one vector for each class, entry by entry, until an exact fit is found. The stored prototype vectors must evidently differ mutually in at least one position (i.e. be separated by a minimum distance one), otherwise unique identification will be impossible.

For economy, it is desirable that the vectors contain the smallest number of measures which ensure they are unique. In this case, the vectors will comprise the sufficient set of minimum distance unity. The distances between some members of the set will, in general, be more than one.

4.9 Automatic Inspection

In the automatic inspection phase, the system has to scan patterns of stored form but unknown identity and to identify them by comparing the vector of features measured for the pattern with those stored until a sufficient fit is found. When a fit is found, the identity assigned to the pattern is displayed, otherwise a "pattern not identified" message is displayed.

The experimenter can specify interactively what minimum distance is accepted at any specific application, and what tolerances are to be used. Other human interaction should not be necessary.

4.9.1 Correct identification

If, after the tolerances for the different measures have been specified as well as the desired minimum distance during the inspection phase, a correct fit is recorded between the vector of the feature set of the unknown pattern and the equivalent vector of the feature set of the

corresponding class, then we have a correct identification.

4.9.2 Mis-classification and rejection

If some of the measurements yield erroneous results, the resulting vector will either not fit any of the stored prototypes (classes), or will fit the wrong prototype.

In the first case, the machine will not be able to identify the pattern and will reject it. That is, it will signal that it cannot make a classification.

In the second case, the machine will assign the pattern to the wrong class, termed substitution, or, in other words, mis-classification.

The consequences of substitution are generally more objectionable than those of rejection. For example, machinery in a later process may be damaged by an incorrect shape, although a rejected shape could be re-inserted and examined again.

For example, suppose the vectors 12, 17, 39, 4 and 12, 18, 39, 4 form a simple feature set comprising two pattern classes only, class "a" and class "b". In this set, measures (1), (3), (4) make no contribution to classification, and are said to be redundant. The set is therefore necessary rather than sufficient. If the second measure from an "a" class pattern were corrupted to give 18 instead of 17, then the resulting vector would be identical to that for patterns from class "b", and a substitution would result. If, on the other hand, the second measurement had been corrupted from 17 to 16, the resulting vector would have matched neither stored prototype and the machine would have had to reject the pattern.

We could improve the performance of the classifier by arranging that the measurement vectors differ mutually in more than one position. For example, our class "a" might have the vector 12, 33, 34, 17 and our "b" class the vector 12, 34, 33, 17. In this case, corruption of a single measurement could not cause a vector measured from an "a" class pattern to match that for the "b" class, and vice-versa. However, a vector so corrupted could not be identified correctly. It would be rejected. Thus, by using features differing mutually in two entries (i.e. separated by a minimum distance two), we can eliminate the possibility of a substitution occurring as a consequence of a single erroneous measurement.

4.10 Concluding Remarks

The penalty for providing immunity to errors by using the minimum distance concept lies in the extra data which must be stored and compared for each pattern class to obtain a sufficient feature set.

The number of extra features which must be included to ensure a given minimum distance between pattern classes can be determined only by experiment, since it depends on the particular set of patterns being examined.

Some idea as to the degree of error protection provided by a given minimum distance may be obtained by calculation.

Assume that the probability of any particular measure being corrupted is p, and that the measures are completely independent so far as probability of error is concerned. Then, the probability P(K) that K or more errors will occur in any vector containing N of the measures is given by a cumulative binomial probability distribution.

Alternatively, we could use a second approach in which vectors are assigned to a class providing a prototype exists from which they differ in no more than (K/2 - 1) entries. If they differ in more, they are rejected. In this case, on the other hand, there will be fewer substitutions but far more rejections.

Although the substitution performance would be worse, it is impossible to determine by how much, except by experiment.

CHAPTER 5: SIMULATION STUDY

5.0 INTRODUCTION

This chapter describes the investigation which was carried out using a sample of shapes, both artificially generated and "real-life" patterns. These were processed in a CDC 7600 computer with a microfilm output.

The objectives of this study were:

(a) To devise and evaluate suitable methods for extracting the boundary description of a shape, by simulating the operation of a "real-time" line scanning system.

(b) To determine the spatial resolution required to achieve the accuracy in feature measurement necessary to distinguish closely similar patterns. The need for the scanning head to accommodate a wide range of pattern size without mechanical adjustment was taken into account.

(c) To determine whether there were any practical difficulties which were not immediately apparent, and to provide solutions for them.

Hence, the simulation study comprised two parts, which were:

(i) An investigation using artificially generated shapes. The ellipse and the cardioid were used, because they can be easily defined mathematically, and also because they provide complexities along their boundaries worth investigating, such as curves and multiple crossings or re-entrants.

(ii) An investigation using real patterns, not necessarily defined mathematically, and supplied by the sponsoring company of this work. During this investigation six shapes were used as data, in two selections as follows:

  (1) The first selection comprised the largest version of the most complex pattern available, the opposite hand version of this pattern and the half size smaller version of the same pattern.

  (2) The second selection contained the smallest pattern available, the opposite hand version of this pattern

and the similar pattern one next size larger.

The objective of the second investigation, with its two selections, was to demonstrate that the processing could distinguish patterns differing by one next available size at both extremes of the size range, and also between left- and right-hand patterns which differed only in just being mirror images of one another at the extremes of the size range.

This distinction must be achieved without adjustment (i.e. change of magnification) of the scanning head.

5.1 Investigation using Artificially Generated Shapes

In pursuit of the present task a simulation study has been performed as follows:

(1) Two artificially generated shapes (i.e. mathematically defined) were used, an ellipse and a cardioid.

(2) Computer programs have been written which

  (a) will generate the shapes of the above objects,

  (b) will extract outlines given a "grey-scale" silhouette, and

  (c) will measure features given this outline, as follows:

    (i) area,

    (ii) perimeter.

    (iii) centre of the centroid,

    (iv) length of the Maximum Distance from centroid to the boundary and its Angle with respect to the x-axis, and

    (v) lengths of other Radii from centroid to the boundary at specified Angle Intervals.

(3) Microfilm subroutines have been used to plot the shapes and features and to demonstrate that these have worked correctly.

(4) The problem of multiple crossing of the boundary by a scan line or by a radius has been investigated, as well as the problem of the presence of holes inside a silhouette shape.

(5) The effect on the accuracy by the changes of the following has been investigated in simulation:

(a) size,

(b) eccentricity (for ellipse), and

(c) orientation.

(6) The computed results have been compared with the ones taken by formulae calculations.

The mathematical analysis for the model solutions with simulation, using a CDC 7600 computer, is carried out by impressing the silhouette shapes on a grid of cells.

A masking operation has been used, which determines threshold differences in intensity for cells located inside the silhouette shape and cells outside it.

5.1.1 Ellipse

Ellipses were generated, of varying size (expressed by the length of the semi-major axis), thinness (expressed by eccentricity), and lying at varying angles to the grid axes.

The computer program simulated the boundary detection process, located the boundary cells, then measured, in turn, area, boundary length and location of centroid.

The program next located and measured the longest radius and then measured the lengths of a number of radii (making them ten in all for this experiment) lying at specified angles to the longest radius.

In Figure 5.1, are shown the parameters used for this investigation, which were:

(a) the semi-major axis, NA, measured in cells,

(b) the angle of rotation, PHI, measured in radians, and

(c) the dimensionless eccentricity, EC, not shown in Figure 5.1.

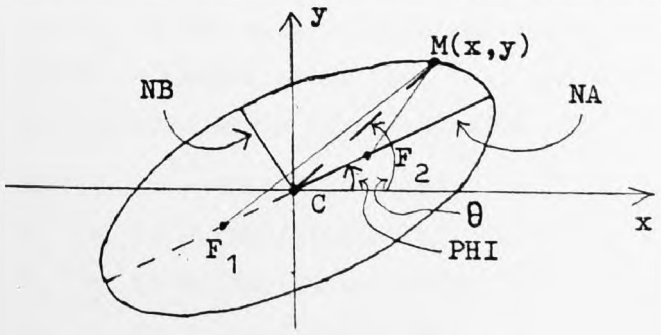


Figure 5.1: Ellipse

From ellipse formulae we have:

$$x = NA\cos\theta \text{ at any point M} \qquad (5.E1)$$

$$x^2/NA^2 + y^2/NB^2 = 1 \qquad (5.E2)$$

$$NB^2/NA^2 = 1 - EC^2 \qquad (5.E3)$$

where NB is the semi-minor axis of the ellipse, and F_1 and F_2 are the two foci.

5.1.1.1 Masking operation

The masking operation used was based on producing a two-dimensional array, A(I, J), of an appropriate size, implying a plane grid divided into I by J squares on which the silhouette shape is to be impressed, by giving values to the array either equal to a fixed threshold level each time a point inside the silhouette is encountered, or, otherwise, equal to zero.

By using the property of an ellipse by which the sum of the distances MF_1 and MF_2 from any point M on the perimeter to the two foci F_1 and F_2 of the ellipse (see Figure 5.1) is less than or equal to the size of the major axis, 2 x NA, and hence from any point inside the silhouette we cannot have a corresponding sum greater than 2 x NA, points inside the silhouette of the ellipse were determined as follows:

(1) The values for size, NA, eccentricity, EC (between 0 and 1), and orientation, PHI, were selected as desired.

(2) The co-ordinates (Xc, Yc) of a reference point inside the grid were chosen. In this case the centre of the ellipse has been selected, which coincides also with the centroid C, because of the symmetry of the shape.

(3) The co-ordinates (X_1, Y_1) and (X_2, Y_2) of the two foci F_1 and F_2 of the ellipse were determined with respect to the above reference point (Xc, Yc), and also in relation to the desired angle of orientation PHI. These were:

$$X_1 = Xc - (NA)(EC)\cos(PHI)$$
$$Y_1 = Yc - (NA)(EC)\sin(PHI)$$
$$X_2 = Xc + (NA)(EC)\cos(PHI)$$
$$Y_2 = Yc + (NA)(EC)\sin(PHI)$$

(4) By using all values of I (starting from 1) for each value of J (starting also from 1) in turn for the whole array, the computation of the sum, S, of the square roots of $(D_1^2 + D_3^2)$ and $(D_2^2 + D_4^2)$ is performed each time, where:

$$D_1 = J - X_1 \qquad D_2 = J - X_2$$

$$D_3 = I - Y_1 \qquad D_4 = I - Y_2$$

(5) The above sum, S, is then compared each time with the size of the major axis 2 x NA of the ellipse.

When S is less than 2 x NA, a point inside the silhouette is encountered and the array A(I, J) is given the chosen threshold value, TH (say), for every such point. In any other case the array A(I, J) is given the value zero.

In this way the silhouette shape of an ellipse is artificially generated on a rectangular grid.

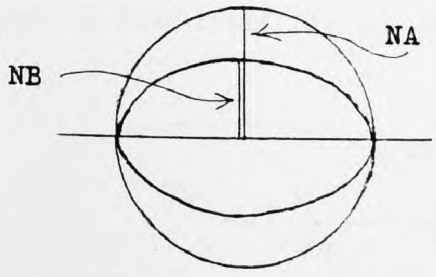
5.1.1.2 Area, centroid, perimeter and radii

(a) The Area is computed during the masking operation by counting cells falling within the boundary, and equal to N.

In other words, every time the array A(I, J) is taking the value TH, a special "counter" in the program is incremented by one until all the elements of the array are evaluated.

For the purpose of comparing the computed value of the area with the one given by calculation, we may deduce the formula for the area of the ellipse as follows:

Since the area of a circle of radius NA is equal to $\pi(NA)^2$, where $\pi = 3.14159$, then by project in ratio NB/NA all lengths at one direction, as is shown in the diagram below, we get for the ellipse:



$$\text{area} = \pi(NA)^2 \times NB/NA = \pi(NA)(NB),$$

but from equation (5.E3) we have:

$$NB = NA \times (1 - EC^2)^{\frac{1}{2}},$$

which gives the area of ellipse by calculation to be equal to:

$$A = \pi(NA)^2 \times (1 - EC^2)^{\frac{1}{2}}.$$

We can see that when EC = 1, then A = 0, something which does not necessarily occur during computation of the area, due to computational approximations at various orientations.

(b) The Centroid is found by locating its co-ordinates, which are obtained using the formulae:

$$Xc = 1/N \sum_{k=1}^{N} X_k \quad \text{and} \quad Yc = 1/N \sum_{k=1}^{N} Y_k$$

in which X_k and Y_k are the co-ordinates of the k^{th} filled cell and N is the total number of cells within the shape.

Since the ellipse is a symmetrical shape around both axes, it should be expected every time that the centroid found by the above technique (i.e. by using separate "adders" for the elements I and J of the array A(I, J) whenever it has a value TH) will coincide with the reference point used originally for the ellipse.

(c) The Perimeter (or boundary points) is sensed by locating 0-TH and TH-0 level transitions.

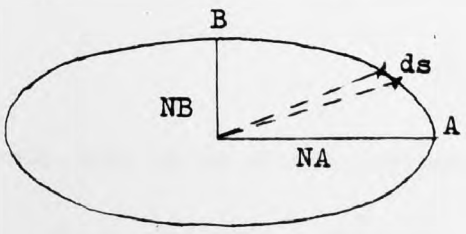
This is achieved by comparing intensities of neighbouring values of the array A(I, J).

A special "counter" is used in the program, which is incremented every time a transition is encountered until all the values of the array have been compared, giving at the end the total number of points located on the boundary.

Approximations in the above computations may also occur, as was indicated in the case of the area, due to the variations in the orientation.

For the purpose of comparing the computed value of the perimeter with a calculated one, only an approximation may be deduced for such a calculated value, as follows:

From the diagram shown below, the perimeter P of the ellipse is equal to four times the arc AB, i.e. $P = 4(\widehat{AB})$



$$= 4 \int_0^{NA} ds = 4 \int_0^{NA} (ds/dx)\,dx,$$

but $(ds)^2 = (dx)^2 + (dy)^2$

and $ds/dx = \left(1 + (dy/dx)^2\right)^{\frac{1}{2}}$,

also, from equation (5.E2), we have:

$dy/dx = -(x/NA^2)/(y/NB^2)$,

thus, perimeter $P = 4 \int_0^{NA} \left(1 + (x^2/NA^4)/(y^2/NB^4)\right) dx$,

and, because of equation (5.E2), we get:

$$P = 4 \int_0^{NA} \left[(NA^2 - x^2/NA^2 + (NB^2/NA^2)x^2)/(NA^2 - x^2)\right]^{\frac{1}{2}} dx$$

Now, from equation (5.E1), we get for the perimeter:

$$P = 4 \int_0^{\pi/2} (NA^2\cos^2\theta + NB^2\sin^2\theta)d\theta,$$

and, because of equation (5.E3), we have:

$$P = 4(NA) \int_0^{\pi/2} (1 - EC^2\sin^2\theta)^{\frac{1}{2}} d\theta.$$

But, from Walli's formulae, we have:

(i) $$\int_0^{\pi/2} \sin^n\theta \, d\theta = (\pi/2)\left[(n - 1)(n - 3) \ldots 1\right]/\left[n(n - 2) \ldots 2\right],$$

for n = even, and

(ii) $$\int_0^{\pi/2} \sin^n\theta \, d\theta = \left[(n - 1)(n - 3) \ldots 2\right]/\left[n(n - 2) \ldots 3\right],$$

for n = odd.

Therefore, the perimeter is equal to:

$$P = 4(NA) \int_0^{\pi/2} \left[1 - (1/2)EC^2\sin^2\theta + \left[1/2(-1/2)\right]/(2!)EC^4\sin^4\theta + \left[1/2\ (-1/2)(-3/2)\right]/(3!)EC^6\sin^6\theta + \ldots\right] d\theta$$

$$= 4(NA)\left[(\pi/2) - (1/2)EC^2(1/2)(\pi/2) - (1/8)EC^4(3/4)(1/2)(\pi/2) + (1/16)EC^6(5/6)(3/4)(1/2)(\pi/2) - (5/128)EC^8 \rightarrow \rightarrow (7/8)(5/6)(3/4)(1/2)(\pi/2) + \ldots\right].$$

Hence, the value of the perimeter of the ellipse may be approximated to:

$$P = 2\pi(NA)(1 - EC^2/2)^{\frac{1}{2}}$$

The above calculated value of the perimeter gives also a zero value when the eccentricity is equal to one.

So, the case of eccentricity when EC = 1, although through the computation this gives us values different from zero, which never agrees with the calculated values for both cases of area and perimeter, it has been considered as an extreme case and has not been taken seriously during the simulation experiments.

(d) The Radii are considered after (a), (b) and (c) have been completed for each silhouette shape.

The distance R_b between each boundary point and the centroid is computed, using the formula:

$$R_b = \left[(X_k - X_c)^2 + (Y_k - Y_c)^2\right]^{\frac{1}{2}} .$$

The maximum value of R_b, R_{max}, is then determined and used as a datum for determining the position of another number of radii selected at specified angle intervals around the boundary, to be used as features in the identification process.

The computed value of the maximum radius of the ellipse is expected to coincide with the value of the semi-major axis NA. Also, its angle should be the same as the angle of orientation PHI.

Because of the symmetry of the shape of the ellipse, two equal radii exist as longest radii at the two opposite sides of the major axis.

For this reason, it has been arranged that only the one which is first encountered during the computation is to be selected as the longest radius.

Comparison of its value can easily be performed with the value of the semi-major axis.

The rest of the selected radii are arranged so as to be situated at equal angle intervals, each $2(\pi)/10$ radians, around the boundary.

So, there are ten radii for each silhouette shape which are dividing the ellipse symmetrically.

In this way it can easily be checked how accurate their relation remains and particularly whenever the orientation varies.

For this purpose microfilm outputs have been produced in order to examine the behaviour of the radii during any change in size,

orientation and eccentricity.

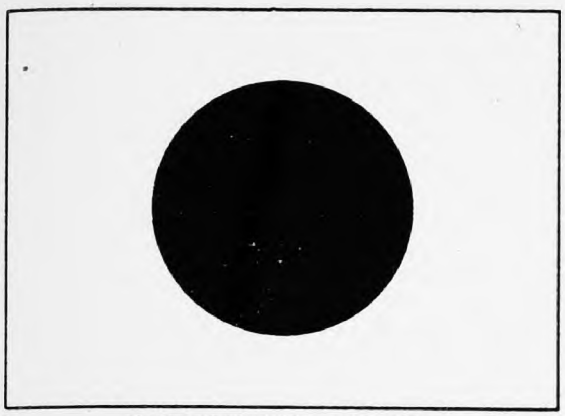
5.1.1.3 Simulation results

Typical images used in this simulation, plotted on microfilms, are shown in Figures 5.2(1) to 5.2(35).

In Figures 5.2(1), 5.2(12), 5.2(13), 5.2(24) and 5.2(34), particularly, microfilm pictures are shown of ellipse silhouettes at different sizes, orientations and eccentricity values, where all the cells inside the silhouettes are recorded and stored as they have been generated originally by the computer using a masking operation.

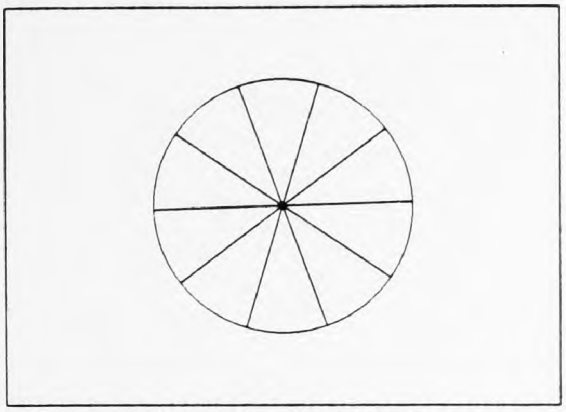
The above operation was performed for each silhouette shape before the actual process of extracting the boundary cells was carried out, as is shown on the rest of the microfilm pictures, also for different sizes, orientations and eccentricity values.

By examining the relations between the radii around the boundaries of the silhouettes, at all different settings of values for their parameters (NA, PHI and EC), it is apparent from the microfilm outputs that the symmetry has remained in every case. Their behaviour, so far as accuracy is concerned, is examined in some detail by the behaviour of the longest radius at different conditions.



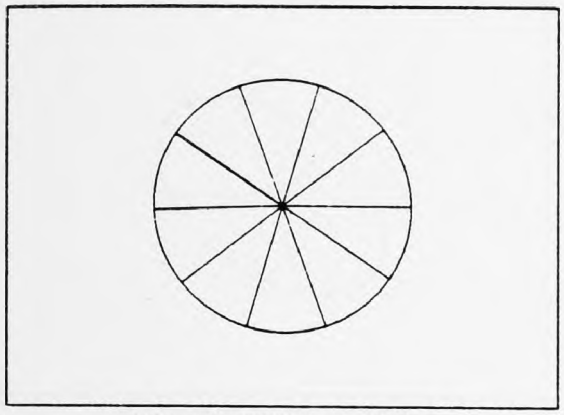
SILHOUFTTE IMAGE

Figure 5.2(1)
NA = 200, PHI = 2.5, EC = 0.1



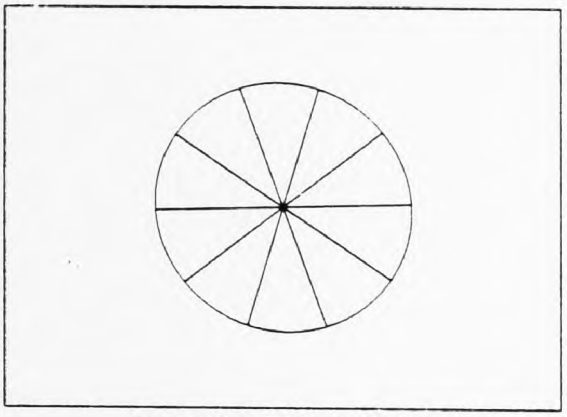
SILHOUETTE IMAGE

Figure 5.2(2)
NA = 200, PHI = 2.5, EC = 0.1



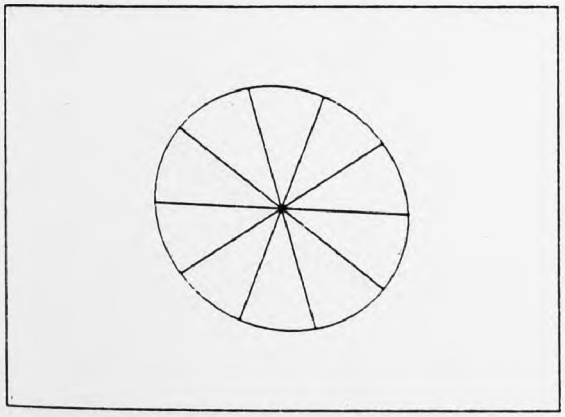
SILHOUETTE IMAGE

Figure 5.2(3)
NA = 200, PHI = 2.5, EC = 0.2



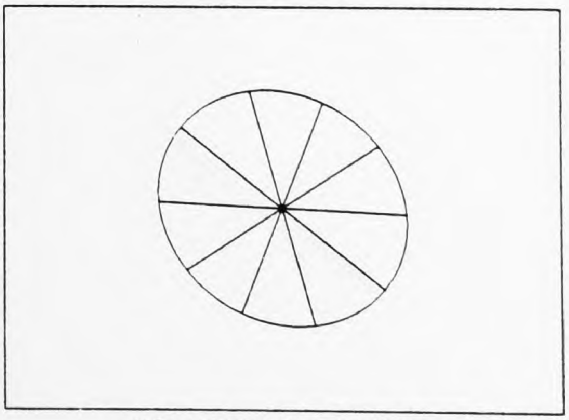
SILHOUETTE IMAGE

Figure 5.2(4)
NA = 200, PHI = 2.5, EC = 0.3



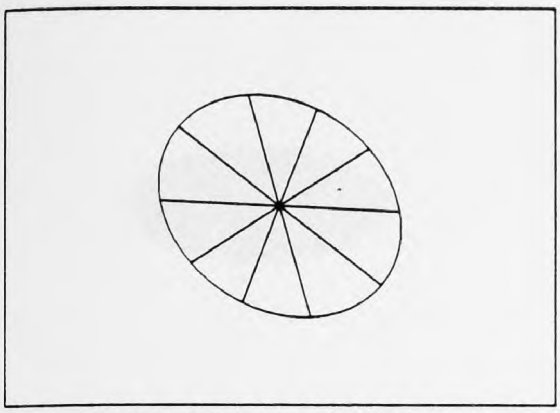
SILHOUETTE IMAGE

Figure 5.2(5)
NA = 200, PHI = 2.5, EC = 0.4



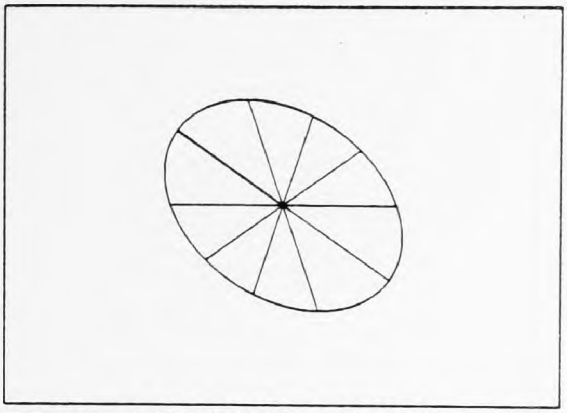
SILHOUETTE IMAGE

Figure 5.2(6)
NA = 200, PHI = 2.5, EC = 0.5



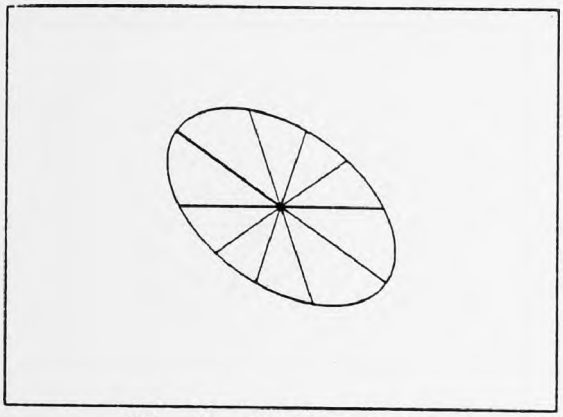
SILHOUETTE IMAGE

Figure 5.2(7)
NA = 200, PHI = 2.5, EC = 0.6



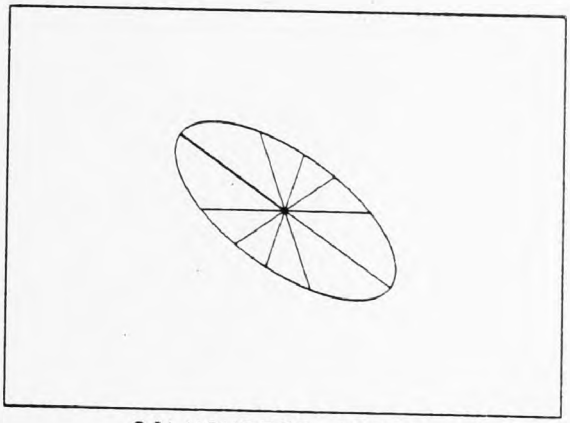
SILHOUETTE IMAGE

Figure 5.2(8)
NA = 200, PHI = 2.5, EC = 0.7



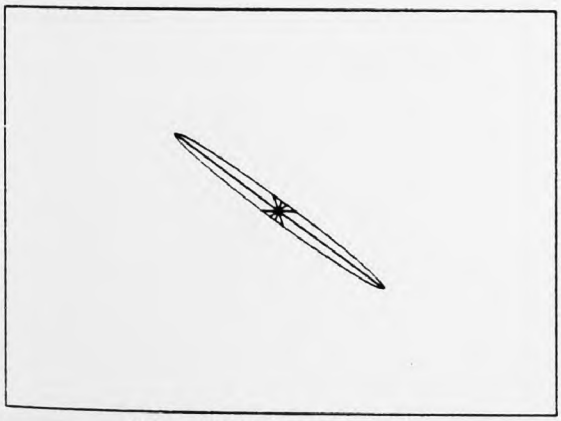
SILHOUETTE IMAGE

Figure 5.2(9)
NA = 200, PHI = 2.5, EC = 0.8



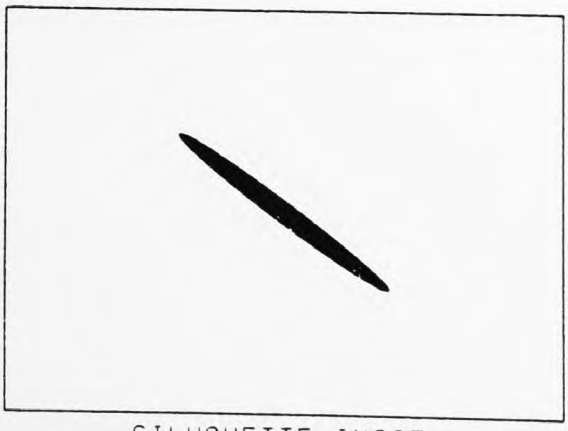
SILHOUETTE IMAGE

Figure 5.2(10)
NA = 200, PHI = 2.5, EC = 0.9



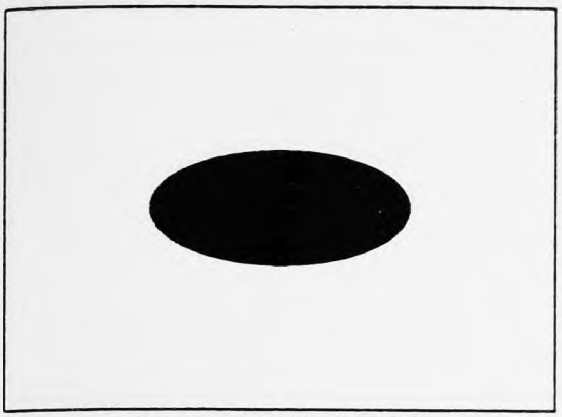
SILHOUETTE IMAGE

Figure 5.2(11)
NA = 200, PHI = 2.5, EC = 1.0



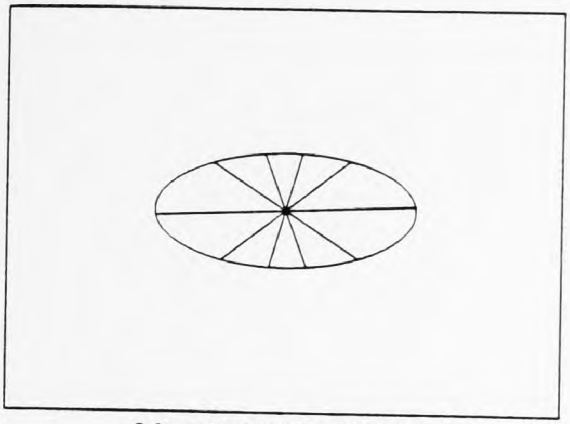
SILHOUETTE IMAGE

Figure 5.2(12)
NA = 200, PHI = 2.5, EC = 1.0



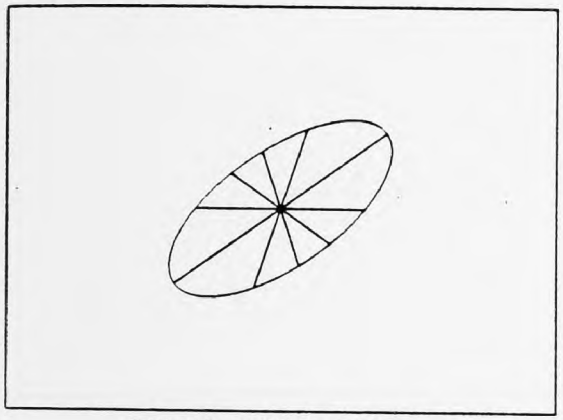
SILHOUETTE IMAGE

Figure 5.2(13)
NA = 200, PHI = 0.0, EC = 0.90



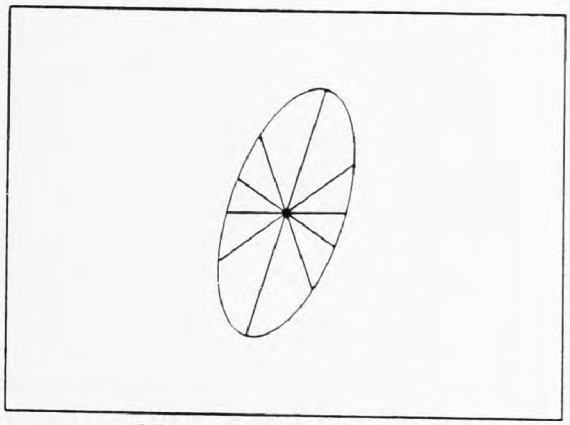
SILHOUETTE IMAGE

Figure 5.2(14)
NA = 200, PHI = 0.0, EC = 0.90



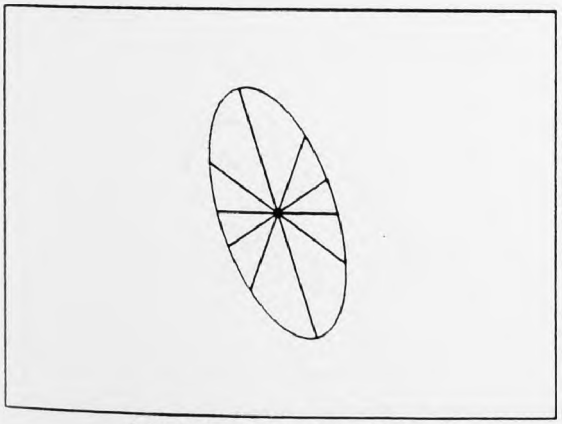
SILHOUETTE IMAGE

Figure 5.2(15)
NA = 200, PHI = π/5, EC = 0.90



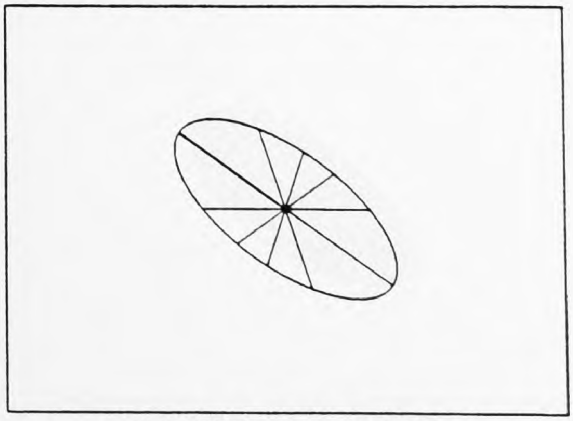
SILHOUETTE IMAGE

Figure 5.2(16)
NA = 200, PHI = 2π/5, EC = 0.90



SILHOUETTE IMAGE

Figure 5.2(17)
NA = 200, PHI = 3π/5, EC = 0.90



SILHOUETTE IMAGE

Figure 5.2(18)
NA = 200, PHI = 4π/5, EC = 0.90

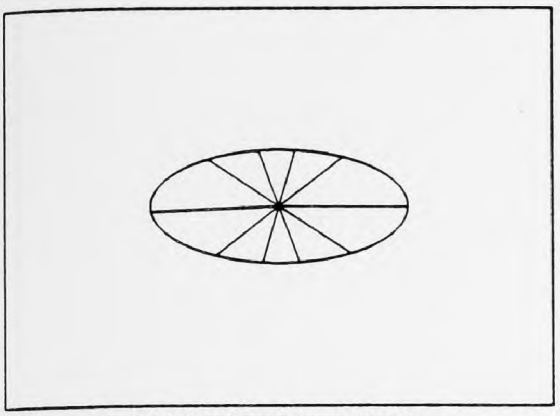


Figure 5.2(19)
NA = 200, PHI = π, EC = 0.90

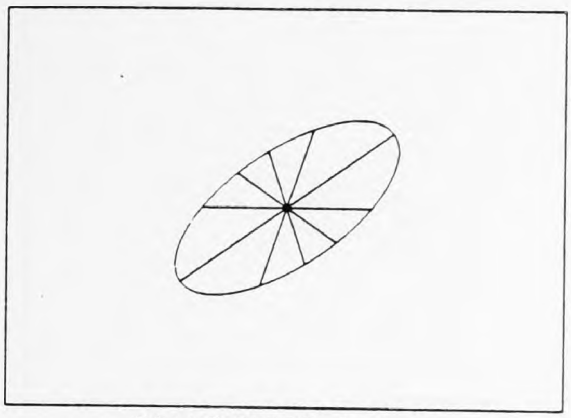


Figure 5.2(20)
NA = 200, PHI = $6\pi/5$, EC = 0.90

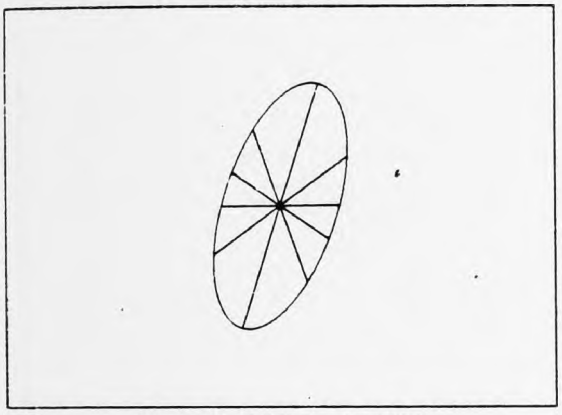


Figure 5.2(21)
NA = 200, PHI = $7\pi/5$, EC = 0.90

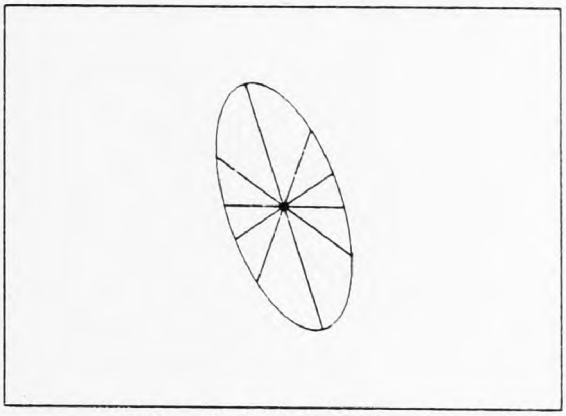


Figure 5.2(22)
NA = 200, PHI = $8\pi/5$, EC = 0.90

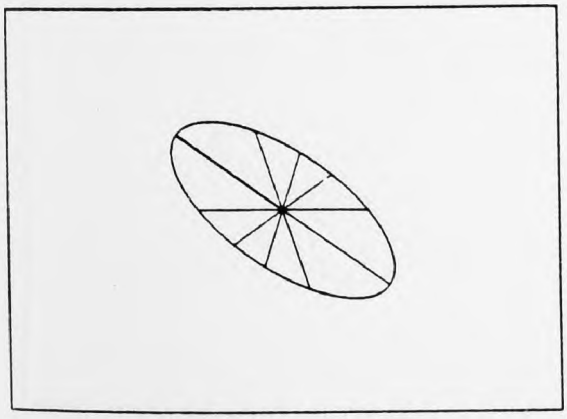


Figure 5.2(23)
NA = 200, PHI = $9\pi/5$, EC = 0.90

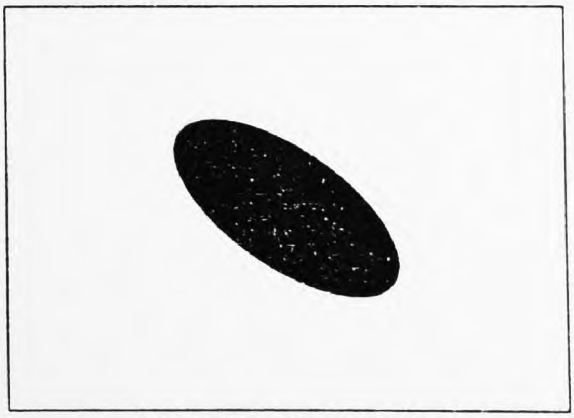
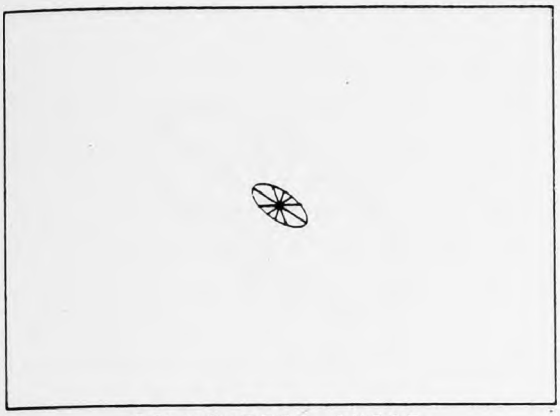
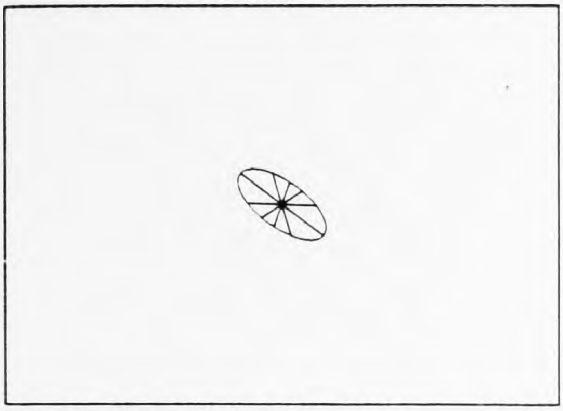


Figure 5.2(24)
NA = 200, PHI = $9\pi/5$, EC = 0.90



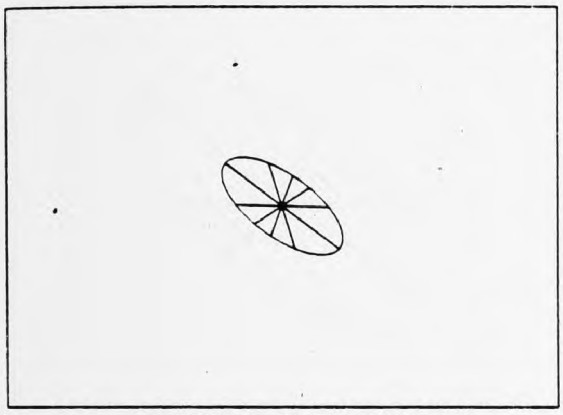
SILHOUETTE IMAGE

Figure 5.2(25)
NA = 50, PHI = 2.5, EC = 0.90



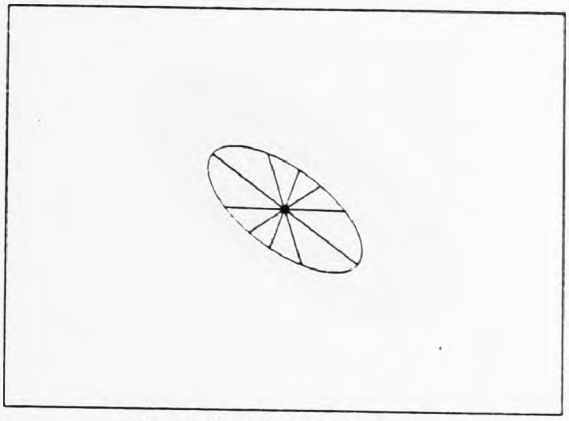
SILHOUETTE IMAGE

Figure 5.2(26)
NA = 80, PHI = 2.5, EC = 0.90



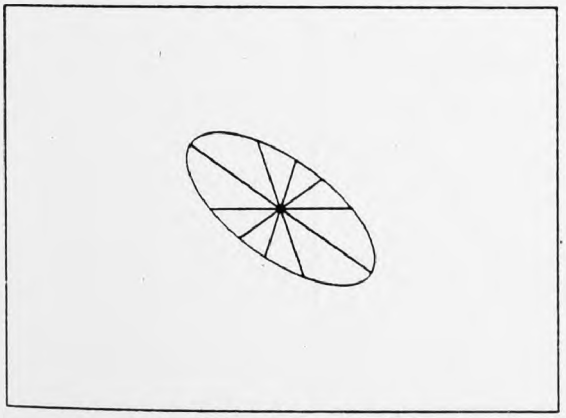
SILHOUETTE IMAGE

Figure 5.2(27)
NA = 110, PHI = 2.5, EC = 0.90



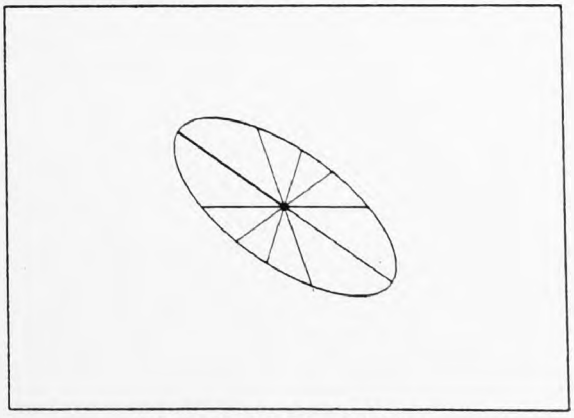
SILHOUETTE IMAGE

Figure 5.2(28)
NA = 140, PHI = 2.5, EC = 0.90



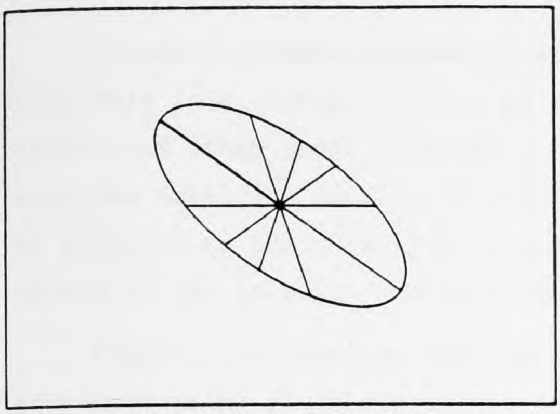
SILHOUETTE IMAGE

Figure 5.2(29)
NA = 170, PHI = 2.5, EC = 0.90



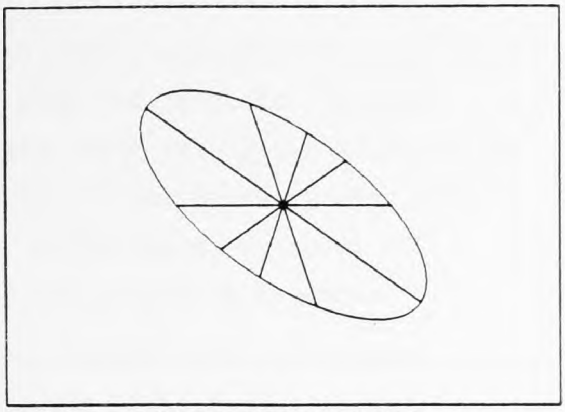
SILHOUETTE IMAGE

Figure 5.2(30)
NA = 200, PHI = 2.5, EC = 0.90



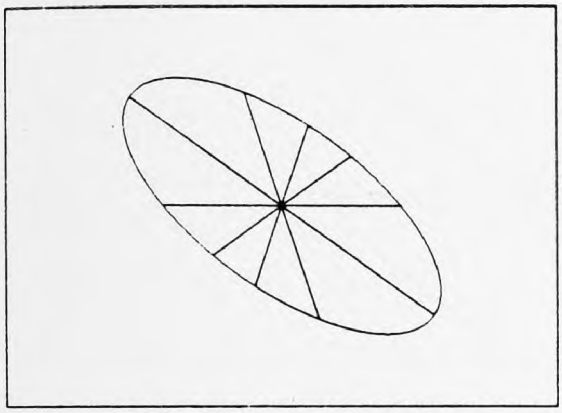
SILHOUETTE IMAGE

Figure 5.2(31)
NA = 230, PHI = 2.5, EC = 0.90



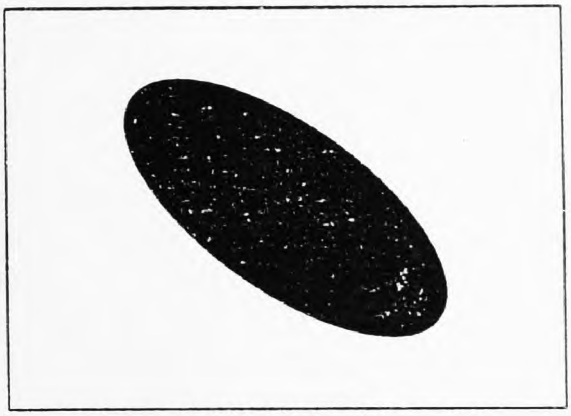
SILHOUETTE IMAGE

Figure 5.2(32)
NA = 260, PHI = 2.5, EC = 0.90



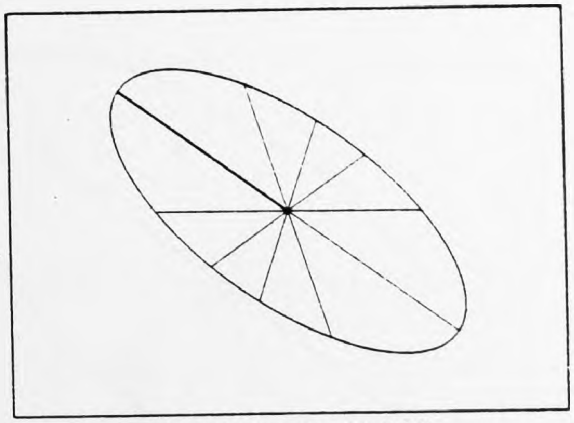
SILHOUETTE IMAGE

Figure 5.2(33)
NA = 290, PHI = 2.5, EC = 0.90



SILHOUETTE IMAGE

Figure 5.2(34)
NA = 290, PHI = 2.5, EC = 0.90



SILHOUETTE IMAGE

Figure 5.2(35)
NA = 320, PHI = 2.5, EC = 0.90

Figure 5.3 shows a boundary extracted with a threshold in one direction only (i.e. along the line of scan), with the centroid, longest radius and other radii inserted. It is seen that taking the gradient in only one direction results in gaps appearing in the boundary when this is parallel to the direction of scanning, which leads, in turn, to errors in the location and measurement of radii used as features.

Figure 5.4 shows the same information plotted with a boundary extracted using gradients in two perpendicular directions; it is apparent that the gaps and hence the errors in radius measurement have now disappeared.

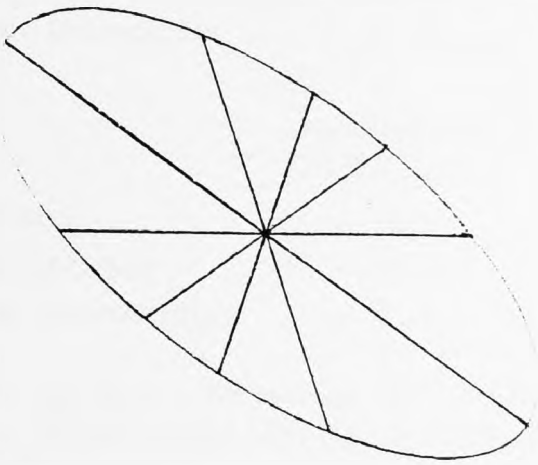


Figure 5.3: Threshold in one direction only

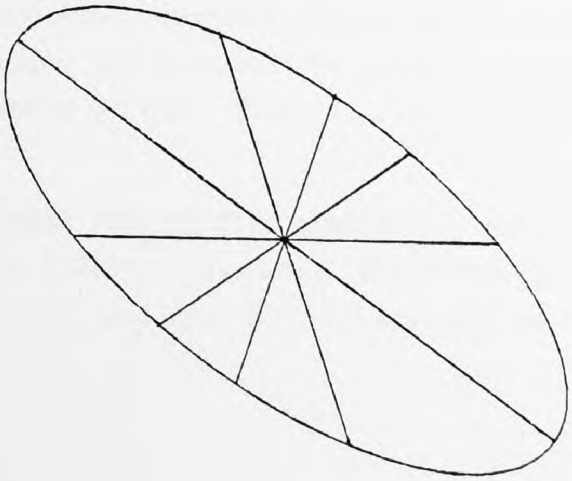


Figure 5.4: Threshold in two perpendicular directions

The quantitative results for the ellipse are presented in Table 5.T1 below:

Property	Minimum Size	Maximum Size
Area vs. Size	+ 0.96	+ 0.313
Perimeter vs. Size	- 14.58	- 15.70
Maximum Radius vs. Size	- 1.38	- 0.06

	Major Axis at 0 Rad. with respect to Grid	Major Axis at $\pi/5$ Rad. with respect to Grid
Area vs. Orientation	+ 0.92	+ 0.85
Perimeter vs. Orientation	- 10.00	- 15.00
Maximum Radius vs. Orientation	+ 0.42	+ 0.05

	Eccentricity = 0.1	Eccentricity = 0.9
Area vs. Eccentricity	+ 0.05	+ 1.40
Perimeter vs. Eccentricity	- 10.10	- 15.40
Maximum Radius vs. Eccentricity	- 0.01	- 0.08

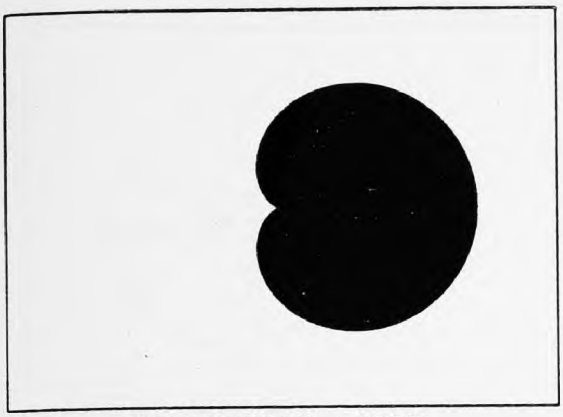
Table 5.T1: Ellipse - Percentage (%) Difference between Measured and Calculated Value of Stated Parameter

5.1.2 Cardioid

As in the case of the ellipse, cardioids were generated, of varying size, and lying at varying angles to the grid axes.

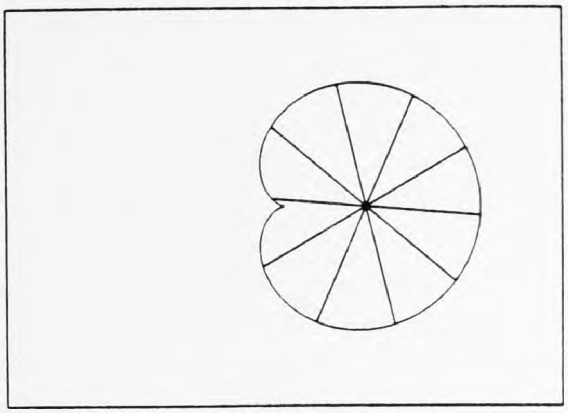
Here again, the computer program located the boundary cells, then, in turn, measured area, boundary length and location of centroid. Then the program located and measured the longest radius and measured the lengths of a number of radii lying at specified angles to the longest radius.

Typical images used in this simulation, plotted on microfilm, are shown in Figures 5.5(1) to 5.5(24). The parameters shown are: NA, for the semi-major axis, and PHI, for orientation, as they were for ellipse.



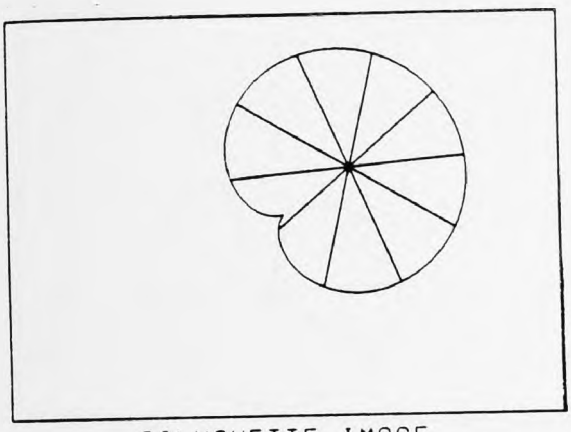
SILHOUETTE IMAGE

Figure 5.5(1)
NA = 150, PHI = 0.0



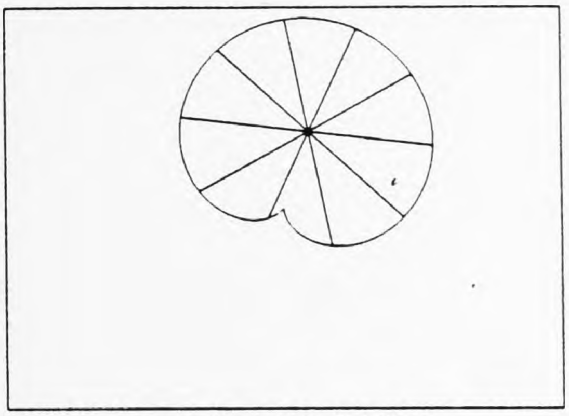
SILHOUETTE IMAGE

Figure 5.5(2)
NA = 150, PHI = 0.0



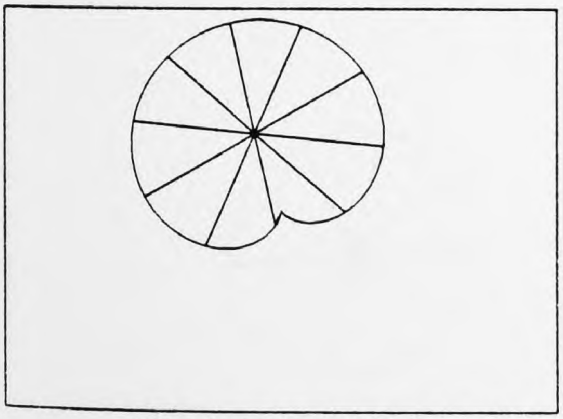
SILHOUETTE IMAGE

Figure 5.5(3)
NA = 150, PHI = $\pi/5$



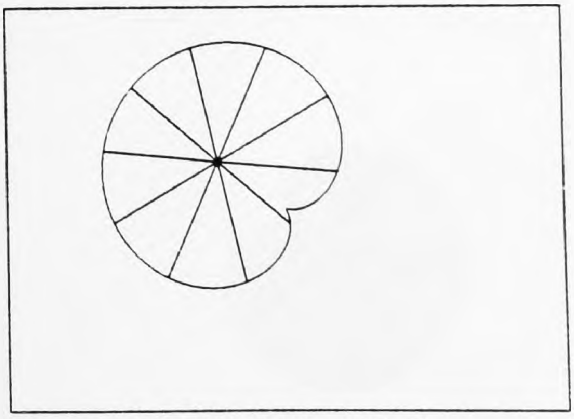
SILHOUETTE IMAGE

Figure 5.5(4)
NA = 150, PHI = $2\pi/5$



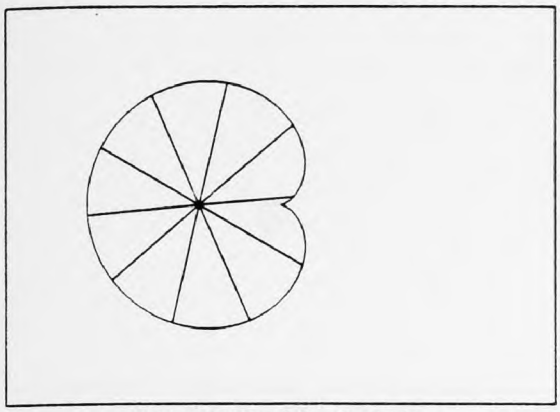
SILHOUETTE IMAGE

Figure 5.5(5)
NA = 150, PHI = $3\pi/5$



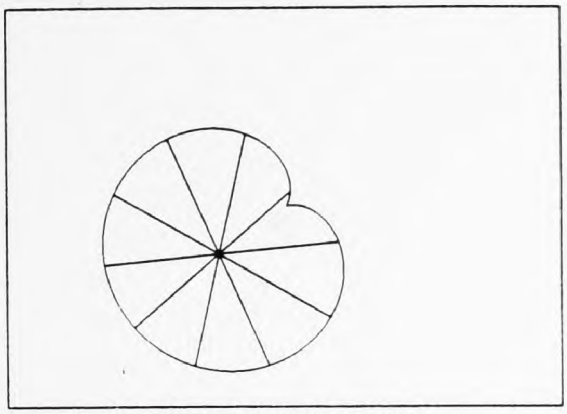
SILHOUETTE IMAGE

Figure 5.5(6)
NA = 150, PHI = $4\pi/5$



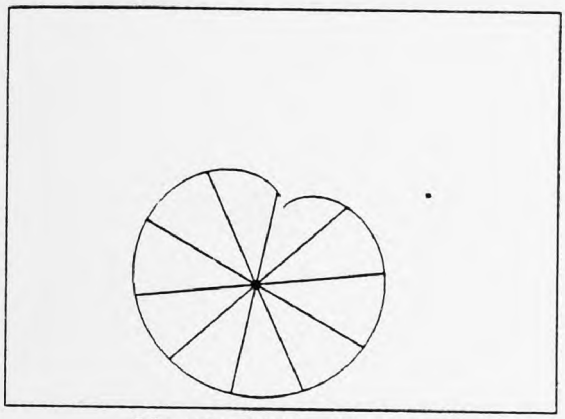
SILHOUETTE IMAGE

Figure 5.5(7)
NA = 150, PHI = π



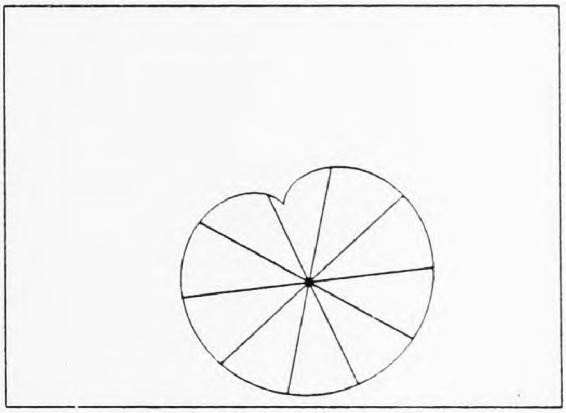
SILHOUETTE IMAGE

Figure 5.5(8)
NA = 150, PHI = $6\pi/5$



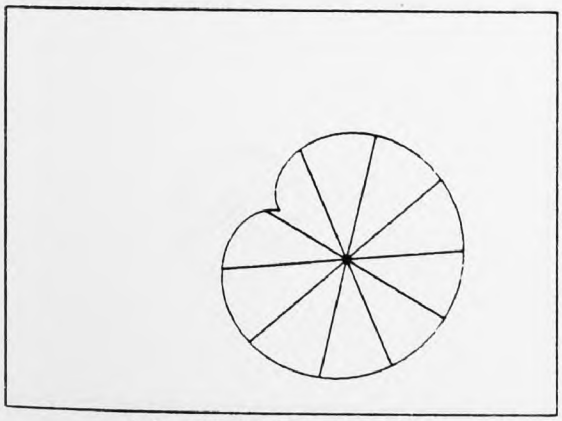
SILHOUETTE IMAGE

Figure 5.5(9)
NA = 150, PHI = $7\pi/5$



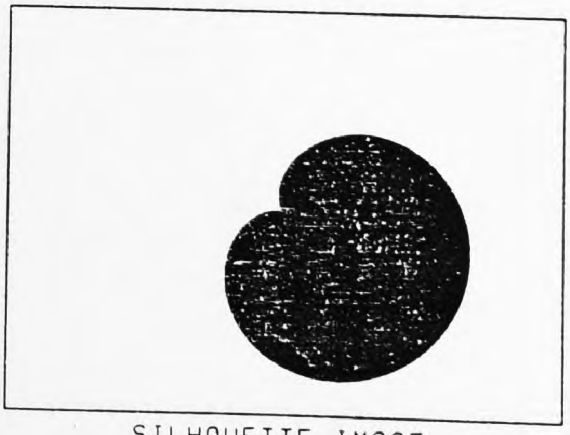
SILHOUETTE IMAGE

Figure 5.5(10)
NA = 150, PHI = $8\pi/5$



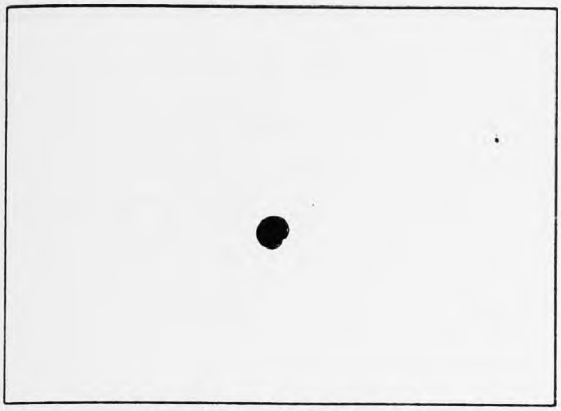
SILHOUETTE IMAGE

Figure 5.5(11)
NA = 150, PHI = $9\pi/5$



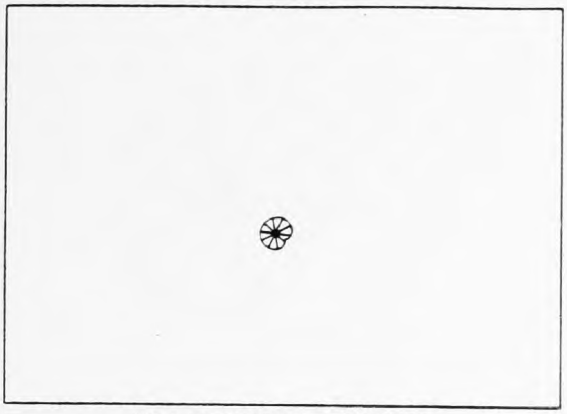
SILHOUETTE IMAGE

Figure 5.5(12)
NA = 150, PHI = $9\pi/5$



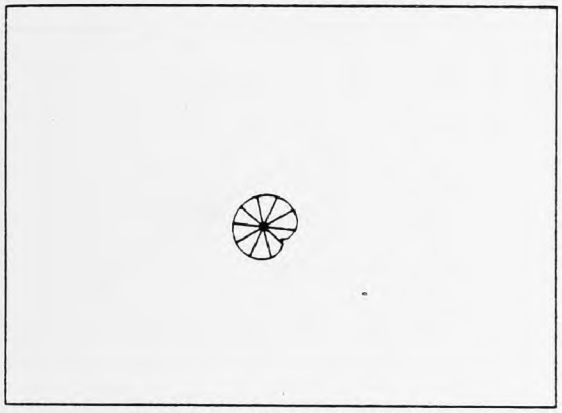
SILHOUETTE IMAGE

Figure 5.5(13)
NA = 20, PHI = 2.5



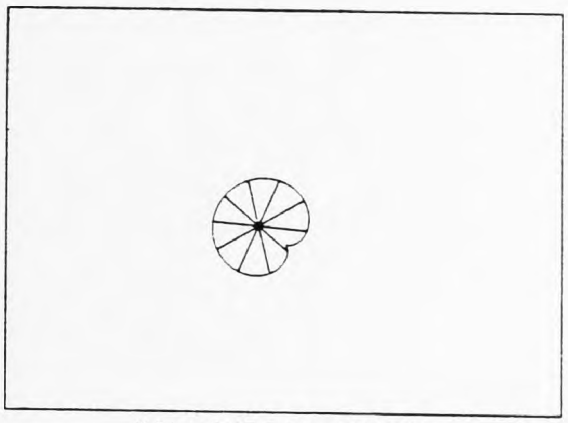
SILHOUETTE IMAGE

Figure 5.5(14)
NA = 20, PHI = 2.5



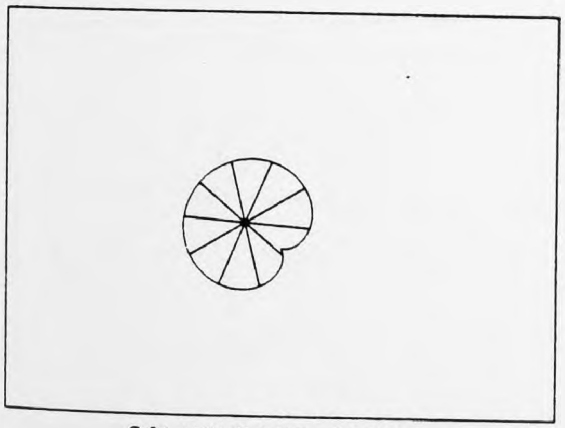
SILHOUETTE IMAGE

Figure 5.5(15)
NA = 40, PHI = 2.5



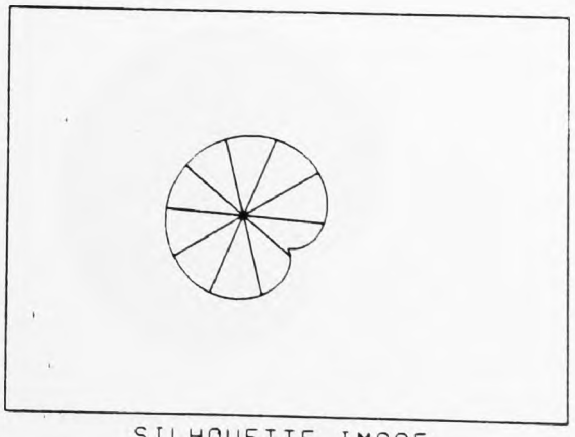
SILHOUETTE IMAGE

Figure 5.5(16)
NA = 60, PHI = 2.5



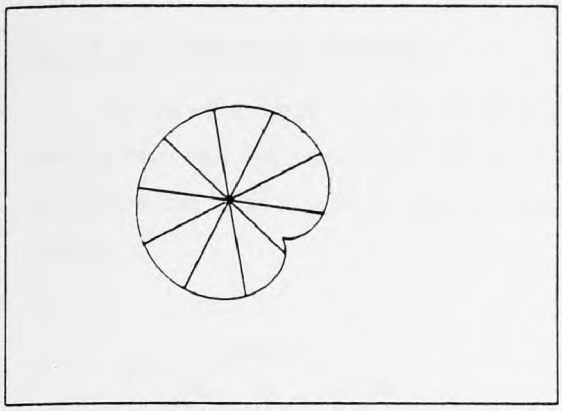
SILHOUETTE IMAGE

Figure 5.5(17)
NA = 80, PHI = 2.5



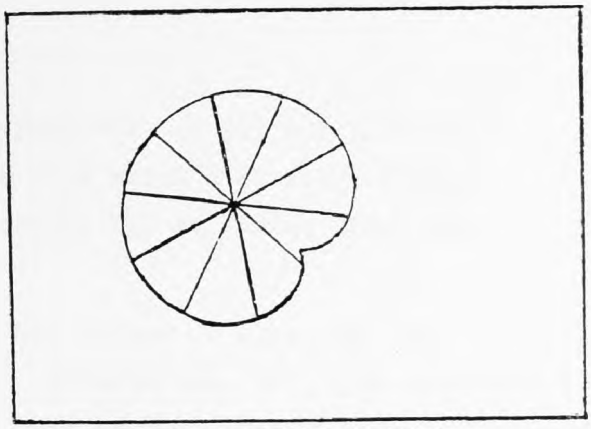
SILHOUETTE IMAGE

Figure 5.5(18)
NA = 100, PHI = 2.5



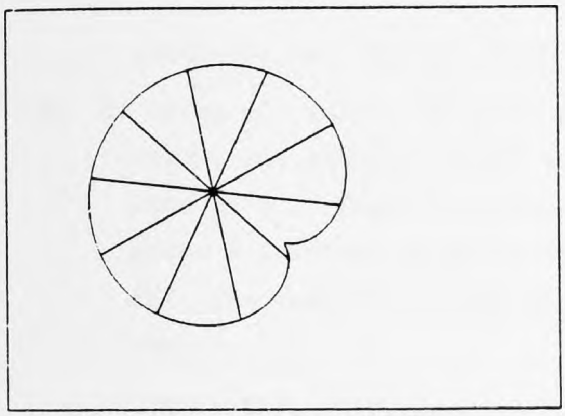
SILHOUETTE IMAGE

Figure 5.5(19)
NA = 120, PHI = 2.5



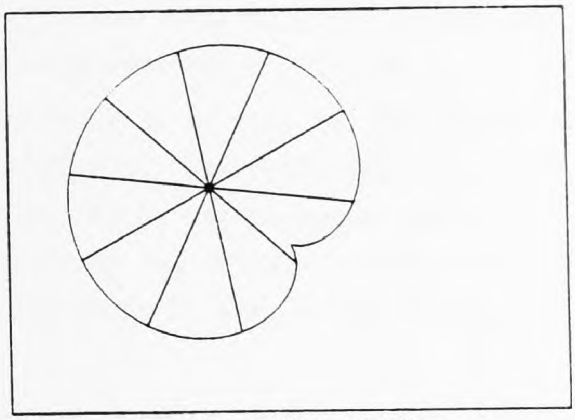
SILHOUETTE IMAGE

Figure 5.5(20)
NA = 140, PHI = 2.5



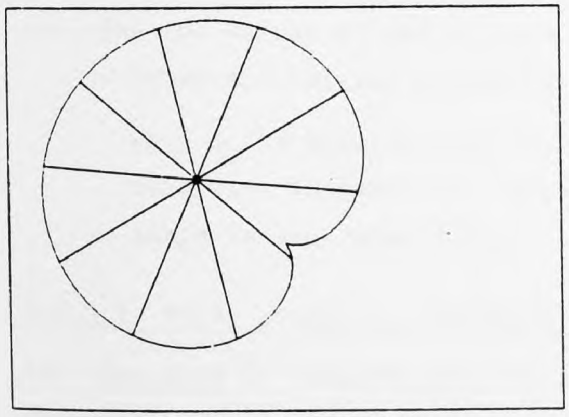
SILHOUETTE IMAGE

Figure 5.5(21)
NA = 160, PHI = 2.5



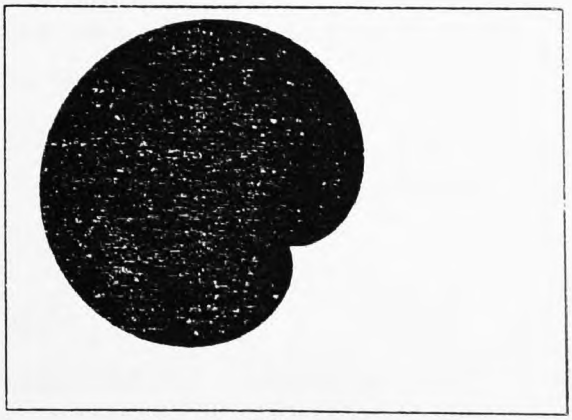
SILHOUETTE IMAGE

Figure 5.5(22)
NA = 180, PHI = 2.5



SILHOUETTE IMAGE

Figure 5.5(23)
NA = 200, PHI = 2.5

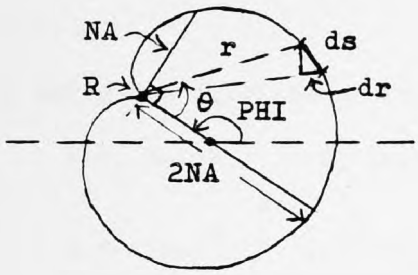


SILHOUETTE IMAGE

Figure 5.5(24)
NA = 200, PHI = 2.5

5.1.2.1: Masking operation

As in the case of the ellipse, an array A(I, J) was produced and was given the values 0 or TH in a manner with which a cardioid shape, as shown in the diagram below, was generated. The following steps were taken:



(1) The values of size, NA, and orientation, PHI, were selected as desired.

(2) The co-ordinates (X_R, Y_R) of a reference point in the grid were chosen. This time the point R was selected, as shown in the diagram, which is not the centroid, but through which passes the semi-major axis, NA.

(3) By using all values of I for each value of J, in turn for the whole array, the square root of the sum $(K_1^2 + K_2^2)$ was performed each time as S_2. Then, in turn, the quantity S_1 was computed, which obeys a cardioid property in relation to the desired orientation PHI, the semi-major axis NA, and any point K, say, on the shape, where:

$$S_1 = NA\left(1 + (K_1 \cos PHI + K_2 \sin PHI)/S_2\right), \text{ and } S_2 = \left((K_1^2 + K_2^2)\right)^{\frac{1}{2}}$$

and the values of K_1 and K_2 are:

$$K_1 = J - X_R, \text{ and}$$

$$K_2 = I - Y_R.$$

(4) The two values S_1 and S_2 were compared each time, giving to the array A(I, J) the values TH or 0 as follows:

When S_2 is greater than S_1, then A(I, J) is equalised to zero, otherwise it takes the value TH (threshold), until the cardioid shape is generated fully.

5.1.2.2 Area, centroid, perimeter and radii

(a) The Area is computed exactly as the area for the ellipse is computed.

For the purpose of comparing the computed value of the area with the one from calculation, a formula is deduced as follows:

Since for any triangle the area is 1/2(bc sinθ), where b and c are two neighbouring sides of the triangle and θ the angle between them, similarly for the cardioid by considering only a narrow triangle with an angle dθ between its two long sides r, we have, by integration, area A equal to:

$$A = 2\int_0^{\pi} (1/2)r^2 d\theta = NA^2 \int_0^{\pi} (1 - \cos\theta)^2 \, d\theta = NA^2 \int_0^{\pi} 4\sin^4(\theta/2)\, d\theta$$

by using the equation of the cardioid: $r = NA(1 - \cos\theta)$.

Now, let $\theta/2 = \delta$, then we get for area:

$$A = 4(NA)^2 \int_0^{\pi/2} \sin^4\delta(2d\delta) = 8(NA)^2(3/4)(1/2)(\pi/2) = [3\pi\, NA^2]/2.$$

(b) <u>The Centroid</u> is computed exactly as in the case for the ellipse.

(c) <u>The Perimeter</u> is again computed in a similar manner as for the ellipse, but, for the purpose of comparing the computed value with a calculated one, the calculation is performed as follows:

As in the case of the area, for a narrow long triangle whose base is equal to ds, we have:

$(ds)^2 = (rd\theta)^2 + (dr)^2$, and hence $(ds/d\theta)^2 = r^2 + (dr/d\theta)^2$

(see cardioid diagram shown previously).

Therefore, the perimeter of the cardioid is:

$$P = 2\int_0^{\pi} \left[(dr/d\theta)^2 + r^2\right]^{\frac{1}{2}} d\theta = 2\left[(NA\sin\theta)^2 + NA^2(1 - \cos\theta)^2\right]^{\frac{1}{2}} d\theta$$

$$= 2(NA)\int_0^{\pi} (2 - 2\cos\theta)^{\frac{1}{2}} \, d\theta$$

$$= 4(NA)\int_0^{\pi} \sin(\theta/2)\, d\theta$$

where:
$1 - \cos\theta = 2\sin^2(\theta/2)$

$$= 4(NA)\left[(-2\cos\theta/2)\right]_0^{\pi}$$

$$= 4(NA)\cdot 2 = 8(NA).$$

Hence, the calculated value for the perimeter is 8(NA).

(d) The Radii are computed in a similar manner as they were for the case of the ellipse shapes.

5.1.2.3 Simulation results

Not having the parameter of eccentricity here, the computations during the present simulation for the cardioid were carried out taking into account only variations in size (NA) and variations in the orientation (PHI), which this time coincides with the orientation of the axis of the major axis, 2 x NA, of the cardioid.

The quantitative results for the cardioid are presented in Table 5.T2 below:

Property	Minimum Size	Maximum Size
Area vs. Size	+ 0.21	+ 0.007
Perimeter vs. Size	- 12.50	- 10.40
	Major Axis at 0 Rads. with respect to Grid	Major Axis at $\pi/5$ Rads. with respect to Grid
Area vs, Orientation	- 0.02	- 0.005
Perimeter vs. Orientation	- 10.20	- 11.80
Maximum Radius vs. Orientation	- 0.02	- 0.012

Table 5.T2: Cardioid - Percentage (%) Difference between Measured and Calculated Value of Stated Parameter

5.2 Investigation using Real Patterns

The objective of this simulation was to demonstrate that the processing could distinguish patterns differing by one-half size at both extremes of the size range and also between left- and right-hand patterns which differed only in being mirror images of one another, at the extremes of the size range.

The above distinction should be achieved without any adjustment, by changing the magnification, of the scanning head.

The patterns were scanned and recorded on magnetic tape using the photodiode array based scanning rig in the Instrument Systems Centre in the Department of Systems Science at The City University (London).

The resulting data were processed in the CDC 7600 computer, using the algorithms developed with the artificially generated patterns.

Microfilm prints of the above images, with feature radii plotted, are shown in Figures 5.6(1) to 5.6(6). The longest radius is distinguished by being drawn more heavily.

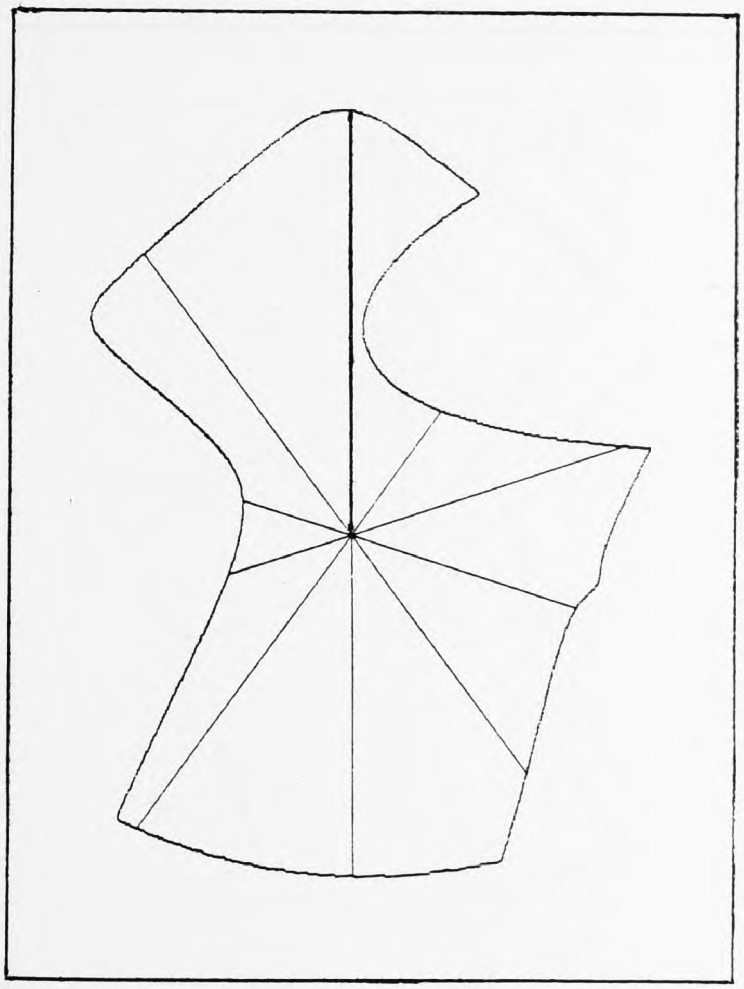


Figure 5.6(1): Real pattern, size 11

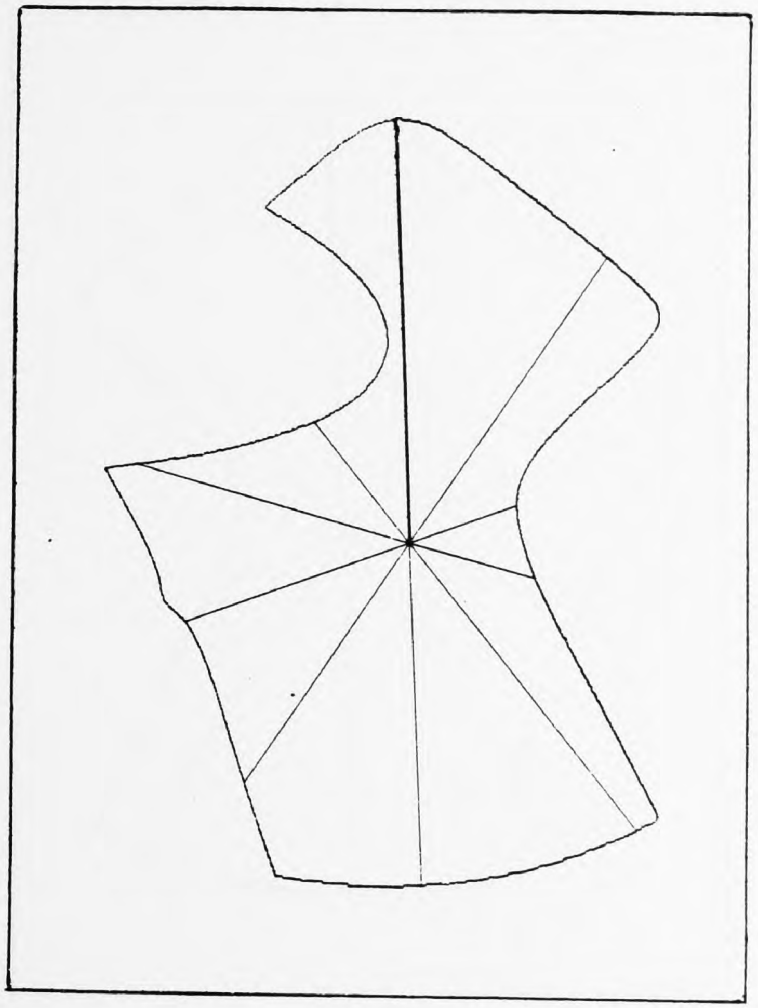


Figure 5.6(2): Real pattern, size 11, turned upside down

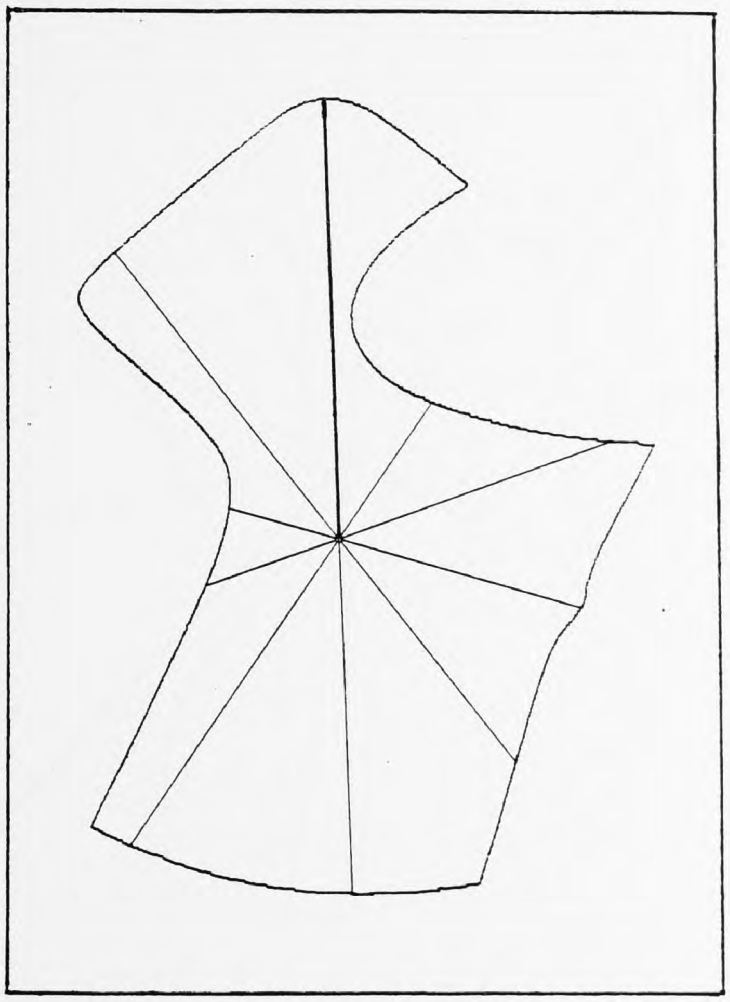


Figure 5.6(3): Real pattern, size 12

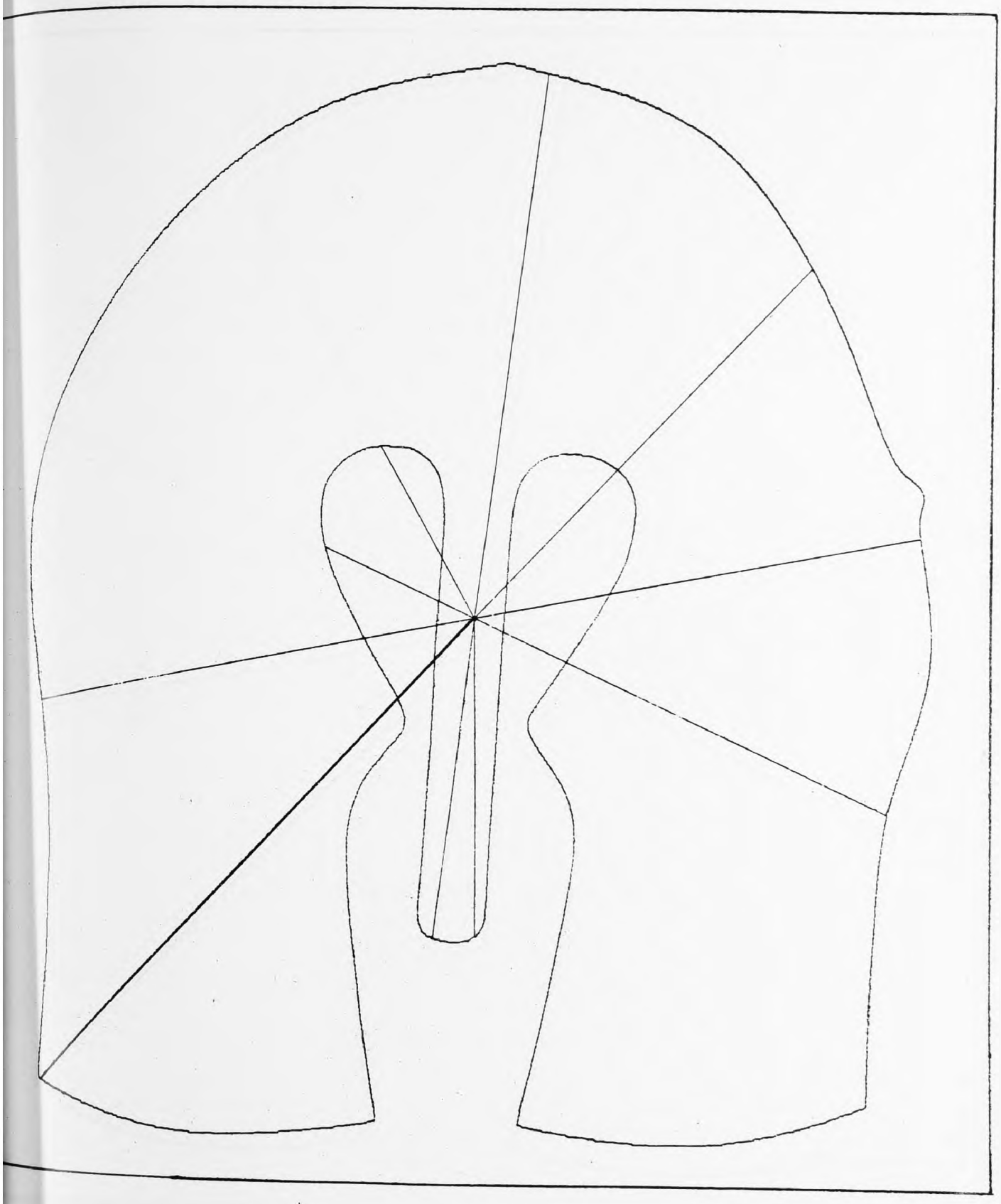


Figure 5.6(4): Real pattern, size $7\frac{1}{2}$

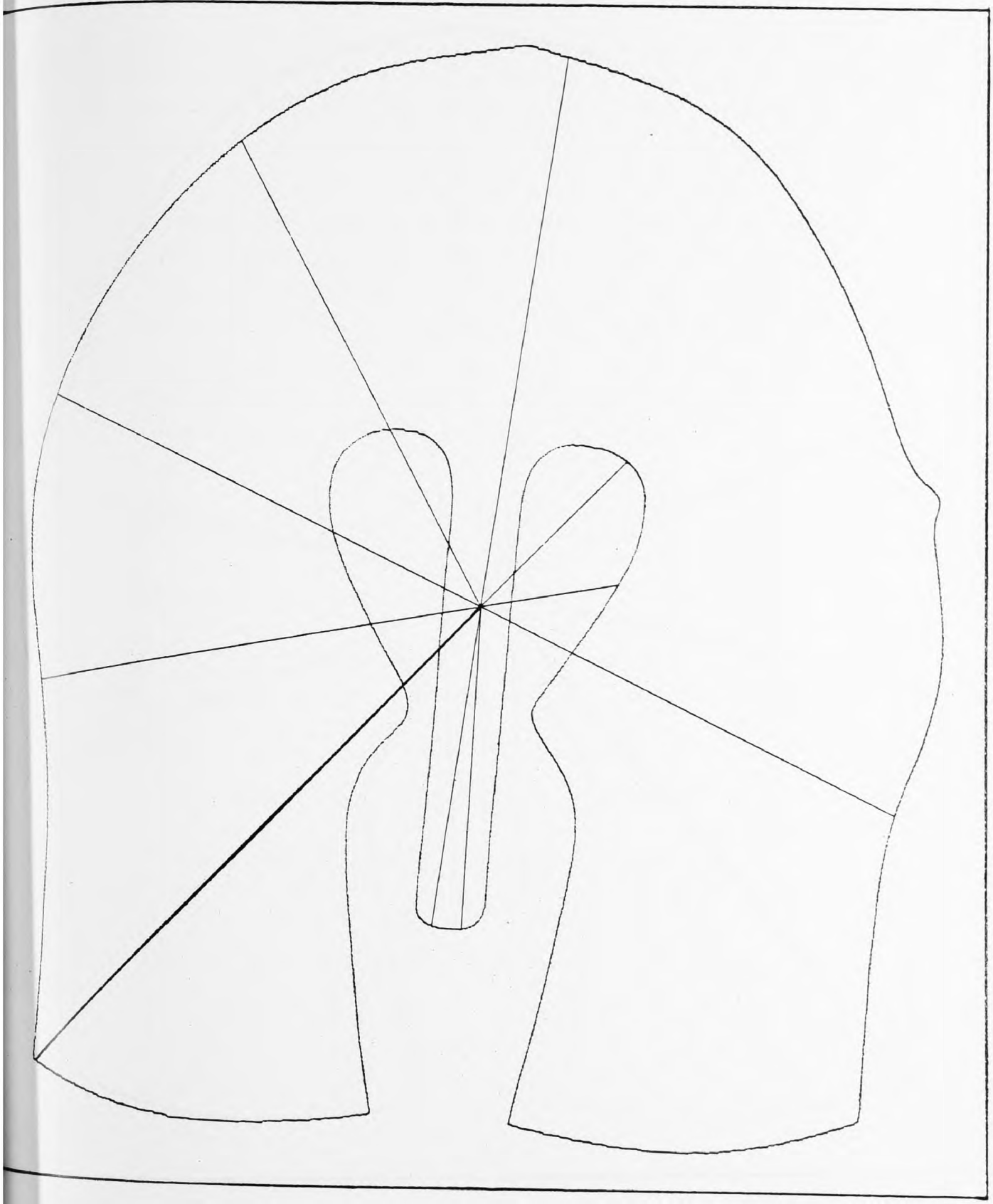


Figure 5.6(5): Real pattern, size 8

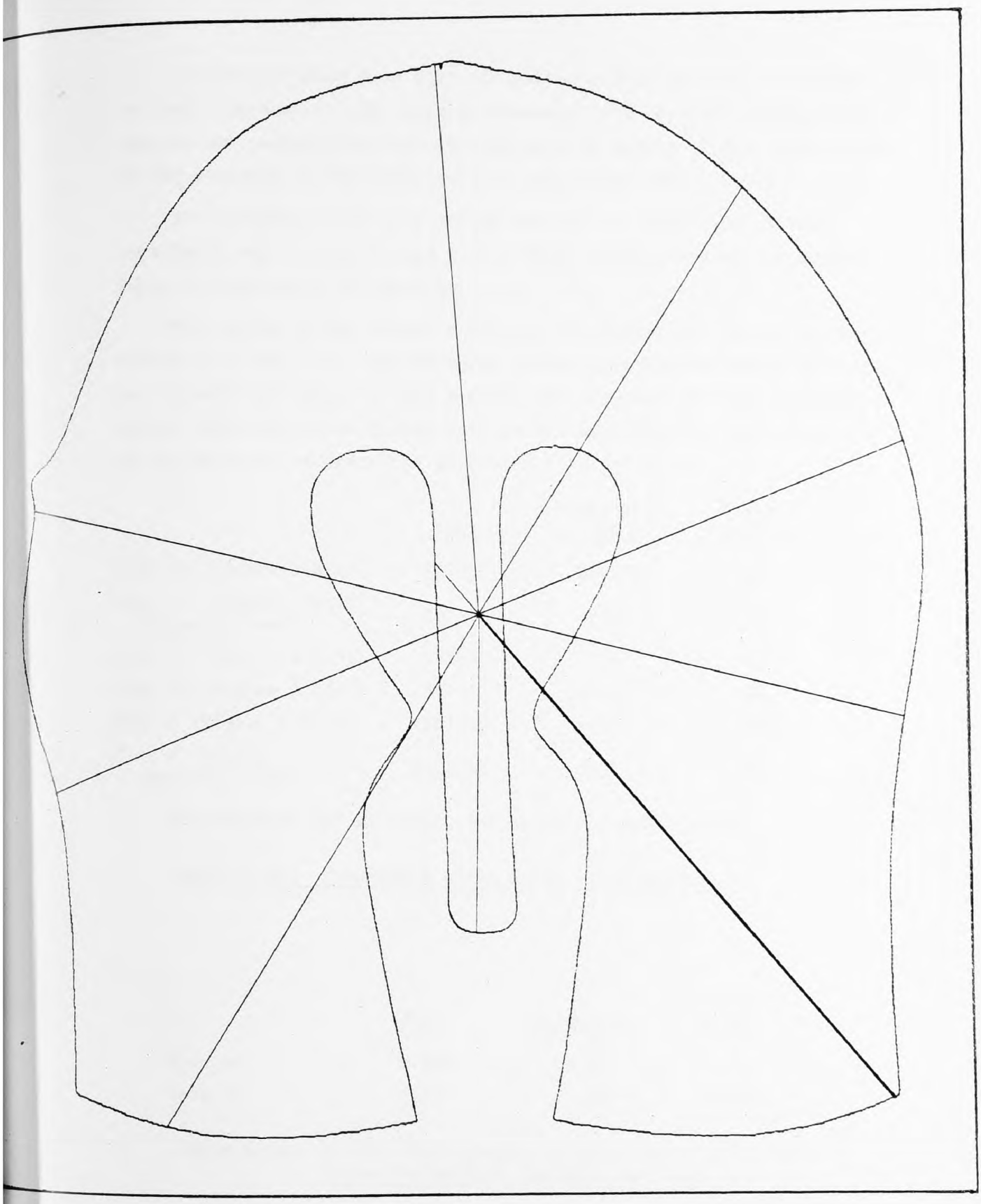


Figure 5.6(6): Real pattern, size 8, turned upside down

It must be noted here that for patterns with multiple crossings on their boundaries, the features measured from centroid to the boundary do not necessarily have to terminate on points of the outer parts of the boundary at the pre-specified angle intervals.

The scanning speed used during the on-line operation of this experiment was 3 scans/second with a vertical movement of the moving table of the rig of 0.2 mm/scan.

The values of the features for the six images are presented in Tables 5.T3 and 5.T4. The measured differences between these features are clearly sufficient to distinguish the patterns and are, moreover, larger than the errors which would result from rotation and translation of the pattern, as presented in Tables 5.T1 and 5.T2.

	Area	Boundary length	Maximum radius
Size 11 (Figure 5.6(1))	229329	2655	418
Size 11 (Figure 5.6(2)) (opposite hand)	229311	2838	419
Size 12 (Figure 5.6(3))	242301	2718	433
Size 7½ (Figure 5.6(4))	1551469	8646	883
Size 8 (Figure 5.6(5))	1587439	8816	888
Size 8 (Figure 5.6(6)) (opposite hand)	1587798	8954	886

Measurements are in cells, and at random orientation.

Table 5.T3: Parameters measured for Real Patterns

	Area	Perimeter	Maximum radius
Size 11	0.035	6.9	0.14
Size 8	0.02	1.56	0.19

Table 5.T4: Errors (Percentage) in Measurement of Stated Parameters for Real Patterns

5.3 Discussion and Conclusions

In this section the results taken for both cases, the mathematically defined shapes and the real patterns, will be examined in some detail, in order to investigate if there is an acceptable validity and agreement in the conclusions between them.

The measurement of the perimeter of a silhouette image cannot be considered as an important characterising property for accurate recognition and inspection purposes. This is because from Tables 5.T1 and 5.T2 we see that the error in computing the length of the boundary takes an average value for both shapes of the ellipse and the cardioid of about - 13.5%, which value can be considered to be acceptable only if the requirements for accuracy fall into this range of error.

On the other hand, the extraction of the boundary points and their spatial relationship with the centroid is essential so as to enable us to extract features (radii) from centroid to the boundary, which can be used for more accurate identification of patterns.

Also from Tables 5.T1 and 5.T2, it can be seen that for the ellipse and the cardioid the error in computing the area and the size of the radius from the centroid to the boundary is kept very low, and decreases substantially as the size of the image increases. This indicates that whenever very small shapes are put under similar tests, extra care should be taken in the choice of the grid size.

The number of cells required in the line scanning array can be deduced as follows. The most useful parameter for distinguishing small shapes or closely similar patterns must be measurable to an accuracy of $\pm$ 0.5% for the smallest items; this will guarantee separation, since measurements on real patterns indicate a spread of 2% to 5% between adjacent size ranges.

The smallest items are 25 mm square, the largest ones approximately 200 x 450 mm. Measurements for the cardioid indicate that for a semi-major axis of length 20 cells (i.e. 40 cells approximate diameter for the whole item) this accuracy can be achieved.

To accommodate a 200 mm length without change in magnification, 320 cells will be needed. The next largest standard array contains 512 cells which will provide some margin for error. To improve resolution bigger arrays are recommended.

From the results of Tables 5.T3 and 5.T4, it can be seen that turning the patterns upside down, keeping also their orientation at random, the errors which occur are very small for the area and the radius, and bigger for the perimeter, as was also found for the ellipse and the cardioid when their orientation was changed.

Again, here, for patterns turned upside down, by inspecting the microfilm images of Figures 5.6(1) and 5.6(2), we can see that their corresponding 10 extracted features are nearly in reverse order. This may clearly be used, according to specific requirements, for inspection of left, or right, foot.shoe pattern identification problems, for example.

Also, by looking at Table 5.T3, we can see that there is a clear difference in size between adjacent areas and corresponding radii (i.e. between patterns "11" and "12" and between patterns "7½" and "8").

So far as the perimeter is concerned, for the real patterns, although distinguishable between the two bigger ones in size, it does not give the same distinction for the two smaller patterns.

Therefore, as was said for the mathematically defined shapes, we can come to the same conclusions for the real patterns, by stressing once more that the area and the radii features are decisive factors for recognition and inspection purposes.

In most of the industrial applications the accuracy required is less than that achieved with the present method.

Specifically, area can evidently be measured to an accuracy of one part in 1500 for the small pattern (size "11") and one part in 15000 for the large pattern (size "8"). The difference in area due to a change of one size is about 5% for the small sizes and about 2% for the large ones, and, hence, is much larger than the measurement error.

Though the exact values will vary somewhat with the shape of the pattern, and would be reduced if a coarser array were to be used, something which applies also for the lengths of the radii, little difficulty should be encountered in distinguishing adjacent sizes.

Further, left- and right-hand patterns of the same form can be distinguished with ease at both ends of the size range, as can be seen by comparing the feature entries of Table 5.T3. The rotation caused by the mirror imaging of the patterns is evident.

5.4 Summary Conclusion

Since the theoretical and simulation investigations have shown that the method suggested is very likely to work successfully in an industrial environment, it was concluded that this phase should be followed by a more involved experimental phase, by developing and using a fast system which could perform a compatible automatic on-line inspection operation.

CHAPTER 6: SILHOUETTE IMAGE SCANNING, ACQUISITION AND PREPROCESSING FOR THE AUTOMATIC ON-LINE INSPECTION OPERATION

6.0 INTRODUCTION

The principal achievement of the simulation phase of this work, previously described, has been the provision of a successful demonstration of the method selected for the inspection operation. As was also seen, by using only a small number of real patterns, some shortcomings were revealed, the worst of which was the excessively long time required to scan each pattern.

The above simulation tests with real patterns used a PDP 11/10 computer interfaced to a data acquisition rig based on a line array of photodiodes, the analog output of which was digitised by an analog to digital (A/D) converter. Further, with all computations (including boundary determination) in Fortran software, about 25 minutes were required to identify a typical pattern. Thus, a slow on-line operation resulted.

However, this difficulty was due to the equipment used rather than the basic method.

It was, therefore, decided to examine more closely the problem of on-line scanning, and to build a test rig which could function according to the requirements of the present work.

The objectives of building the test rig were:

(1) To demonstrate the capability of the basic design concepts of the image acquisition system to make the whole operation of the automatic on-line inspection as accurate and as effective as possible, including a reasonable speed of operation, and

(2) To produce a suitable rig, which is desired as an industrial requirement, for experimental development tests where large amounts of data are involved, for which a systematic evaluation is required, and, hence, a reasonable speed of operation is essential.

In order to fulfil the above objectives, the scanning rig, which was originally used during the simulation study, was considerably modified

so as to enable an on-line operation of higher accuracy.

Again, the rig is based on an array of photodiodes since, in this way, the requirement of spatial stability between the sensing elements is secured. Also, a special hardware interface system between scanner and computer is produced, which now allows each sample under test to be scanned effectively in only a few seconds of time, instead of several minutes (about 25 minutes) as was normally taken before.

The new interface detects boundary points by logically examining triples of binarised scans. Binarisation uses a hardware threshold comparison. The location and type of the boundary points are stored in high speed memory until a scan is complete; they are then directed into the computer for further processing.

This enables the photodiode array to be clocked at its maximum rate without overloading the computer input port and eliminates much of the software computation.

Using an array of photodiodes, it gives the advantage of the spatial stability between the sensing elements. Also, by choosing a suitable array, interms of separation distance between every two neighbouring elements (pitch) and sufficient total number of sensing elements in the array, we can have in this way appropriate resolution for the scanning system.

The moving table of the scanner, on which samples may be placed and examined during each line of scan by the photodiode array through a camera lens situated opposite the table, is moved by a stepping motor controlled manually or by the computer.

The samples are illuminated, according to the application requirements, from the top or from underneath.

A multiplexing electronic system had also to be designed, so that both the original slow operation (clocked at 175 KHz) and the new fast operation (clocked at 5 MHz) could be accommodated in the scanning rig, and performed one at a time without any interference with each other. Thus, other research projects based on the slow operation of the scanning rig, which started after the present work was initiated and before the fast system was developed, could be carried out independently.

Also, the problems of illumination and the accuracy of the movement of the table of the scanning rig, as well as several sources of distortion, were examined in some detail, and are included in this chapter,

in order to assess their effect on the overall performance of the present inspection system.

6.1 The Scanning Rig

At this point it is proposed to discuss in some detail the scanning rig used for silhouette image acquisition in the tests which were carried out for the needs of the present work, by using real patterns in an on-line automatic scanning operation.

Figure 6.1 below shows in some detail the scanning rig as it is developed for the present work.

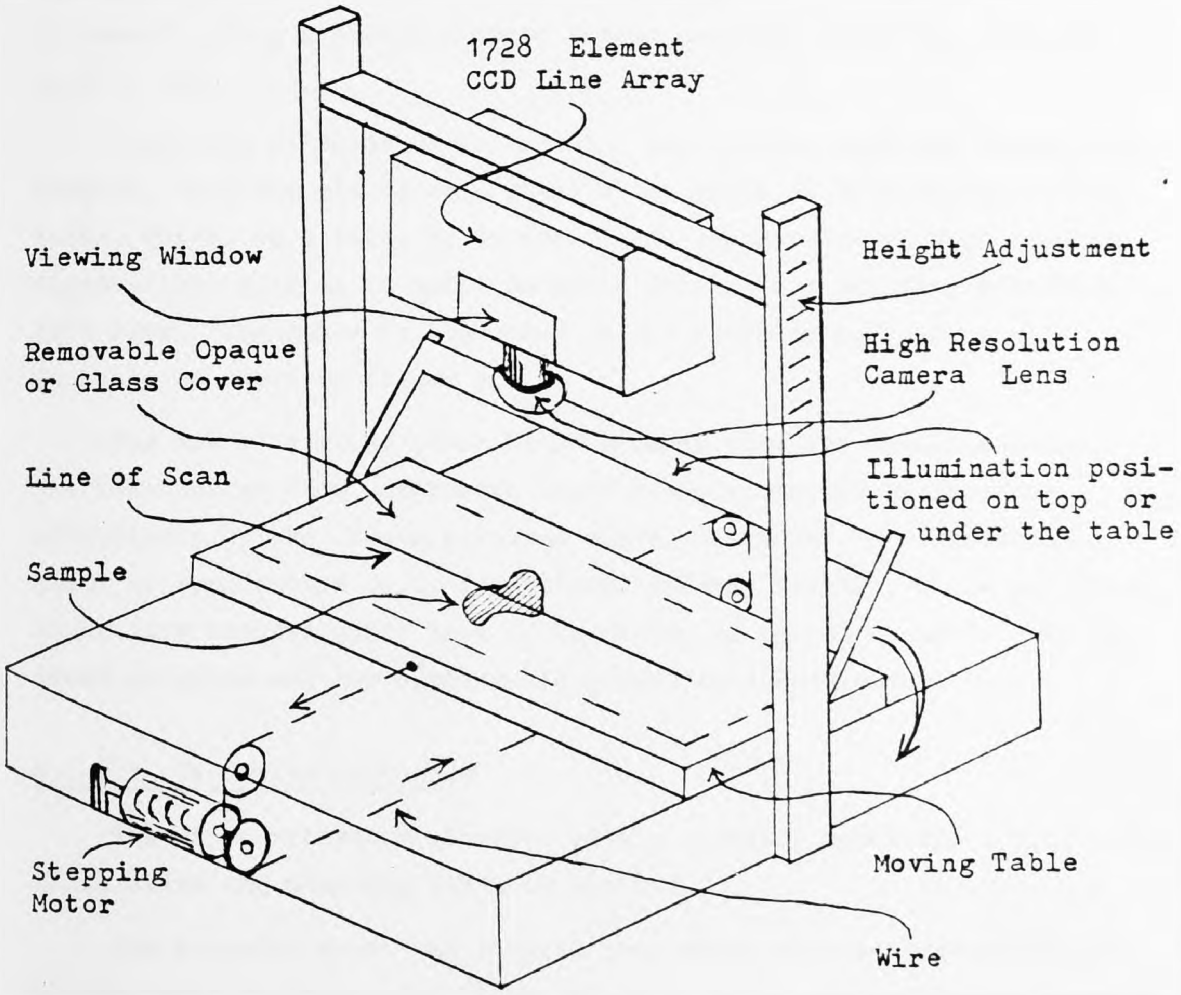


Figure 6.1: The Scanning Rig

The basic functions of the rig are:

(i) Stepping of scan,
(ii) Optical image formation and sensing,
(iii) Illumination,
(iv) Power and control,
(v) Image acquisition.

Each of the above functions is considered separately in the following sections of this chapter.

6.1.1 Stepping of scan

The stepping of the scan is performed by a moving table of 500 mm square, which is moved in 50 micron increments (or in multiples of this increment) using a stepping motor driven manually or by the computer, slow or fast.

Since the patterns under test for the present work are opaque flat samples, they are placed on a glass cover which is then placed on the table, which, in effect, is an iron frame (to provide strength and inflexibility) sliding on small metallic wheels in a separate stationary iron base. The table is connected to the stepping motor by a wire (cable), as shown in Figure 6.1.

The use of a glass cover helps because when the opaque samples are illuminated from underneath their silhouettes may be obtained effectively by the camera situated above the table. For different kinds of application (e.g. for printed circuit boards), where reflected light from samples under test is required, an opaque cover is used instead of glass and the samples are illuminated from above.

6.1.1.1 The drive mechanism

The drive mechanism consists of the stepping motor and a wire cable which drive the scanning table of the rig.

The stepping motor has a wheel gear which enables incrementation of the table in steps of 0.05 mm for every pulse applied to the stepping motor (the motor being of a type of 96 steps per revolution of its output shaft). This motor has replaced the original one which could provide only steps of 0.10 mm each, which was a restriction on the achievement of more accurate resolution.

The wire is a steel cable wrapped around the drum of the motor and attached to each end of the moving table with tensioning clamps. Rotation of the drum causes the cable to wind on one side and unwind from the other, and so to move the table. Built into the drum is a clutch mechanism for disengaging the motor drive from the gearing and drum. There is also provision for a mechanical damper to be bolted to the motor/gearing/drum assembly.

The cable drive works satisfactorily and, although stretching of the cable could cause inaccuracy in measurements, it is highly improbable that the cable would stretch by a significant amount within the scanning of a particular sample.

6.1.1.2 Measurement of accuracy of table movement

The quality of the drive has been investigated using a Hewlett-Packard laser distance measurer, and the variation in size of this nominal increment showed that the nominal 50 micron advance may vary for individual increments by about ± 10%, but that over increments of 2 mm or greater this variation is averaged out.

The periodicity evident in the variation of the individual increments is observed to correlate with the lie of the strands in the cable used to drive the table. It seems that single-strand drive wires which would be suitable are not available, since material having the necessary strength and lack of stretch would not be sufficiently flexible.

The investigation comprised three runs at each of three intervals of table advance. The first interval comprised 6 x 50 micron increments (0.3 mm), the second 7 (0.35 mm), and the third 8 (0.4 mm). Each interval indicates the separation from one scan to the next or, in other words, the resolution setting along the y-direction.

The results for the three different scanning intervals are summarised in Table 6.T1 below:

No. of steps per scanning increment	Nominal increment	Total Traverse	Average percentage (%) error in increment	Range of percentage (%) error in increment
6	300 micr.	99.90 mm	± 2.22	- 5.56 to + 6.12
7	350 micr.	99.75 mm	± 0.76	- 3.14 to + 4.57
8	400 micr.	100.00 mm	± 0.01	- 1.02 to + 2.16

Table 6.T1: Measurement of Accuracy of Table Movement

The above measurements show that with the 350-micron increment for each scan step, which is used in the evaluation process of the present work, the probable error in measuring distances is better than 0.8% when the traverse (i.e. the length measured) is about 100 mm, which is also the average size of the samples used in the inspection tests.

With the 350-micron interval, as y-direction resolution setting we could have the same x-direction resolution by moving the camera of the scanning rig at the right distance, using the new facility of the height adjustment mechanism which was added to the rig, as shown in Figure 6.1, and be able to accommodate in the field of view (i.e. along the line of scan) even the biggest in size sample, and also to keep the amount of data for processing at a reasonable level. The higher the resolution, the more data are required for processing. The 300-micron scanning step was considered as too high resolution, and the 400-micron one as too low for the present application.

Thus, although the probable error in individual 350-micron increments is of the order of $\pm$ 0.76%, the drive currently available appears to be adequate even for samples much bigger than 100 mm long.

A ball screw drive would probably be better (no backlash or oscillation), although undoubtedly this would be much more expensive.

6.1.2 Optical image formation and sensing

The scanning table is viewed from above by a high resolution (35 mm lens) camera, the sensing element of which is a line of 1728 CCD photodiodes (Fairchild type CCD 121 HC), with a 13 μm pitch. This size of array was considered to be quite adequate for the present work, or any other similar application for which the developed rig could be used as a scanning device.

The distance between the camera and the scanning table may be varied by the height adjustment mechanism, as was mentioned in the previous paragraph, so resolution adjustments may be possible whenever this is necessary.

6.1.2.1 Camera lens

The purpose of the camera lens is to have an effective accumulation of light on the sensing elements of the array which correspond to edge points along each line of scan of the sample being scanned.

Since the above sample is obstructing the light, which is coming from the light source positioned underneath the scanning table, the contrast between brightness and darkness around the edges of the sample may be enhanced by adjusting the focus of the camera lens to obtain the sharpest possible edge, and, in effect, to obtain a more accurate silhouette shape of the sample under test.

6.1.2.2 CCD array

Since the CCD array is the main part of the camera system, the operation of such an array used in the present scanning rig is described here as follows:

Firstly, charge is accumulated on the sensors (photosites) during an interval t(1). The line of charges is then transferred, effectively instantaneously (i.e. in less than 10**-7 seconds) in parallel to a corresponding line of storage sites which are shielded from the incident light. Each element of this line is then read out serially during a time t(2), during which the photosites are again accumulating light. If the array contains N elements (ignoring odd-even considerations, as they are actually divided) readout of the complete signal takes a total time of t(3), where t(3) = N.t(2).

Because the photosites are isolated from the readout process and can accumulate charge during the interval t(3), this can, in fact, comprise the total time used for light acquisition. It sets a lower limit to the total scan time, at t(1) = t(3) = N.t(2).

With the present array, t(2) is 10**-7 seconds and N is 1728. Thus, t(1) (i.e. t(3)), the minimum charge accumulation time, is 1.728 x 10**-4 seconds).

A signal, the amplitude of which is 50% of maximum (i.e. 0.5 volt for the CCD 121 HC, Fairchild), is obtained (according to the manufacturer's data) if the sensor is illuminated by radiation with energy density one microjoule per square centimeter (10**-6 joules/sq. cm). Since one watt is one joule/sec and the array integrates charge during an interval t(1) seconds, the total energy required for adequate illumination is 10**-6/t(1) watts, giving a density of 5.815 x 10**-3 watts/sq. cm.

Actually, an "equivalent" power of illumination must be used, which takes into account the variation of energy with wavelength for the illumination, and the variation of sensitivity with wavelength for the sensing

array. If the variation with wavelength of energy in the illumination is f(w) and the variation of sensitivity with wavelength in the sensor is g(w), then the equivalent energy W(eff) will be:

$$W(eff) = \int_{w(1)}^{w(2)} f(w).g(w).dw$$

It is assumed that the response of the sensor, or the energy content of the source, or both, is effectively zero, except for wavelengths between w(1) and w(2). This equivalence is important because the arrays are highly sensitive in the near infra-red (around 0.9 microns), and filament lamps emit strongly at this wavelength, whereas other sources, such as fluorescent lamps, do not.

Manufacturers' data normally specify f(w) and g(w) at M discrete wavelengths w(1), w(2), ..., w(k), ..., etc., which are equally spaced. Using this information, an adequate approximation of W(eff) may be obtained as the sum:

$$W(eff) = \sum_{k=1}^{N} f\big(w(k)\big).g\big(w(k)\big)$$

If f(k) and g(k), i.e. the source distribution and response, respectively, had been approximately equal for all k, then source and sensor would have been "well matched".

In the present circumstances, best use would have been made of the energy provided by the source in generating an output signal. A practical consequence of this effect is that a smaller output signal is obtained with a standard silicon photosensor if fluorescent illumination is used than with tungsten illumination.

Although both sources may provide the same total energy, the energy from the fluorescent tube is distributed uniformly over the visible spectrum, whereas that from the tungsten filament lamp is concentrated in the red and near-infra-red, where the silicon photodiode is preferentially sensitive.

Matching, of course, is not the only consideration which is important in the selection of a good source of illumination.

6.1.3 Illumination

Illumination is provided either by a line of filament lamps with a diffuser to even out the illumination, or by fluorescent light.

The rig is housed in a curtained enclosure, so that changes in the ambient lighting do not disturb readings during scanning.

At present the rig is illuminated by a fluorescent tube with no means for concentrating and directing light. As a consequence, the equivalent energy density at the camera, determined directly by measurement, is only about 2.4 x 10**-5 watts/sq. cm, which is too low by a factor of 240.

The situation is not so bad as it would seem, however, since the energy density which matters is that directly at the array, which in the rig lies in the focal plane of the camera lens. This concentrates the light, and in theory should increase the energy density by a factor of about 16. The concentration factor due to the camera lens is, unfortunately, impossible to calculate precisely, since there are losses due to reflections within the camera lens, a dependence on the aperture setting, and so on.

The improvement provided by the camera lens is, in fact, less than half that predicted. Nevertheless, the illumination needs to be increased from that currently available by a factor of only about 40 instead of 240, as would appear from measurements made directly in front of the camera lens.

Three approaches are available to the problem of obtaining adequate illumination:

The first is to increase the intensity of the source. That currently used wastes about 10 watts. Assuming the new source has the same efficiency, it will waste about 40 times as much, i.e. 400 watts. This is acceptable (provided a suitably intense source can be found), but the waste heat would have to be removed using a fan.

The second solution is to use lenses and reflectors to concentrate the light within a strip of 500 mm to 2 mm, and to make the light emerge with its rays parallel so there is no loss due to geometry. The extent to which this is possible using an inexpensive system of cylindrical lenses is not yet known.

The third solution is a combination of the first and second, if the

second alone cannot provide a sufficient concentration of energy.

The original rig illumination comprised three staggered parallel rows of small tungsten filament bulbs, driven from stabilised d.c. and housed in a reflector. This arrangement has approximately 55 microwatt/ sq. cm at the camera lens, with 60 watts input.

However, the illumination fell off appreciably towards the ends of the strip, and individual lamps frequently failed. The source was reasonably matched to the CCD array sensor, but was too unstable and uneven to be acceptable.

Consequently, this arrangement was replaced by a Thorn 13 watt, 525 mm long white fluorescent tube, which gave a lower effective illumination but was constant and stable.

It is driven by high frequency (20 KHz) a.c., which eliminates flicker, since the response time of the fluorescent coating is much greater than 50 microseconds.

Green light is found experimentally to be preferable to white, since the CCD array/lens combination gives best resolution in this region, although matching is worse, so the effective level of illumination is lower.

A green fluorescent tube having an effective emission about six times that of a normal white fluorescent tube is used in Xerox copiers. This is difficult to obtain commercially, but would seem a good choice. The remaining increase in light intensity can hopefully be provided by a combination of lenses and reflectors.

6.1.4 Power and control

The power and control units are not housed on the scanning rig but on separate units beside the rig.

The power supply unit provides power for the scanner and its electronics and for the preprocessor.

All the supplies have a common basic circuit in which there is a mains transformer whose secondary feeds a bridge rectifier, after which some smoothing occurs in the form of a large capacitor. Monolithic fixed voltage regulators are used to stabilise the different d.c. supplies, with internal foldback overload and short circuit protection. For extra safety, fuses in the d.c. lines have also been included.

The control unit accepts control signals from the computer, or sends control signals to it if it is necessary, and in particular through the "DR11-B" interface unit of the PDP 11/10 computer during data transfer and through the "DR11-C" interface unit of the computer during table movements.

The control signal switching board is an electronic circuit whose prime function is to route drive pulses, from the computer and from a manual hand-held control unit, to the stepping motor electronic drive circuit.

In the case of manual operation, switching contained within the power and control units enables the table to be driven from the hand-held control unit providing such facilities as FAST/SLOW and SINGLE SHOT (i.e. one pulse to move the stepping motor shaft one step of rotation).

The switching logic also contains provision for preventing the table from being driven hard against its end stops, by using micro-switches to "sense" when the table is at the end stop, and to perform the appropriate logic function to inhibit the drive pulses to the stepping motor. As soon as the direction command is reversed, either manually or by computer, the table drives away from the end stop immediately.

Due to the amount of logic gating employed, standard 74 series digital integrated circuits were mainly used.

6.1.5 Image acquisition

In this case, the function of image acquisition in connection with the scanning rig is performed mainly by the components of the camera, and in particular by the CCD array and the electronics associated with it, enclosed in a special unit, as shown in Figure 6.1.

The image acquisition chain is shown schematically in Figure 6.2 below.

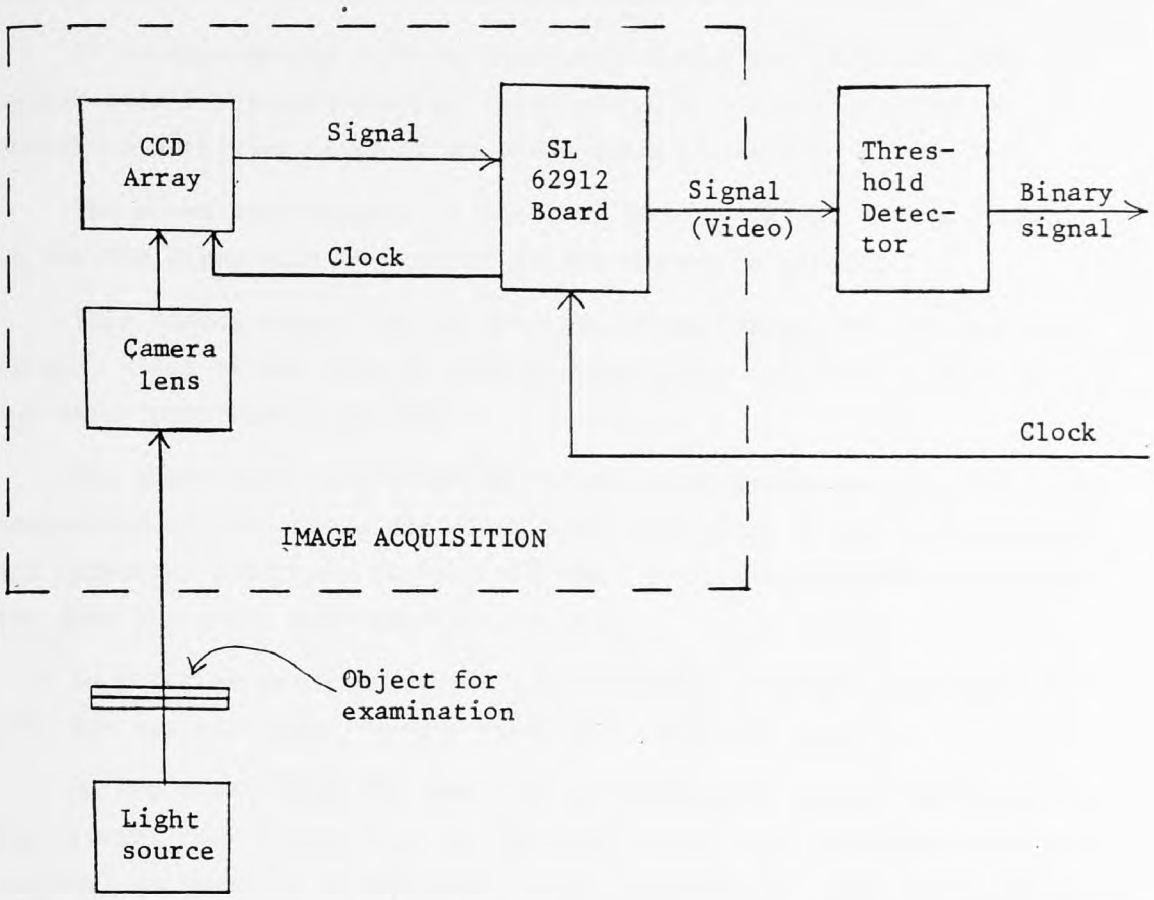


Figure 6.2: Image acquisition chain

The SL 62912 board is a development board unit provided by the manufacturers of the CCD array which is used in the present work. This unit consists mainly of the electronic circuitry which drives the array when the light source, the clock pulses and the power supplies are present.

The main components of the SL 62912 board are shift registers, which shift the resulting electric signal from the photodiodes at a rate determined by the external clock. This analog signal is then pre-processed externally by the preprocessor, which is not housed on the scanning rig but beside it on a separate interface unit.

The function of the CCD array, as it is described earlier, is vital for the image acquisition function and depends enormously on the illumination which is used.

Through a viewing window, which is provided in front of the camera unit (as shown in Figure 6.1), the experimenter may examine when necessary (e.g. for large samples) if the whole sample under test is inside

the field of view before its scanning commences.

It is appropriate here to describe briefly the original slow system used for preprocessing, which actually consists only of a threshold detecting circuit, as shown schematically in Figure 6.2.

The above preprocessor is based on digitising the analog output of the CCD array with the use of an A/D converter circuit.

This analog output is the so-called video signal of the scanning device, since it has similar characteristics to the video signal of any television camera device.

The above operation gives an overall slow performance in the inspection system, since the clock rate with which we can operate the A/D converter used here is only 175 KHz. Fast A/D converters do exist but they are still very expensive to buy.

In order to get an improved output signal, a sample and hold circuit has now also been incorporated with the above circuit.

On the other hand, the new fast preprocessing system developed for the present work (which will be described in a later paragraph of this chapter) is based on a specially constructed circuit with which effective thresholding is provided as well, as most of the computational requirements are eliminated by hardware means.

6.2 Distortions during Scanning

Distortions cause the apparent shape of a flat object to vary as its axis is rotated with respect to the line of scan.

The main effects which may cause distortion include:

(a) Finite workpiece thickness, and
(b) Refractive material between workpiece and camera.

6.2.1 Finite workpiece thickness

The general effect of finite workpiece thickness is indicated in the next Figure 6.3. The property that the nearside of the workpiece is appreciably closer to the camera than the remote side causes dimensions in the scan direction (perpendicular to the pattern motion) to appear enlarged.

The extent of this increase can be seen as follows. Referring to Figure 6.3, we see from simple geometry that the ratio of apparent width w' to true width w, given an object t metres thick placed on a table d metres from the camera, is given by:

$$w'/w = d/(d - t)$$

for d = 1 metres and t = 10 mm (say), then:

$$w'/w = 101\%, \text{ i.e. a } 1\% \text{ increase.}$$

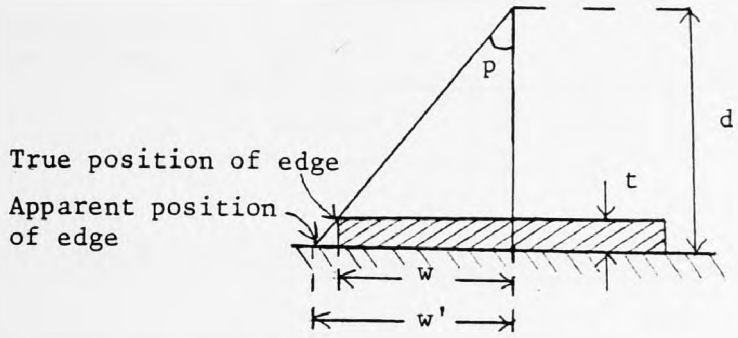


Figure 6.3: Effect of finite thickness of sample

Since the patterns under test are generally no thicker than 2 or 3 mm, the distortion thus caused should not exceed 0.2% or so, and should be acceptable. However, the camera distance d is adjusted to provide compatibility with the 50 micron table advance increment by scanning carefully machined discs until distortion is minimum. To eliminate finite thickness effect, which would ensure that a zero distortion position could never be found, it has been necessary to bevel the discs used for adjustment, so that their effective thickness approaches zero at the edge.

6.2.2 Refractive material between workpiece and camera

Because the material of the workpieces under test can be bent easily (e.g. leather), a second glass sheet (the first supports the shapes on the moving table) is used on top of the shapes to hold them flat during scanning. In this case, however, a distortion is introduced due to refraction.

The extent of this distortion may be determined by reference to Figure 6.4. If the intervening refractive material has an index greater

than one (invariably true in practice), the result is to increase the apparent width of the shape, without changing its dimension in the perpendicular direction.

In Figure 6.4, p corresponds to the angle of incidence and p' to the angle of refraction, of the ray touching the extreme edge of the shape.

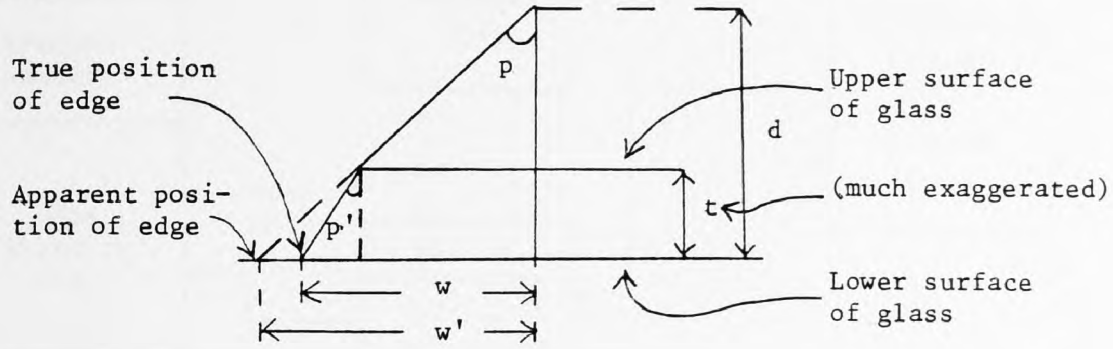


Figure 6.4: Effect of refraction through glass cover

Thus, we have from elementary theory:

$\sin p / \sin p' = mu$ (the refractive index).

For glass, mu is about 1.5. Actually, the ray proceeds in the reverse direction in our problem, but this does not affect the analysis in any way. We see that the error width Dw is given by:

$$Dw = t(\tan p - \tan p')$$

$$= t\left((w/d - w/(mu - 1) + mu.d)\right)$$

If w/d is less than 0.1, the above (after some manipulation) reduces to:

$$Dw = t.w/d(mu - 1/mu).$$

Substituting in the above reasonable values, e.g. t = 0.003, w = 0.1, d = 1 (all in metres), and mu = 1.5, we obtain:

$$Dw = 200 \text{ microns}.$$

Thus, for glass $\frac{1}{8}$ inch thick and a sample 100 mm wide, an apparent increase in width of 0.2 mm can be expected.

This is significant if a precision of 0.15 mm, say, is required, something which does not apply in the present work.

6.3 The Computer

Figure 6.5 below shows schematically the major parts of the computer unit in connection with the complete system used for the on-line scanning operation.

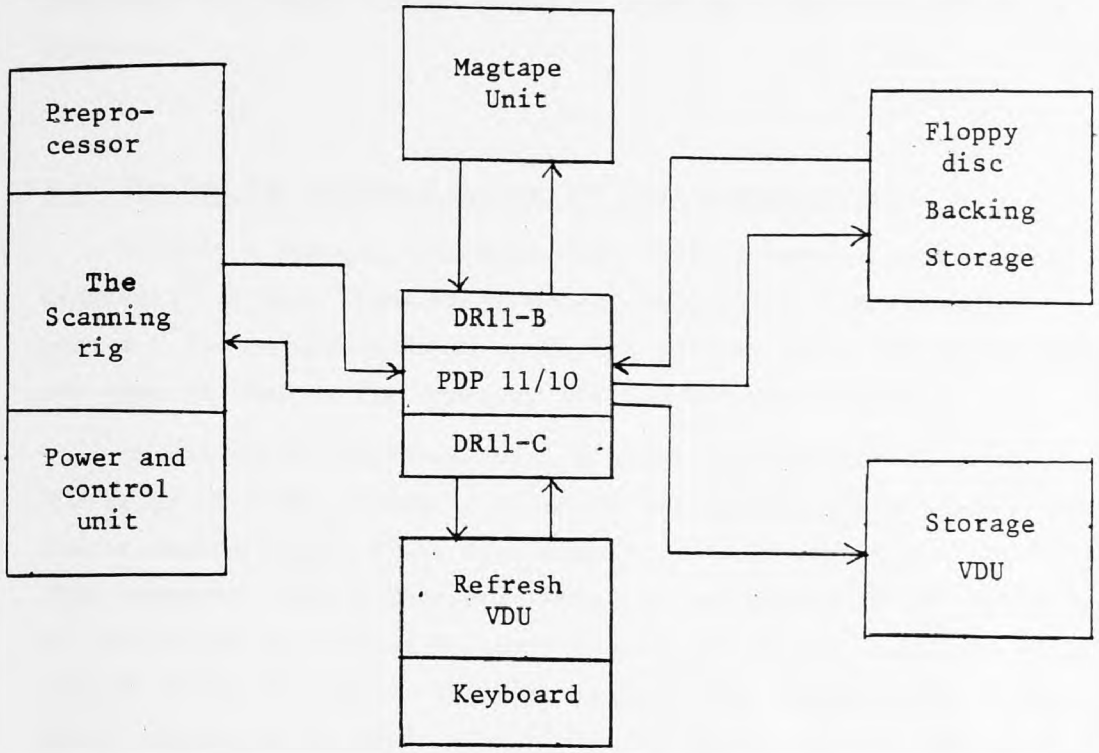


Figure 6.5: The computer in the scanning system

The computer used for the scanning system is a DEC PDP 11/10 type. It has 24K words of 16 bit random access memory, and two 7-inch flexible discs to provide backing store. Each disc can hold bytes and can be replaced by a new disc when it becomes full.

There are also two built-in hardware interfaces in the computer, named as DR11-B and DR11-C, respectively, used for any control or data communication between the computer and any other external hardware system.

As shown in Figure 6.5, there are two visual display units. One is a refresh display; it is used for displaying the response of the computer in a dialogue, through a keyboard, between experimenter and machine. The other is a high resolution storage display; it shows an area 1024 points by 768. The storage display cannot be changed quickly, but holds much more information than the refresh display. It

is used for plotting pictures of shapes (e.g. silhouettes), for showing features and also for displaying signals.

There is also a Racal magtape unit, which can take half-inch wide magnetic tape, on which data information may be stored and then be processed on a much larger computer, such as a CDC 7600, if it is required.

6.4 The Special Hardware System for Fast Preprocessing

This is a special interface unit located between scanner and computer. It uses hardware to detect silhouette shape boundary points. The co-ordinates of such edge points, which are extracted, can then be used by the computer for further processing.

In the above interface unit, a clock is generated to drive the CCD array at 5 MHz, which is close to its maximum rated speed. Its output analog signal (i.e. the video signal) is amplified. It is then compared with a threshold which is adjustable on the front panel of the device to provide a binary signal, which is a sequence of 1's (signal above threshold) and 0's (signal below threshold). A high-speed comparator is used. The binarised signal is then stored in a first block of high-speed random access memory (RAM). When the next scan is taken, data from the previous scan are transferred to the RAM.

Binary signals from the current scan and from the two previous scans are examined using a 3 x 3 element binary mask. The presence and direction of the boundary points are detected logically by locating the arrangement of '1' and '0' points within the mask in a look-up table.

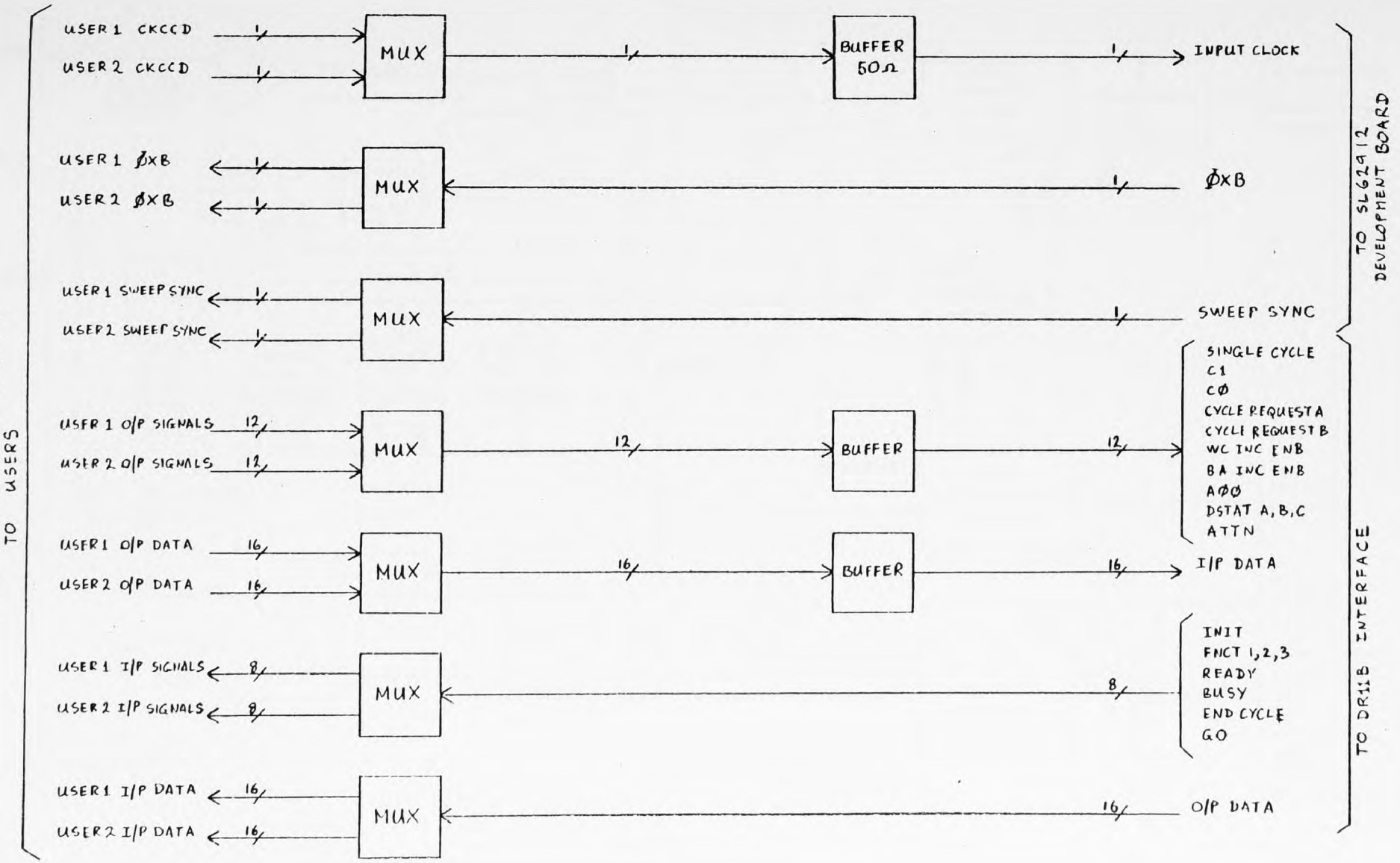
The x co-ordinate of each boundary point is stored in the next free location in the 'x co-ordinate' RAM and its direction in the 'edge type' RAM.

Once a scan is completed, the x co-ordinate and type of each boundary point discovered during the scan is given to the computer for further processing. The computer also keeps track of the number of scans completed and can, hence, supply the y co-ordinate.

The functions of the multiplexing system which was designed in order to link the slow and the fast scanning system of the rig

without any interference of each other, as well as the functions of the fast silhouette image preprocessor, which was designed for the present application, are presented schematically, as block diagrams, in the next two figures - 6.6 and 6.7, respectively.

The full description of the design of the above interface hardware, as well as its interaction with the PDP-11 computer used, is provided separately in an appendix of this thesis, which also includes the design of the multiplexing system developed for the present work.



BLOCK DIAGRAM: MULTIPLEXING FUNCTION

Figure 6.6: Multiplexing function

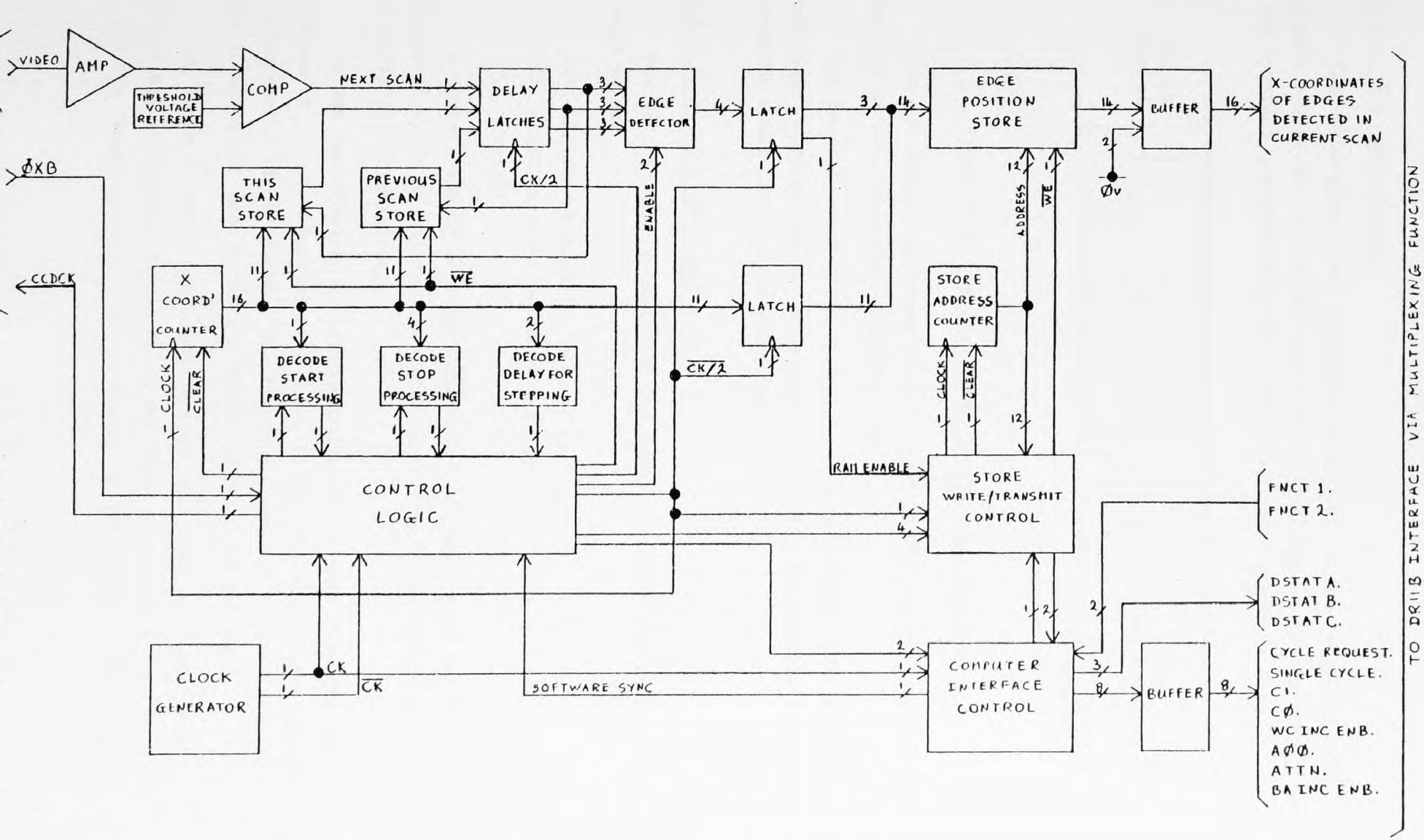


Figure 6.7: The Fast Preprocessor

BLOCK DIAGRAM: SILHOUETTE IMAGE PREPROCESSOR

6.4.1 The video signal

The video signal was originally deficient in two respects: the signal amplitude was low and there was a considerable inconsistency in the apparent response from different video cells across a scan.

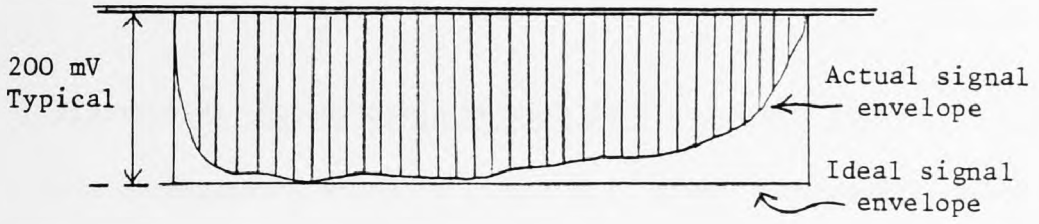


Figure 6.8: Typical clear field video signal

The low signal amplitude is not a serious difficulty since it can be, and is now, amplified appropriately.

The inconsistency shown, though, is a more serious difficulty since the signal size at the two ends approaches zero, so that the threshold detector would "see" edges where none exist.

6.4.1.1 Video signal amplitude

The factors mainly affecting the overall signal amplitude were found to be:

(i) Lighting intensity, and

(ii) Video integration time, i.e. length of time between transfer pulses.

A certain amount of experimentation took place with the present system in order to improve the video signal amplitude before it could be sufficiently amplified.

As was described in a previous paragraph of this chapter, the problem of illumination was considered to be sufficiently solved with a fluorescent light source. The lighting level has little effect on the video output once a certain threshold level is achieved.

In order to allow sufficient integration time, a "Dummy scan" period was introduced during which the video clock was slowed to 0.625 MHz (approximately 6 mS) necessary to integrate a "Real scan".

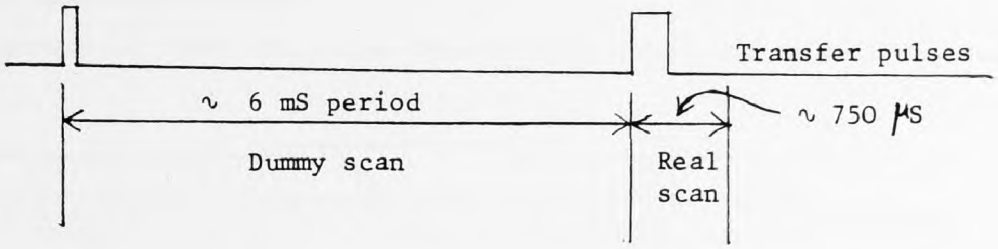


Figure 6.9: Relationship between real and dummy scans

This arrangement, shown in Figure 6.9, gives a reasonable video signal level during the real scan period at the expense of making the processing cycle slower than was originally intended.

However, this time penalty has very little effect on the overall processing cycle time. Greater delays are caused by the time needed for stepping the scanning table, as will be discussed in a later paragraph.

6.4.1.2 Video signal distortion

The video signal depicted in Figure 6.8 shows two types of distortion:

(a) Distortion at the two ends of the scan where the signal level is very small, and

(b) Variable response of individual video cells, also more strongly at the two ends of the scan.

The cause of the above distortions is a charge storage phenomenon associated with the circuitry surrounding the CCD chip. In particular, the different response of the video cells reflects the variability in electrical and optical characteristics of the photodiodes along the array.

These distortion effects were tackled in the present system, firstly, by ignoring the severely distorted end regions during processing, and, secondly, by relying on the threshold detector to filter out effectively the smaller distortions in the video signal.

Also, because the distortion effects became more obvious when the SL 62912 board of the array was run at high clock speeds, it was found that 5 MHz clock rate was the highest possible speed for the array to run effectively.

6.4.2 Signal thresholding

Figure 6.10 shows the video thresholding scheme which was adopted.

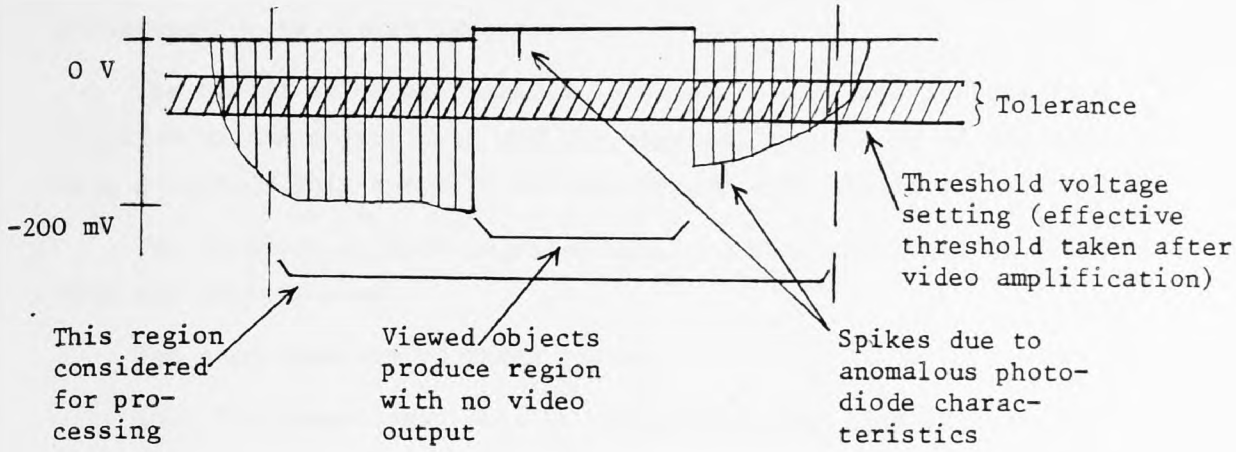


Figure 6.10: Video thresholding scheme

The new threshold detector is a circuit of a video amplifier followed by a comparator. This circuit is described fully in the appendix of this thesis.

A fast digital A/D converter was not used because of its very high cost, which is mainly due to the extra hardware required.

However, the thresholding circuit used gives a much faster binarised signal than the original one, and is sufficient for processing by the scanning system developed for the present work.

6.4.3 System speed

With the configuration of the fast system, the processing time is taken up by two essentially independent functions:

(1) Silhouette data acquisition, which is mainly a hardware function, and

(2) Silhouette data processing to obtain object features, which is primarily a software function.

At present these two operations occur one after the other since feature extraction cannot take place until a complete silhouette has been obtained. Thus, total processing time is the sum of times for each of these functions.

6.4.3.1 Silhouette data acquisition

In order to obtain a set of edge position co-ordinates, the processor hardware, in co-operation with the computer, scans the object under investigation in raster fashion.

The number of scans required to obtain a complete picture is a function of the object size and the vertical resolution of the scanning process. This quantity is thus irrevocably fixed.

The duration of each scan, however, is a quantity which is available for manipulation.

The scan consists of three parts:

(a) The Dummy scan, used to integrate a new line of video,
(b) The Real scan, used to process the integrated line, and
(c) The Delay period, which is used to allow time for the scanning table to reach the next position.

The above three parts are unequal in time duration as would have been expected.

Therefore, each scan in a picture frame is constituted as in Figure 6.11 below:

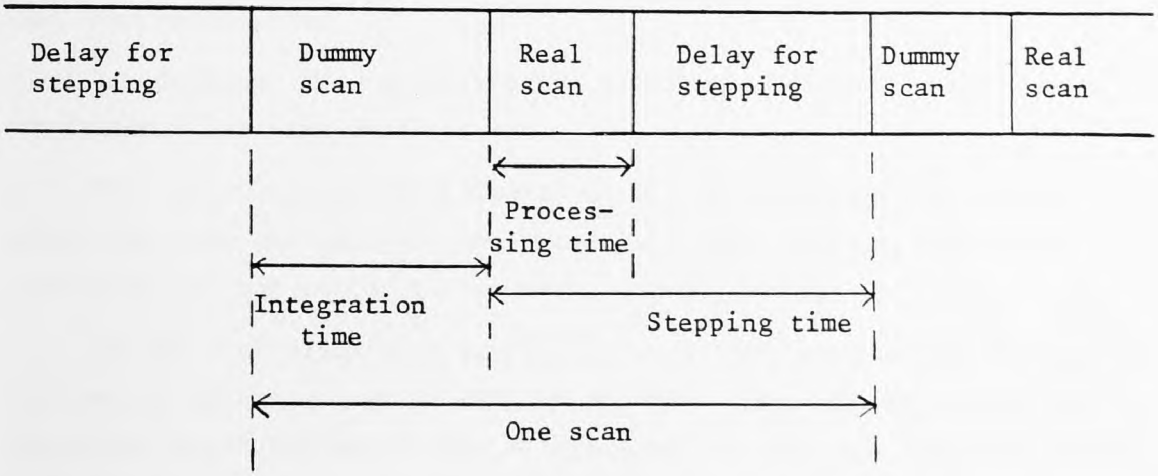


Figure 6.11: Scan cycle, constituent parts

The integration and processing time (dummy and real scans) takes up less than 50% of the time for one scan. The rest is just for moving the table.

So far as processing is concerned, the "delay for stepping" is dead time and represents the major hold-up during data acquisition. Since the

computer is fully occupied in performing stepping, nothing else can be done about it at present.

6.4.3.2 Silhouette data processing

This function, which is performed in software, consists of four distinct phases:

(a) Calculation of area,
(b) Location of centroid position,
(c) Extraction of object features, and
(d) Search through files, where other sets of features are stored as classes, to identify object.

The first two functions in the above sequence are calculated at the same time by "pipelining" the stored silhouette data through the appropriate software routine, i.e. the data are processed one scan at a time, the results of one pass being used as a data input for the next pass. This is a scheme which is well suited for hardware implementation.

Since calculation of the object features (radii) consists of calculating distances from the centroid to the edges of the object, this process cannot, therefore, start until the position of the centroid has been established.

In addition, the identification search cannot start until a set of features has been obtained.

The last two processing functions are thus performed serially after the area and centroid are found. In their present form these functions are not suited to hardware implementation.

Of the four processing functions, it is the calculation of features and search which take up by far the greatest time, and hence are the functions which one would like to speed up the most now that the actual scanning operation has become much faster.

6.5 Optimising the Present System

The two most serious holdups in the present system are:

(1) The speed at which the table can be stepped from one scan position to the next. This directly affects the rate at which data can be obtained for processing.

(2) The speed at which the computer can calculate a set of features for the object under investigation. This affects the overall processing speed.

In order to increase the system speed we can try to optimise further the present system. The approach clearly involves attacking the two main problems directly.

6.5.1 Stepping the table

The problem here appears to be fundamental in that there is a limit to how quickly a massive object (the moving table) can be accelerated and decelerated by the drive system currently used.

There are three possibilities for improving the situation:

(a) Change the drive system to a more heavy duty one.

(b) Reduce the mass of the table so that it can be more easily moved.

(c) Change the method of moving the table, so that instead of halting the table at each new scan position, the table is stepped continuously through a frame.

The brute force approach of the first method does not seem very promising.

The second method seems like a good approach, which would give improved performance with minimum system change. It is, however, not clear how big an increase in performance this would actually give.

The third method is probably the best approach, but it would involve some complication of the method of data acquisition.

With the present method, as we have seen, between scans, the table is stepped a definite distance from the previous position and halted while a scan is integrated. This ensures accuracy in vertical picture resolution and a stable picture for scanning.

If the table steps continuously, vertical resolution will have to be determined either by timing the distance between scans, or, more reliably, by counting stepping pulses.

Scan integration becomes a problem with this method. Ideally, one would like to integrate a scan in the time between stepping pulses to take advantage of the transient stability at this time. However, this is not likely to be sufficient time to obtain a satisfactory signal.

If this is found to be the case, then there will have to be some compromise accepted on the stability of the picture which is scanned.

6.5.2 Processing the data

Consideration of alternative approaches for data processing eventually comes down to a decision on how to divide the various system functions between hardware and software.

As previously discussed, the "find feature" and "search in file" functions are the most complex and not particularly suited to hardware implementation. They are also time-consuming and the only ones which cannot be performed in parallel with any other function. Of necessity, then, system timing will look something like the Figure 6.12, shown below:

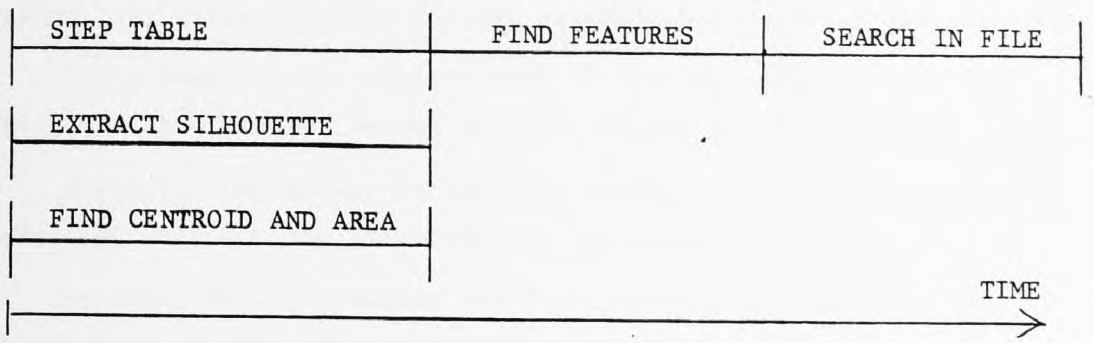


Figure 6.12: Optimum processing timing

In spite of the fact that we would like to minimise the time for "find features" and "search in file", the complexity of these functions necessitates a software approach.

The other arithmetic function presently performed in software is "find centroid and area". This could, in principle, be retained in software, since, according to the scheme of Figure 6.12, the computer could be otherwise unoccupied during silhouette extraction.

This would mean that the computer would not be used to step the table. This would become a hardware function, which is the most efficient way to do it, i.e. together with the extract of the boundary points of the silhouette shape.

At the moment there is no elegant system scheme, which can fundamentally reduce further the computation time than the present system.

At this point it can be said that the present system is not far from

the optimum already, particularly so far as data processing is concerned.

However, a purpose-built processor, using a multimicroprocessor system, might produce a more desirable result. However, the further complexity and expense might be a discouragement, at least for the near future.

6.6 Conclusions

After this phase of the work was completed, i.e. the development of the scanning system to an acceptable standard for an industrial application, it was thought that the effort was worthy from the point of view that the final scanning system, and in particular the design details contained within this thesis, represent a physically realised system that works and provides the services for which it was intended.

The present system fulfils most of the requirements that were initially laid down or became apparent during its development.

Further, theoretical background has been provided to account quantitatively for the behaviour of the system.

An important achievement was that the cost involved for the development of the scanning system was kept at a low level without any serious effect on the system as a whole, although it is thought that some extra consideration should be given to financing a more effective lighting source and further improvement in the construction of the scanning rig.

Finally, taking into account all sources of error and distortion, which were examined here, the system should be able to fulfil future requirements of much greater accuracy and speed than the ones required and achieved at present.

CHAPTER 7: PROCESSING ALGORITHMS

7.0 INTRODUCTION

In this chapter a systematic presentation is made of the algorithms used in the present work. The purpose of this presentation is to give an insight into the problems which this work has tried to solve.

The basic problem is the classification of the various patterns under test, on the basis of a suitable set of measurements. The method for choosing such a set is of critical importance for the solution of the problem.

In fact, the measurements should allow the extraction of all significant features from the shape under test, while discarding non-relevant ones, so that the recognition procedure becomes fast and reliable.

Until recent years, the word "algorithm" was unknown to most educated people; indeed, it was scarcely necessary. The rapid rise of computer science, which has the study of algorithms as its focal point, has changed all that; the word is now essential.

Generally, a computer program is the statement of an algorithm in some well-defined language.

Programs for numerical problems were written as early as 1800 B.C., when Babylonian mathematicians at the time of Hammurabi gave rules for solving many types of equations. The rules were stated as step-by-step procedures applied systematically to particular numerical examples. The word algorithm itself originated in the Middle East, although at a much later time. It comes from the last name of the Persian scholar, Abu Ja'far Mohammed Ibn Mûsâ ALKHOWÂRISMÎ, from the town of Khiva, whose text-book on arithmetic (about 825 A.D.) had a significant influence for many centuries.

There are several other words that almost, but not quite, capture the concept that is needed here: procedure, process, routine, method. Like these things, an algorithm is a set of rules or directions for getting a specific output from a specific input. The distinguishing feature of an algorithm is that all vagueness must be eliminated; the rules must describe operations that are so simple and well-defined that they can be executed by a machine. Furthermore, an algorithm must always terminate after a finite number of steps.

It is appropriate here to distinguish the different meaning of another, relatively new, term as the "software" is, which can very easily be confused with the term "algorithm".

Software is a general term for programming or compiling accessories used for computing or data processing systems. It covers both actual equipment and library routines. Algorithm, on the other hand, is only related to the solution of a specific problem.

Experience with computers has shown that the data manipulated by programs can represent virtually anything. Accordingly, the emphasis in computer science has now shifted to the study of various structures by which information can be presented, and to the branching, or decision making, aspects of algorithms, which allow them to follow one or another sequence of operations depending on the state of affairs at the time.

We can have, therefore, the so-called "processing algorithms" which, in effect, may work simultaneously with a machine which provides the data needed by the programs of the algorithms in order to get the desired outcome during an "on-line" operation.

It is, hence, the above features of algorithms that sometimes make algorithmic models more suitable than traditional mathematical models for the representation and organisation of ways to solve problems.

In this chapter, therefore, an attempt has been made to organise and represent all aspects of the present work, which are connected with the automatic on-line operation, with a distinct set of algorithmic models.

The first type of algorithm is related to the problem of data acquisition. Here, data are collected simultaneously during scanning of the samples by the scanning rig.

The second type of algorithm is connected with the manipulation of the data detected during scanning, in order to select a suitable set of measurements for the inspection operation.

The third, and final, type of algorithm deals with the way the actual inspection operation is controlled and carried out by the programs of the algorithm.

7.1 Data Acquisition Algorithm

Data acquisition is performed with the help of the first processing algorithm.

This algorithm consists of a computer program and its subroutines, which control the operation of the scanning rig. At the same time, the program activates the fast preprocessor of the system, which transforms the analog signal collected from the scanner into a digital one. Then, by hardware means, which are also controlled by the algorithm, the pre-processor transforms the digital signal into meaningful data, which are stored next in the memory of the computer for further processing. This operation is repeated, scan after scan, automatically until the sample under test is fully scanned.

Figure 7.1 below shows schematically the different steps through which the data acquisition algorithm is performed.

A low level, or assembly, language has been adopted for the writing of the program and its subroutines for the data acquisition processing algorithm. In this way, and in particular by using the so-called "MACRO" language, which is the most suitable for the PDP 11/10 computer used in the system, a faster and more effective on-line operation is achieved.

Special "flags" are used in the program, such as "start", "step", "get", "data" and "end" flags, which control the whole operation in an appropriate sequence, as shown in Figure 7.1.

An interrupt function is available (Figure 7.1) in the present algorithm, which ensures synchronisation between computer and pre-processor during data detection and data transfer periods.

The data produced by the pre-processor are the x co-ordinates of the detected edge points, their y co-ordinates and their type. The type of each point is expressed in terms of its relative position with respect to the boundary line of the shape under test along the line of scan from where the point is extracted, i.e. "left", "right", "up", or "down" type of point.

The x co-ordinate and type of each point on the boundary are readily available in binary form by the hardware pre-processor. The y co-ordinate is taken for each point according to the value of the special counter, which is set by the algorithm. This counter is incremented by one each time the scanning table is advanced by one scanning step.

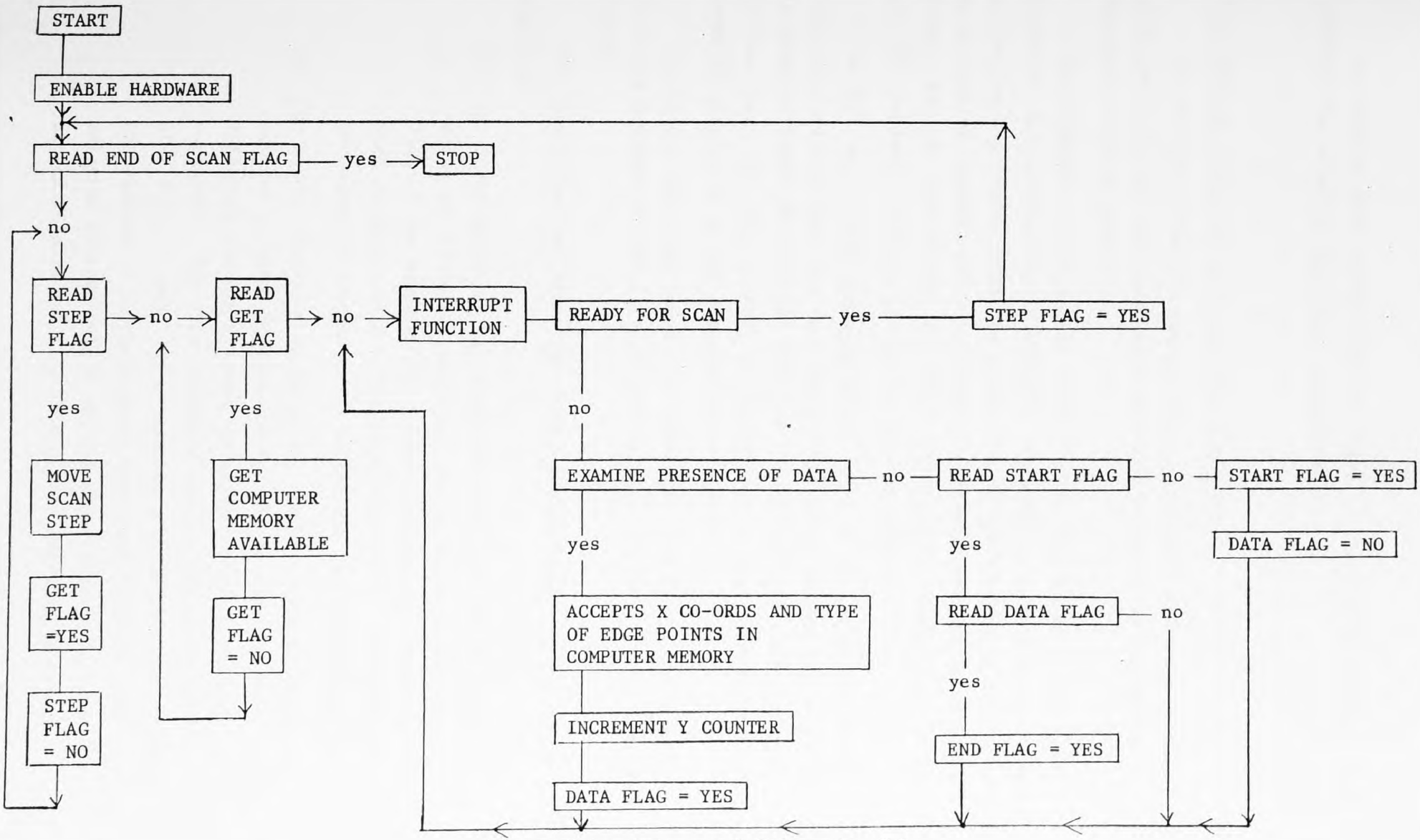


Figure 7.1: Data acquisition algorithm

The program also senses when the shape has been fully scanned, and inhibits the scanning operation automatically.

7.2 Feature Extraction and Selection Algorithm

The feature extraction and selection algorithm works in parallel with the previously described data acquisition algorithm during the automatic on-line operation.

The programs for the second algorithm are written in a high level language, and particularly in FORTRAN. This was requested by the company sponsoring the present work, in order to have the processing of the data in a computer language which is acceptable to most computers available today. For the same reason, the third algorithm for inspection, which will be examined later on, is written also in FORTRAN.

In this way, if the scanning system is to be used with any other computer facility, only the data acquisition algorithm need be adjusted in order to adapt the computer with the pre-processor of the scanning system, since the pre-processor is designed to be able to obey certain commands attainable by any computer system used.

The second algorithm comes into effect as soon as the first data, which are acquired by the first algorithm, are available in the computer memory.

Two operations, one after the other, are performed by the second algorithm:

(1) The first operation is to extract a feature set for each class of silhouette patterns, and to store it on a disc file of the computer for further processing. This operation is also called a "learning" operation. One pattern per class is assumed to be sufficient for identification.

(2) The second operation is to select the necessary feature set per pattern, in order to be able to separate the different pattern classes, by using a minimum distance technique. For this operation the content of the above disc file is used, which contains all the features of all the classes of patterns which have been scanned. The selected feature set of all the above classes is then stored in a second disc file of the computer, in order

to be used in the automatic inspection operation of the present system.

Figure 7.2 shows schematically the different steps performed by the second processing algorithm.

7.2.1 Feature extraction program

The features to be extracted are the area and a number of radii from the centroid to the boundary of each pattern class. Thus, the extraction of the centroid co-ordinates is essential.

The above features are extracted by using the co-ordinates of the edge points and their corresponding type, which are readily available by the pre-processor during scanning. The "left" and "right" type of points are used to compute the area and the centroid, while all types of points, including "up" and "down" ones, are used to locate the radii. The area and centroid are computed simultaneously during scanning, and the radii as soon as all edge points are detected.

The program plots on the storage display the silhouette of the pattern under test with its radii features if it is requested interactively through the keyboard. It also returns the scanning table automatically to the starting position at the end of scanning, if it is requested, again interactively through the keyboard.

7.2.1.1 Extraction of area

The area is extracted by accumulating the area elements (in cells) of the shape under test along every line of scan, until the shape is fully scanned.

Every time a "left" point is found along a scan line, it is paired with the next "right" edge point on that line, and the actual number of point cells between them is computed, i.e. $(R_{1j} - L_{1j}) + 1 = A_{1j}$, where R_{1j} is the first "right" edge point on the jth line of scan, L_{1j} is the first "left" edge point on the jth line of scan, and A_{1j} is the area in point cells between R_{1j} and L_{1j}. If another pair of edge points exist on the same line of scan (i.e. the shape has re-entrants), then this area element is also computed $(R_{2j} - L_{2j} + 1 = A_{2j})$. At the end of the line of scan its total area in cells inside the silhouette shape will be A_j, where $A_j = A_{1j} + A_{2j} + \ldots$, for the jth line of scan.

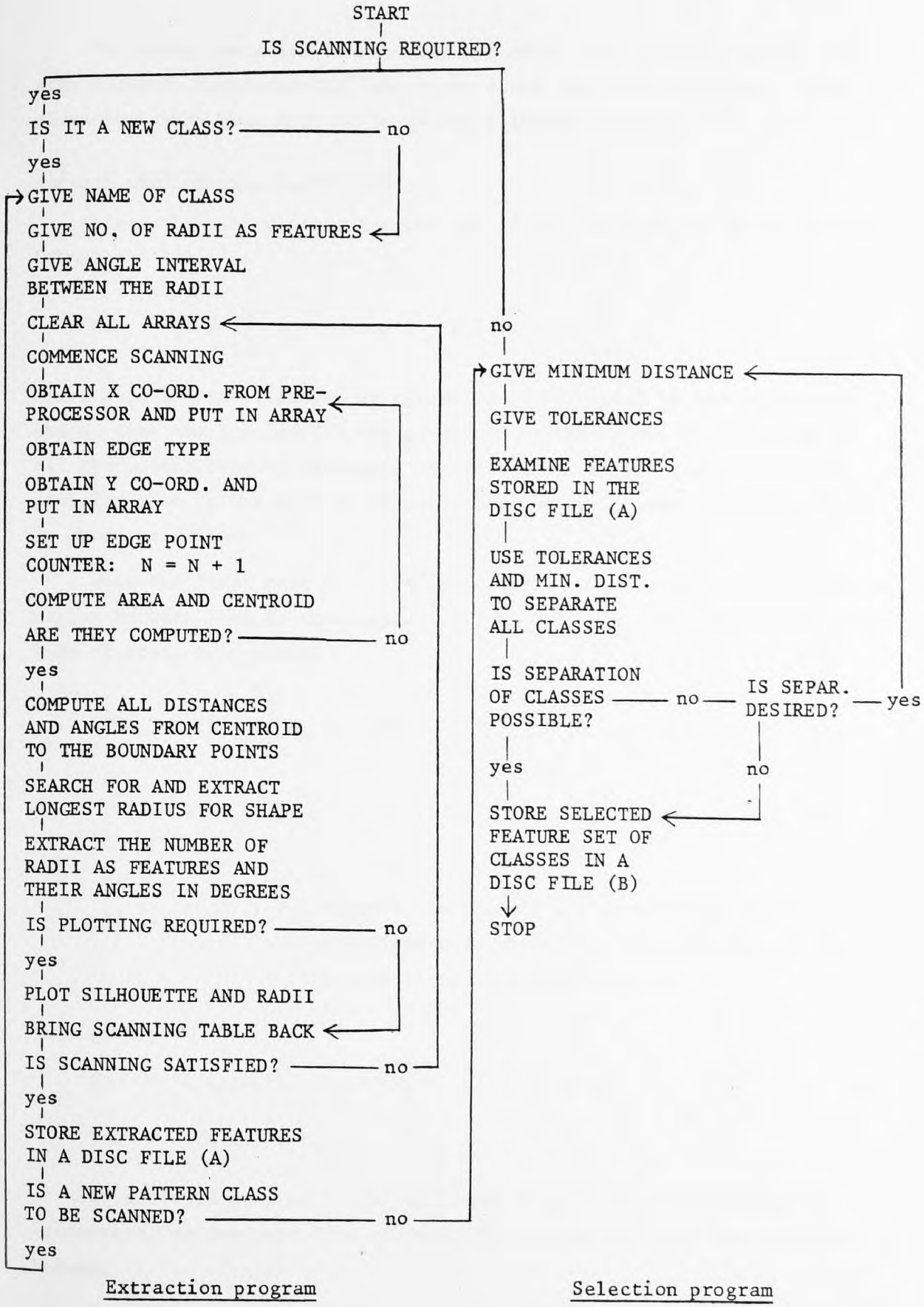


Figure 7.2: Feature extraction and selection algorithm

The above operation is repeated for every line of scan and all the area elements found for the individual scans are added together. This gives the total area A in cells of the silhouette shape.

7.2.1.2 Extraction of centroid

Since the co-ordinates Xco and Yco of the centroid are given by the formulae:

$$Xco = 1/A \sum_{i=1}^{A} x_i \quad \text{and} \quad Yco = 1/A \sum_{i=1}^{A} y_i$$

where A is the total number of points (area in cells) in the silhouette shape, then the Xco and Yco are extracted by adding the co-ordinates of all the points between each pair of "left" and "right" edge points and dividing them by the area cells each time area cells are computed. This is shown as follows:

When the first pair of "left" and "right" edge points are encountered having between them A_1 area cells (i.e. $A_1 = R_1 - L_1 + 1$) along the first line of scan, this makes:

$$Xco = 1/A_1 \sum_{i=1}^{A_1} x_i \quad \text{and} \quad Yco = 1/A_1 \sum_{i=1}^{A_1} y_i$$

$\sum_{i=1}^{A_1} x_i = A_1\{2.L_1 + (R_1 - 1)\}/2$, i.e. an arithmetic series, and

$\sum_{i=1}^{A_1} y_i = A_1 \cdot Y_1$, where Y_1 is the first line of scan, which indicates here also the y co-ordinates of all points along this line of scan.

This gives:

$$Xco = 1/A_1\{A_1(2 \cdot L_1 + (R_1 - 1))/2\}, \text{ and}$$

$$Yco = 1/A_1(A_1 \cdot Y_1).$$

If a second pair of "left" (L_2) and "right" (R_2) edge points are encountered on the same line of scan, the co-ordinates of the centroid become:

$$Xco = 1/(A_1 + A_2)\{A_1(2.L_1 + (R_1 - 1))/2 + A_2(2.L_2 + (R_2 - 1))/2\},$$

and $$Yco = 1/(A_1 + A_2)(A_1 \cdot Y_1 + A_2 \cdot Y_1).$$

When there is not another pair of "left" and "right" edge points on the first line scan and the scanner moves to the second line of scan, then, after the next "left" (L_3) and "right" (R_3) pair of edge points are encountered, we have:

$$Xco = 1/(A_1 + A_2 + A_3)\big(A_1(2.L_1 + (R_1 - 1))/2 + A_2(2.L_2 + (R_2 - 1))/2 + A_3(2.L_3 + (R_3 - 1))/2\big), \text{ and}$$

$$Yco = 1/(A_1 + A_2 + A_3)(A_1.Y_1 + A_2.Y_1 + A_3.Y_2)$$

(where Y_2 is the y co-ordinates for all points in the silhouette along the second line of scan).

The above computation is carried on until the sample is fully scanned, at which time the correct values for the co-ordinates of the centroid are extracted.

7.2.1.3 Extraction of radii

Having first extracted the co-ordinates of the centroid of the sample under test, and having readily available the co-ordinates of all the edge points of the shape, all the distances from the centroid to the edge points and their angles with respect to the x-axis which passes from the centroid are computed.

For example, for the edge point (x_i, y_i), we have its distance d_i and angle a_i computed as follows:

$$d_i = \big((x_i - Xco)^2 + (y_i - Yco)^2\big)^{\frac{1}{2}}, \text{ and}$$

$$a_i = \tan^{-1}\big((x_i - Xco)/(y_i - Yco)\big).$$

The above results are computed in the same order as the edge points are detected, keeping at the same time a special boundary points counter in the program until all these points N, say, are encountered.

During the above computation the program of the algorithm compares also every distance with the next one, and keeps the longest one between the two for comparison with the following computed distance. This operation is repeated until all distances are computed, and, thus, the longest one is extracted.

If more than one equal longest distance exists, the program locates

their angles and selects the one which is at a media position between those angles.

The extracted longest radius and its angle is then used as a datum for the extraction of the number of the radii of the shape which are to be used as features. These radii are extracted at specific angle intervals, measured anti-clockwise from the location of the longest radius.

Whenever no point exists on the boundary at any one of the chosen angles, then a search is carried out to find the nearest edge point to the desired angle position.

7.2.2 Feature selection program

The program for feature selection examines the features for each class, which are stored in a disc file during the previous feature extraction phase, and selects the necessary feature set, in order to separate the different pattern classes, by using a minimum distance technique.

Firstly, two scale factors are chosen and supplied interactively through the keyboard, one for the area and the other for all the radii which are to be selected as features. Thus, a degree of tolerance is introduced into the measures. Each feature is then expressed in tolerance cells.

For example, if a feature has a length of 42 millimetres and the tolerance for the feature is 5 millimetres, the size of the feature is 8 when expressed in tolerance cells.

In other words, cellularisation is carried out by making the size of each cell 5 instead of 1.

The tolerances determine the degree of accuracy which is desired for the inspection operation of the system. Also, they provide smaller numbers for the features, which occupy less computer memory, and make faster the computations involved.

Secondly, a minimum distance is chosen and supplied interactively also through the keyboard. The use of the minimum distance allows only the minimum number of features necessary to be retained for the inspection, as will be seen in the following paragraph.

The feature data are presented in a table, the Ith row of which contains measured values of a particular feature (e.g. area or radius) for all classes, and the Jth column of the table contains all features measured for a particular class.

The features are examined by the program in the order in which they appear in the feature table, i.e. the order in which they were gathered on the disc backing store.

A feature set provides unique identification only if all columns differ mutually in at least one entry. In this case a minimum distance one is said to be necessary.

So, the program checks the columns against one another to determine if they indeed differ in at least one entry. If not, the next feature in the table is added (e.g. next row), and so on until a usable set is obtained. The features not needed are made redundant.

However, because, in practice, erroneous measurement may occur, e.g. for some patterns which at certain orientation may produce several similar features with other patterns, which is also enforced with the cellularisation, some controlled redundancy into the feature set is necessary, i.e. it is required that all columns in the feature set chosen differ in at least, say, two entries instead of one. In this case, the features would be separated mutually by a minimum distance 2.

If less redundancy of features is necessary, a higher minimum distance is incorporated. Area and longest radius are always selected first.

To illustrate the way the feature selection program works, an example is given in Figure 7.3. Suppose we have 4 pattern classes and 4 features are extracted for each class during the "learn" phase of the algorithm (see Figure 7.3(a)). The table resulting after cellularisation is also shown in Figure 7.3(b).

By using a minimum distance one, the table shown in Figure 7.3(c) results. Here, one feature only is necessary per class to separate all classes.

By using minimum distance two, the table shown in Figure 7.3(d) results. Here, three features per class are now necessary to separate all the classes.

From Figure 7.3 below, it is therefore apparent that with minimum distance 3, all four features for each class will be necessary in order

to separate the classes. However, with minimum distance 4, separation will not be possible since the features on the second row are the same for more than one class, which makes all columns differ mutually in only three entries instead of four. In this case, more features should be available per class to ensure separation.

(a) Candidate feature measures (typical)

Feature	Class 1	Class 2	Class 3	Class 4
1	7543	7302	8010	7905
2	153	157	151	144
3	250	209	302	127
4	61	85	29	79

(b) Candidate feature measures after cellularisation

Feature	Class 1	Class 2	Class 3	Class 4	Cell size
1	75	73	80	79	100
2	15	15	15	14	10
3	25	20	30	12	10
4	6	8	2	7	10

(c) Necessary feature set selected with minimum distance 1

Feature	Class 1	Class 2	Class 3	Class 4
1	75	73	80	79

(d) Necessary feature set selected with minimum distance 2

Feature	Class 1	Class 2	Class 3	Class 4
1	75	73	80	79
2	15	15	15	14
3	25	20	30	12

Figure 7.3: Feature selection processing

7.3 Inspection Processing Algorithm

The third processing algorithm performs the inspection operation of the present system.

With the above algorithm an unknown pattern is recognised, or identified, as a member or not of one of the pattern classes, which are stored already in a disc backing store by the second processing algorithm as was examined in the previous paragraph.

The inspection is performed by the programs of the inspection algorithm in two stages:

First, the pattern, which is to be identified, is scanned and a number of features extracted.

Secondly, a feature set for the pattern under test is appropriately selected, and it is compared with the feature sets of all pattern classes of the second algorithm of the system.

7.3.1 The scanning and feature extraction for the pattern under test

The program for scanning the unknown pattern and for extracting a number of features, which are to be used for its recognition, works in parallel with the data acquisition processing algorithm in a similar fashion as does the second processing algorithm.

The area, the longest radius and a number of radii around the boundary of the above pattern are also extracted in a similar way to the one used by the second algorithm.

The main differences here are:

(a) No name of class is given to the pattern which is to be identified.
In the case of the pattern classes a class name is given to each of them at the beginning of their scanning.

(b) The extracted feature set for the pattern under test is stored in a separate disc file from the one where the pattern classes are stored by the second algorithm.
In the above new disc file created by the third algorithm, only one set of features is at any time present, i.e. only the features of the latest pattern to be identified. The features stored previously on this disc file for any other

unknown pattern, which has been examined, are becoming redundant as soon as a new pattern is put under the inspection test.

In the case of the pattern classes, on the other hand, new feature sets may be added at the corresponding disc file of the second algorithm whenever the addition of a new pattern class is required.

7.3.2 Identification program for pattern under test

Figure 7.4 below shows schematically all the steps which the program performs for identification of an unknown pattern.

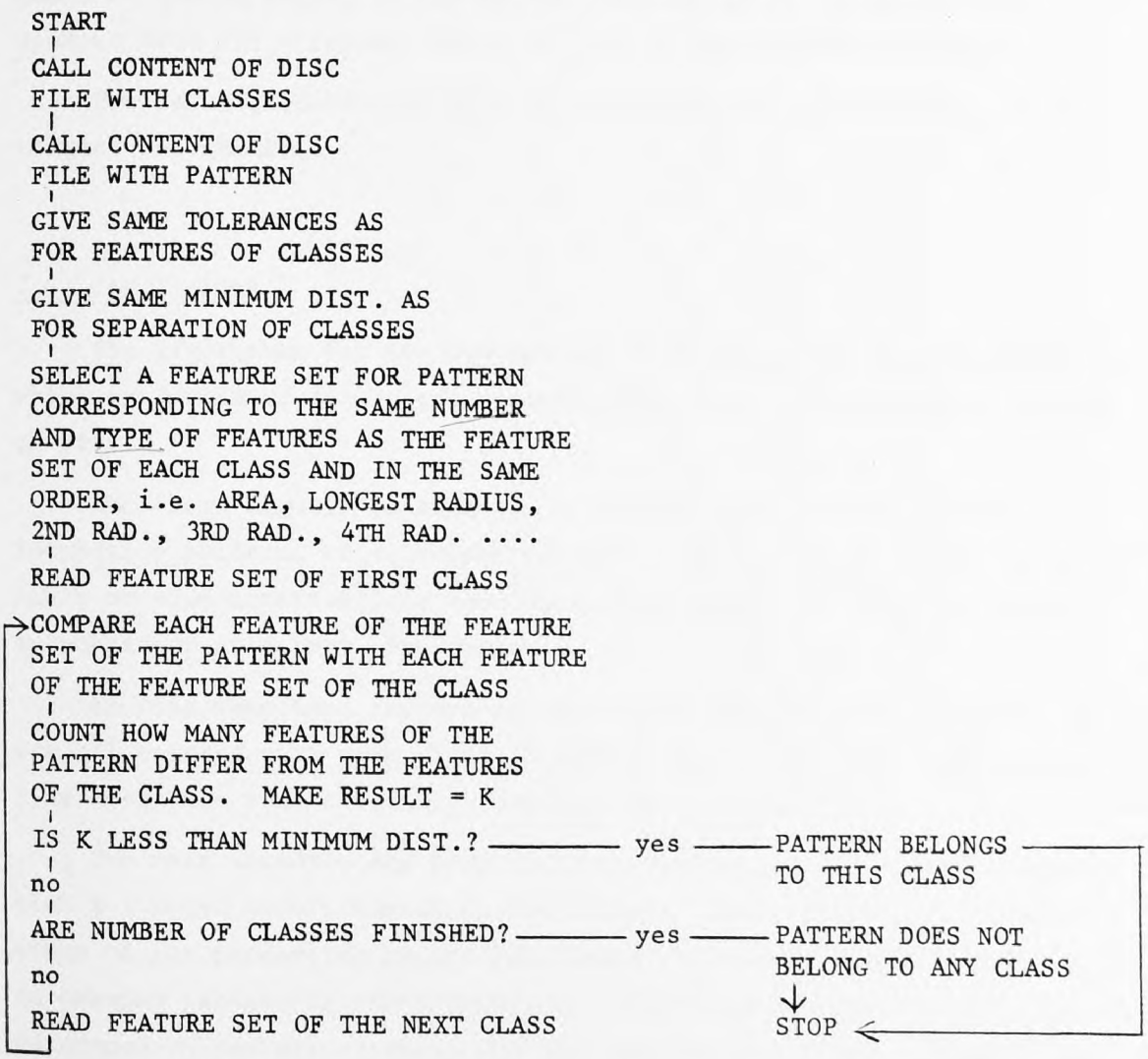


Figure 7.4: Program for identification of pattern under test

When the identification of a pattern under test is not possible, this may be caused mainly by two reasons. Either the number of features

is not sufficient, or the tolerances used are not "tight" enough.

In the first case, the minimum distance should be changed by using the second algorithm without performing any scanning, i.e. by performing instead only the second operation of the algorithm with a different minimum distance.

In the second case, the above should be repeated, but instead of changing the minimum distance, to change only the tolerances. The tolerances should be made smaller, i.e. smaller cell size.

If, after both the above alterations, the system fails to produce recognition when the pattern is put to test, then either the pattern does not indeed belong to any of the classes, or it is damaged, or altered from its original state, or even it was recorded wrongly.

The result, therefore, will be rejection, or substitution, if not correct recognition.

7.4 Conclusions

The algorithms for the new method for shilhouette shape recognition, which is developed during the present work, have been presented in this chapter.

The above algorithms succeed in producing an on-line operation for inspecting patterns of different pattern classes with as little as possible outside intervention, something which makes the whole operation automatic up to a very high degree.

Another important feature of the algorithms described is that they are all related with each other in such a way as to ensure that the on-line operation runs smoothly, fast and effectively.

The main disadvantage here is that, having to use a small computer with a limited memory capacity, the "select" and the "inspect" operations of the processing algorithms need to be run separately every time an unknown pattern is under test and a different minimum distance or tolerance values are required for its recognition. With a bigger computer, such as a PRIME 550 type, the above problem should not exist.

CHAPTER 8: EXPERIMENTS AND EVALUATION

8.0 INTRODUCTION

After the development of the methodology, the design and construction of the scanning system and its fast preprocessor and the development of the algorithms, which were examined in detail in the previous chapters of the present thesis, it was necessary also to evaluate the total system in order to determine its capability and to locate possible weaknesses.

Thus, a programme of experimentation with a systematic evaluation were carried out in order to fulfil the following objectives:

(a) To determine the accuracy with which orientation may be measured.

(b) To determine the accuracy and repeatability with which features could be measured.

(c) To determine the number of features for recognition as a function of the minimum distance used, the tolerance cell size and the total number of patterns learned as classes.

(d) To determine how the method used for identification may be optimised for its practical application.

Two types of experiments have been performed: an "off-line" experiment and an "on-line" experiment, together with a performance prediction and optimisation analysis for the present inspection system. These are presented in this chapter.

The first type of experiment was a second simulation study, different from the one described in chapter 5 of this thesis, using an enhanced version of the previous package for the computer. Here, with the use of a smaller computer (a PRIME 550 type, instead of the large CDC 7600 computer), an off-line series of experimental evaluation was performed by generating again mathematically defined silhouette shapes on a grid of cells, at a different orientation each time with respect to the grid axes, in order to determine particularly the ability to measure orientation accurately.

The second type of experiment was a series of automatic on-line inspection operations with the use of the complete scanning system developed

for the present work. Here, 100 different real samples were chosen and examined in order to provide a reasonable test of the above system. The primary task was to provide a convincing evaluation which could determine the accuracy with which the inspection operation may be performed with the method developed in the present work.

Further, a statistical analysis was carried out in order to show that prediction and optimisation of performance is possible, using an analysis based on binomial statistics, given certain information obtainable from the experimental measurements of the above 100 different real patterns.

8.1 Off-Line Experiments and Evaluation

In the off-line experiments ellipses were generated, with the semi-major axis, NA, (i.e. the longest radius in this case) at a specified angle to the axes of the grid of cells, and with different values of eccentricity, EC, for different experiments.

Additionally, the radius of curvature, R, of the ellipse was used as an independent variable in the above experiments, since it appears to be a parameter which, besides the length of the longest radius, is also significant in determining measurement accuracy. R is given by the formula:

$$R = NA(1 - EC^2).$$

8.1.1 Orientation and position determination

As to the accuracy with which orientation may be measured, the first type of experiment was based on the following intuitive idea. Consider the shapes to be defined on a grid of K x K cells. As the shape is rotated through 360 degrees, its longest radius can fall into one of only π x K different cells, thus, only π x K distinguishable orientations can be resolved, distributed roughly uniformly around a circle of radius K/2.

Actually, this is an upper bound, since squares at 45 degrees and 135 degrees from the axes of the cell grid will appear to be increased in size by 40%, thus fewer of them are required to fill the circle.

This approach suggests that the angular resolution should decrease at around 45 degrees, 135 degrees, ..., and also that for a shape, the longest radius of which is 100 cells long, for example, the boundary swept

out as the shape is rotated has no more than 628 (i.e. 100 x 2π) distinguishable points.

Thus, positioning even to one-half degree is not possible, since this would require 720 points (i.e. 360 + 360).

Uncertainty in determining the exact longest radius degrades this precision still further, though it is difficult to determine by calculation how big this degradation will be.

The orientation as measured differs from the true orientation A by an error dA, since the tip of the longest radius is assigned to the centre of the boundary cell which is most distant from the centroid.

Figure 8.1 illustrates this problem for a shape having a longest radius of 10 cells long.

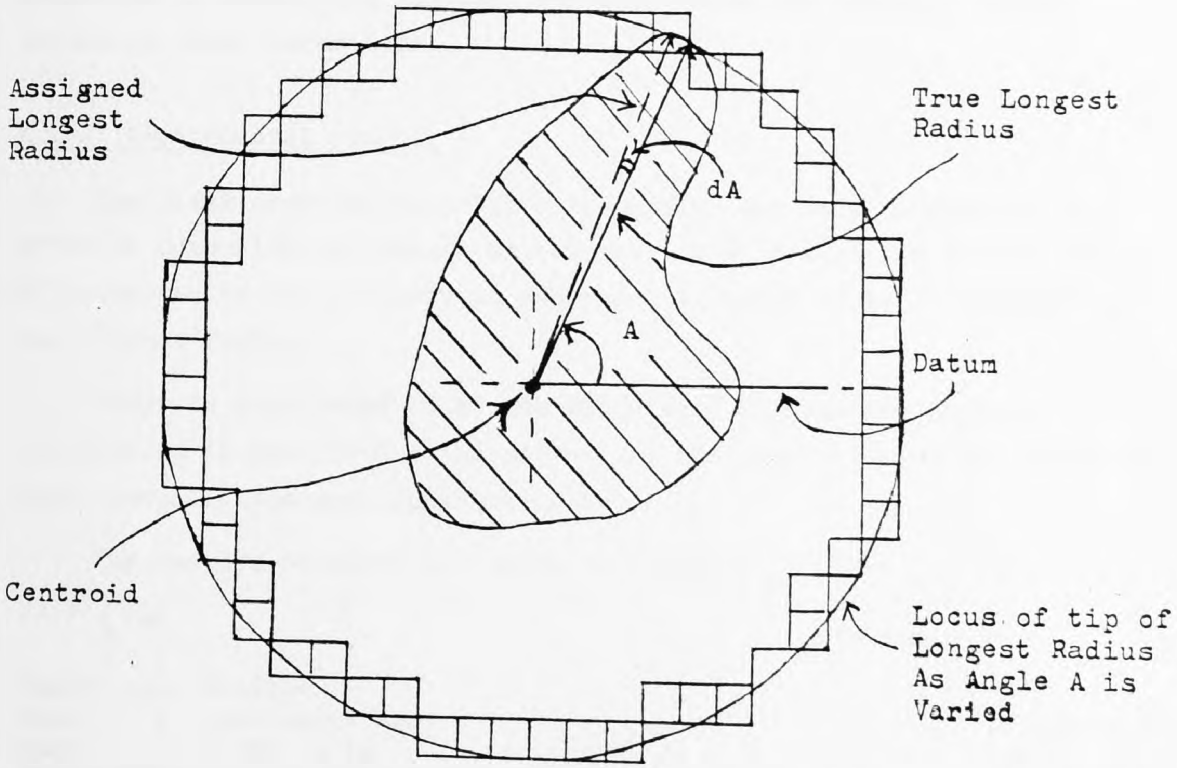


Figure 8.1: Determination of orientation

Counting shows that there are only 56 possible boundary cells (representing distinguishable orientations) against an expected 62 (i.e. $2\pi \times 10 = 62.8 \approx 62$).

The orientation is, however, computed from the expression:

$$A = \arctan(Y/X),$$

where X and Y are the co-ordinates measured in cells from the centroid to the boundary cell at the longest radius.

The analysis using differences, presented in chapter 4 of this thesis, shows that in this circumstance the accuracy with which the orientation may be determined is equal at 0 degrees, 45 degrees and at 90 degrees, and at integer multiples thereof. This does not, however, invalidate the upper bound to accuracy of orientation desired above.

The additional degradation caused by uncertainty in determining precisely which boundary cell is the extremity of the longest radius has been determined by the experimental results, as described next, where area, position of centroid and orientation of ellipse silhouette were estimated by simulating the same way with which the scanning system estimates these parameters.

8.1.2 Experimental results

The first experiments which were carried out were to measure the error in measuring the angles of orientation as a function of the radius of curvature of the boundary at the point at which it is intercepted by the longest radius.

Error is considered to be the difference between the angle of orientation as specified originally (i.e. the true orientation) minus the angle actually measured experimentally.

The results obtained are shown in Figure 8.2 below:

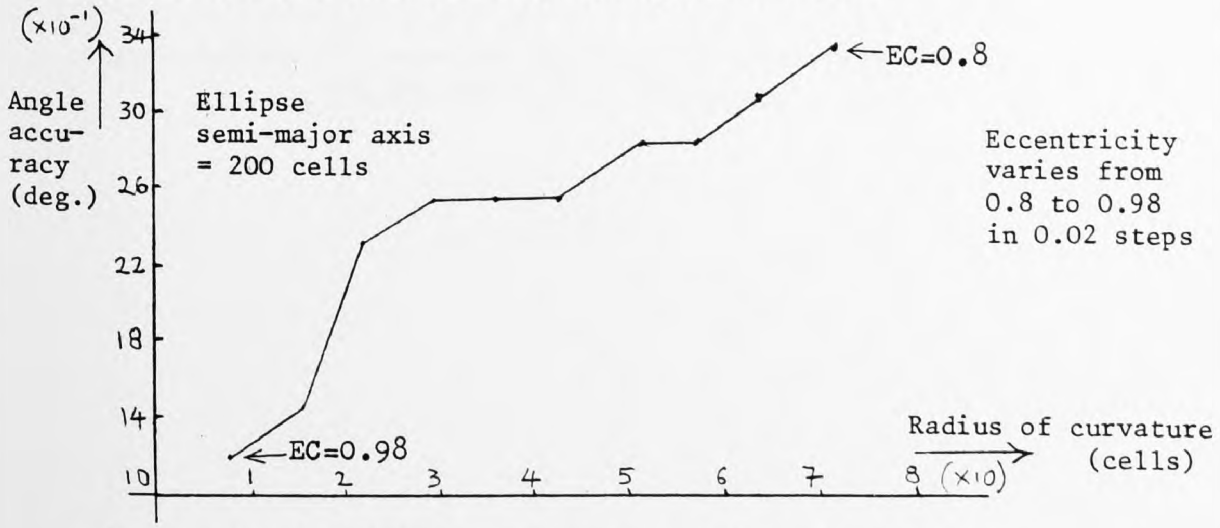


Figure 8.2: Orientation error versus radius of curvature

Since in the on-line operation the shapes will be presented at any orientation, and the accuracy may also be affected by the location of the centroid with respect to the quantisation cell grid axes, error should more properly be investigated statistically, with more orientations taken into account.

Hence, more experiments were performed, in which the ellipse was positioned at a selection of angles uniformly spaced within the range 0 to 360 degrees, with the centroid firstly at the centre and secondly at the corner of a resolution cell.

The results are shown as histograms in Figures 8.3(a) and 8.3(b) below:

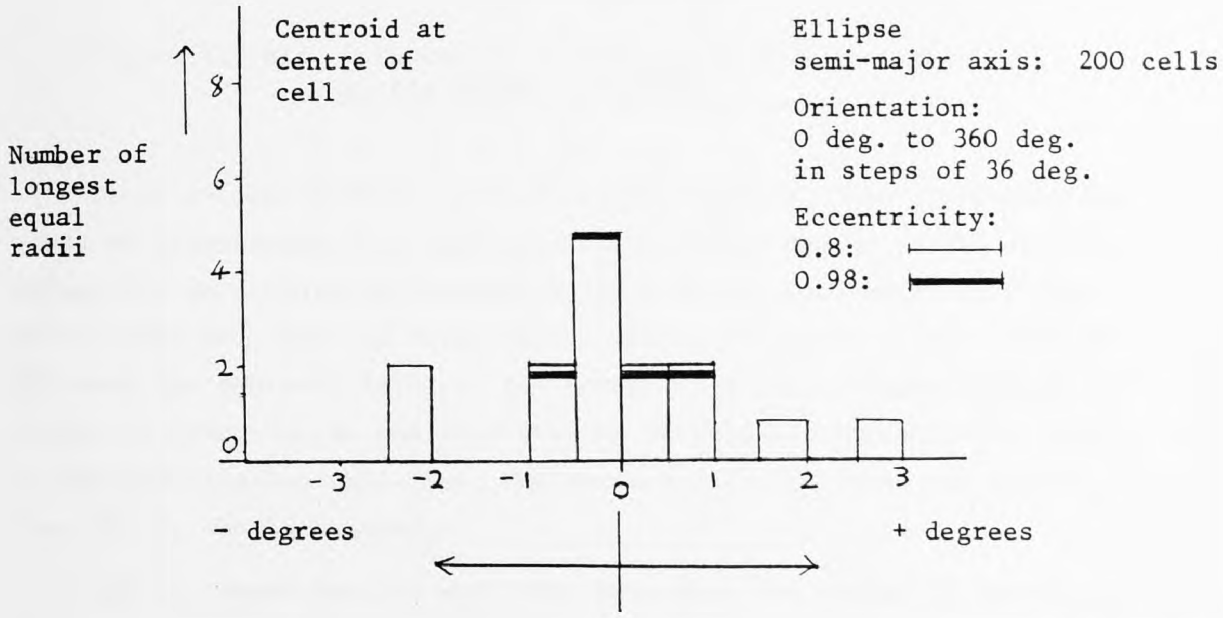


Figure 8.3(a): Orientation determination with centroid at the centre of a cell

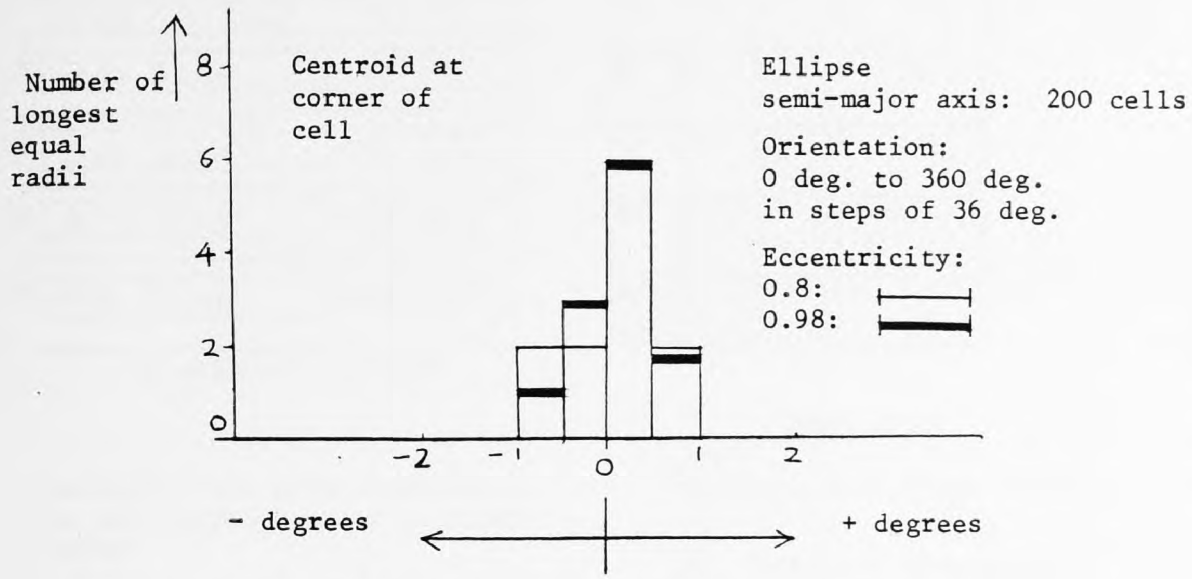
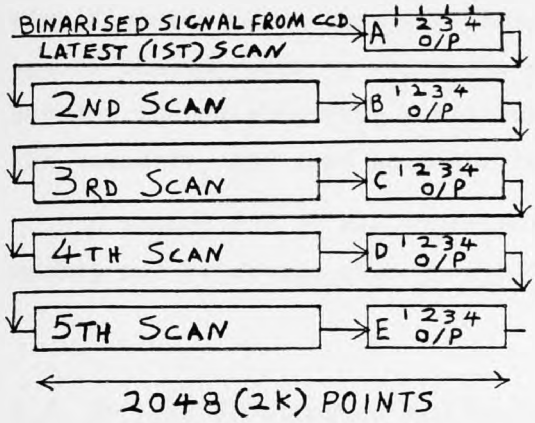


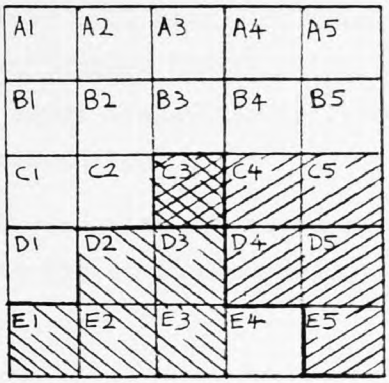
Figure 8.3(b): Orientation determination with centroid at the corner of a cell

It is evident from Figure 8.3(a), for example, that 63.6% measurements of orientation will fall within $\pm$ one-half degree of the correct value, for an ellipse with eccentricity 0.98 and semi-major axis 200 cells, when the centroid falls at the centre of a cell. This rises to 85% when the centroid falls at the corner of a cell (Figure 8.3(b)). Since, in practice, the centroid will be positioned randomly with respect to the quantisation cell axes, the average value of these two values, i.e. 69.3%, should be used.

All the above results show that even when the radius of curvature is small (i.e. with EC = 0.98), the error in measuring orientation is likely to be between two and three times that predicted theoretically in chapter 4 of this thesis, from consideration of the longest radius size as expressed in cells.

The solution to this problem might lie in using boundary points at which there is a sharp corner. These points can be detected almost instantaneously in hardware, using masks, say, 5 cells by 5, as shown in Figure 8.4 below:





MASK ARRAY

Cells in Mask Array correspond to designated outputs in binary store

Indicates sharp corner

Indicates same corner turned through 90 deg.

Figure 8.4: Detection of sharp corners by masks

The mask is convolved with the binarised image,and logical operations which are an extension of those used currently for detecting boundary points applied to detect corners. This procedure would, incidentally, also be able to detect and selectively eliminate false points outside the boundary, along the line of scan, due possibly to any presence of excessive dust.

However, the mask possibility has several disadvantages:

Firstly, many shapes will have more than one "sharp" corner, whilst others will have none at all. The longest radius will not, in general, end at a sharp corner, so some means must be provided for specifying (economically) which sharp corner is to be used as datum.

Further, a 5 x 5 cell binary mask can contain 2^{25} (= 33, 554, 532) distinguishable binary patterns. Although most of these will be irrelevant, including, for example, those which contain less than 5 or more than 10 cells at level "1", or for which the centre cell is not a "1", many thousands of possibilities will have to be considered. The sharpness of corners may vary from 45 to 135 degrees, and a corner may appear at any orientation.

Thus, sharp corner detection is not recommended for measuring orientation and hence positioning of longest radius.

The best solution for obtaining an accuracy of orientation not significantly worse than the limit set by cell size is found to be the use of the approach by which the median longest radius point is selected between the neighbouring points where equal longest radii terminate on the boundary, and used as datum for orientation.

A second run of the experiments showed that the above approach works, and it should also be applied to the on-line operation.

8.2 On-Line Experiments and Evaluation

The on-line experiments, which were carried out during the second phase of the experimentation, involved the learning of 100 real shapes which were selected to include samples introducing most of the expected difficulties, such as extremes of size, centroid outside the boundary, symmetry, or similarity.

The above 100 patterns were supplied by the company sponsoring the work, some examples of which are shown in Figures 8.5(a) to 8.5(h), where the samples are presented using a 1:1 scale.

During the above experiments, the scanning system which has been developed was able to scan samples in about 4 seconds, but still required about 2 minutes for recognition. Whenever plotting was also required, a further 2 minutes were necessary.

The recognition phase was slow only because Fortran programs were used, on the slow(e.g. 1 millisec per floating point multiplication) PDP 11/10 computer of standard architecture which was incorporated.

The resolution cell size was 0.35 mm (i. e. seven times the single step of the stepping motor), determined by the limited (24K word) random access memory available in the computer, which prevents the finest resolution of the scanning rig, 0.05 mm, from being used.

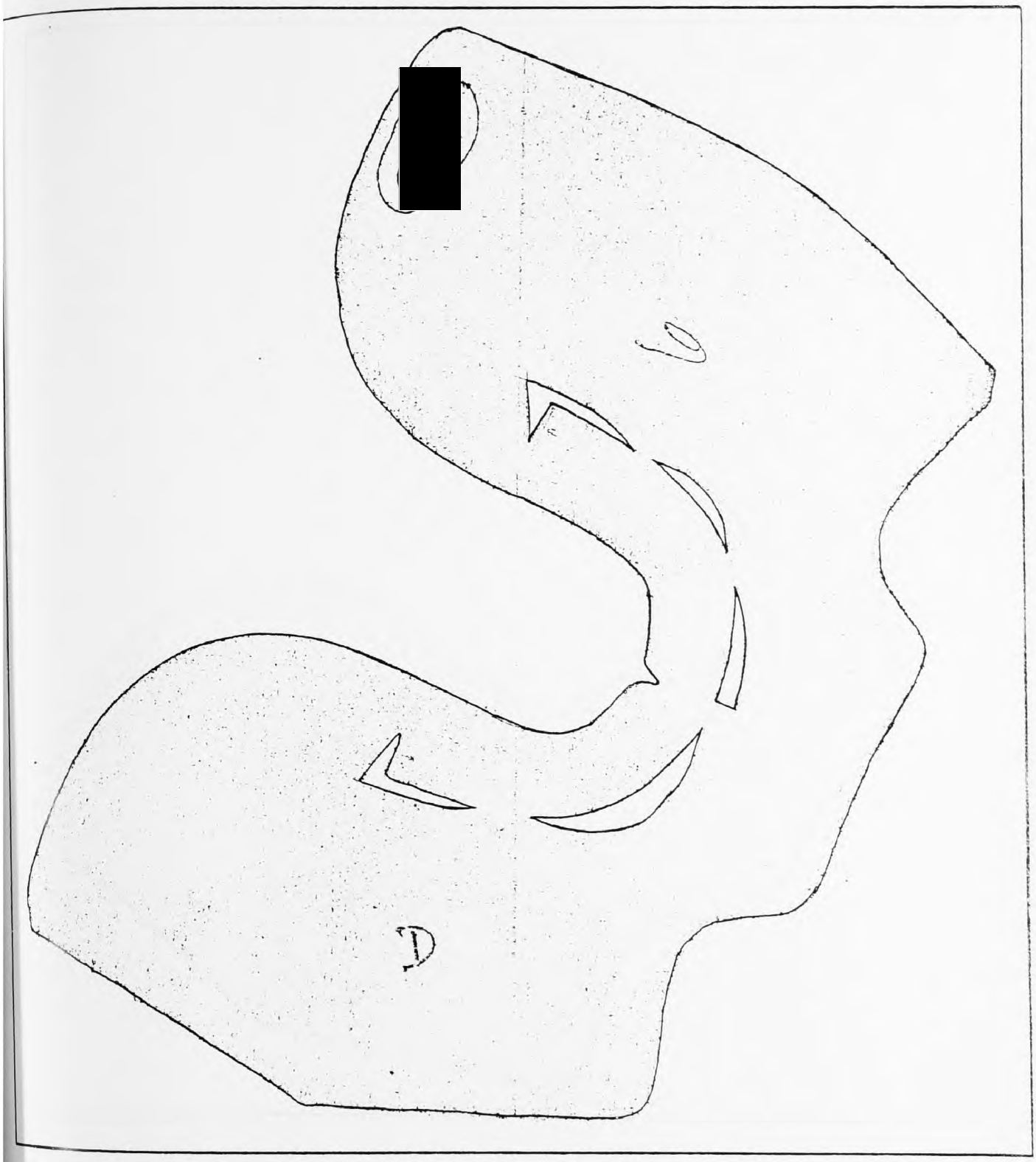


Figure 8.5(a)

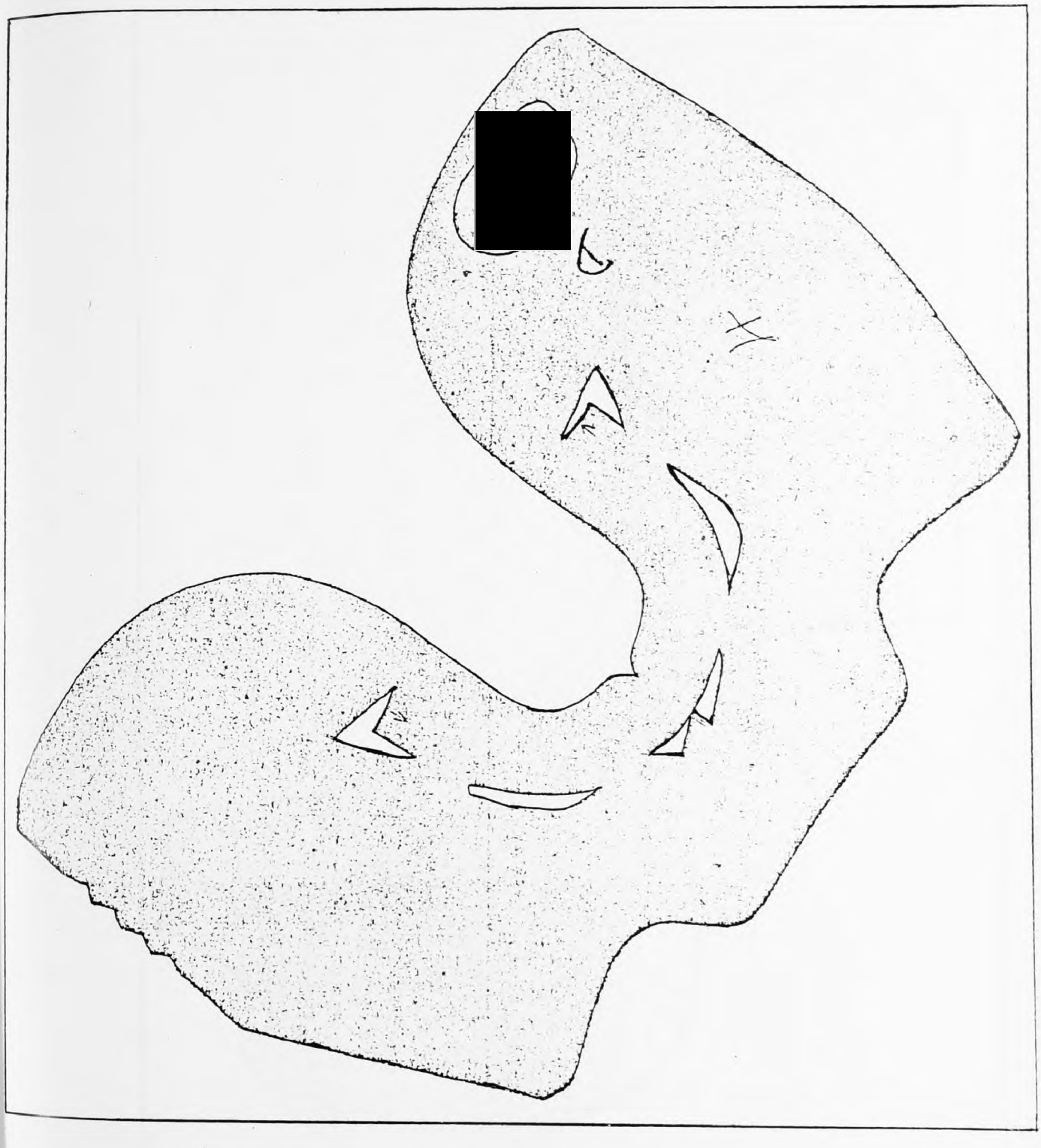


Figure 8.5(b)

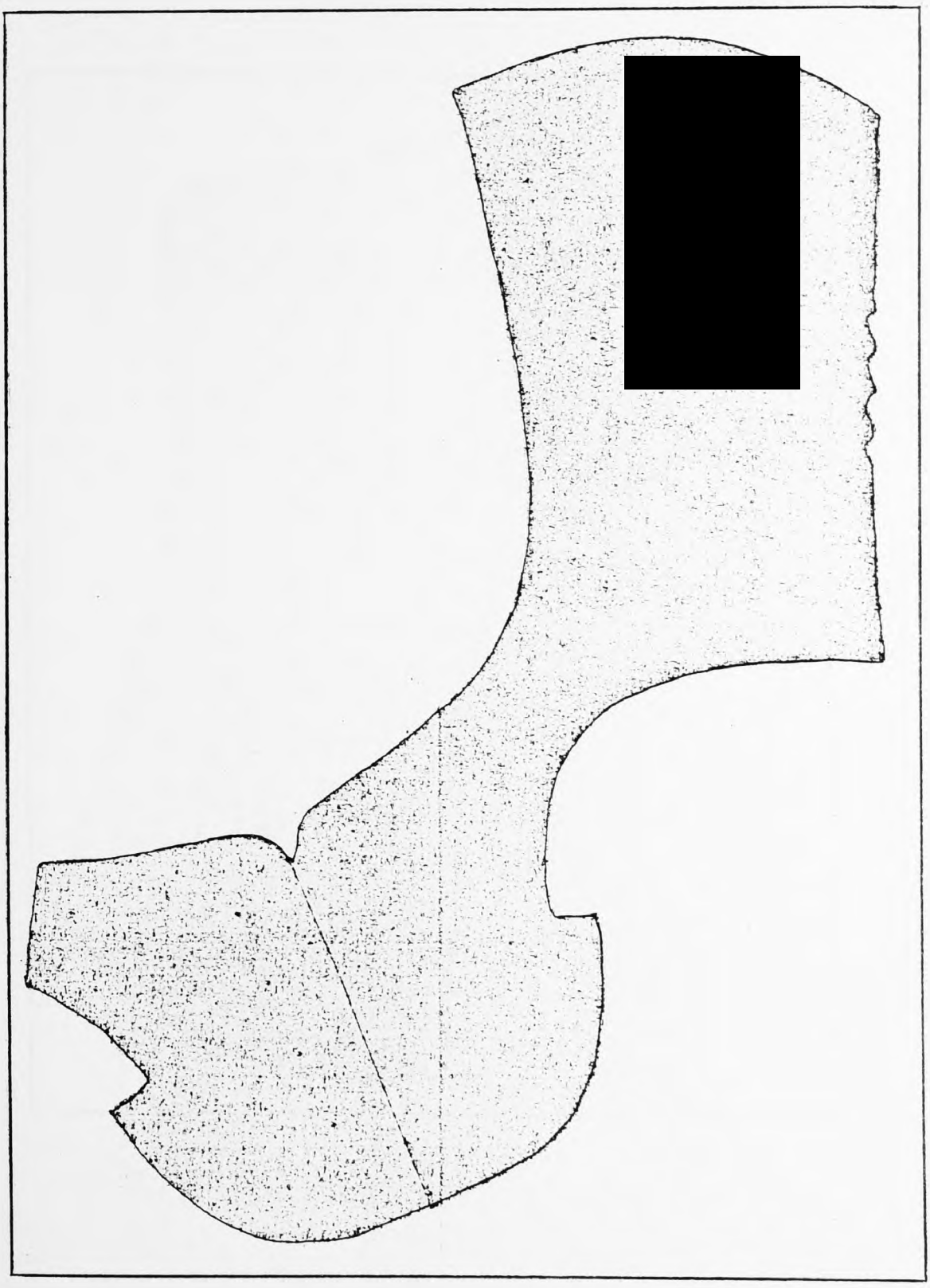


Figure 8.5(c)

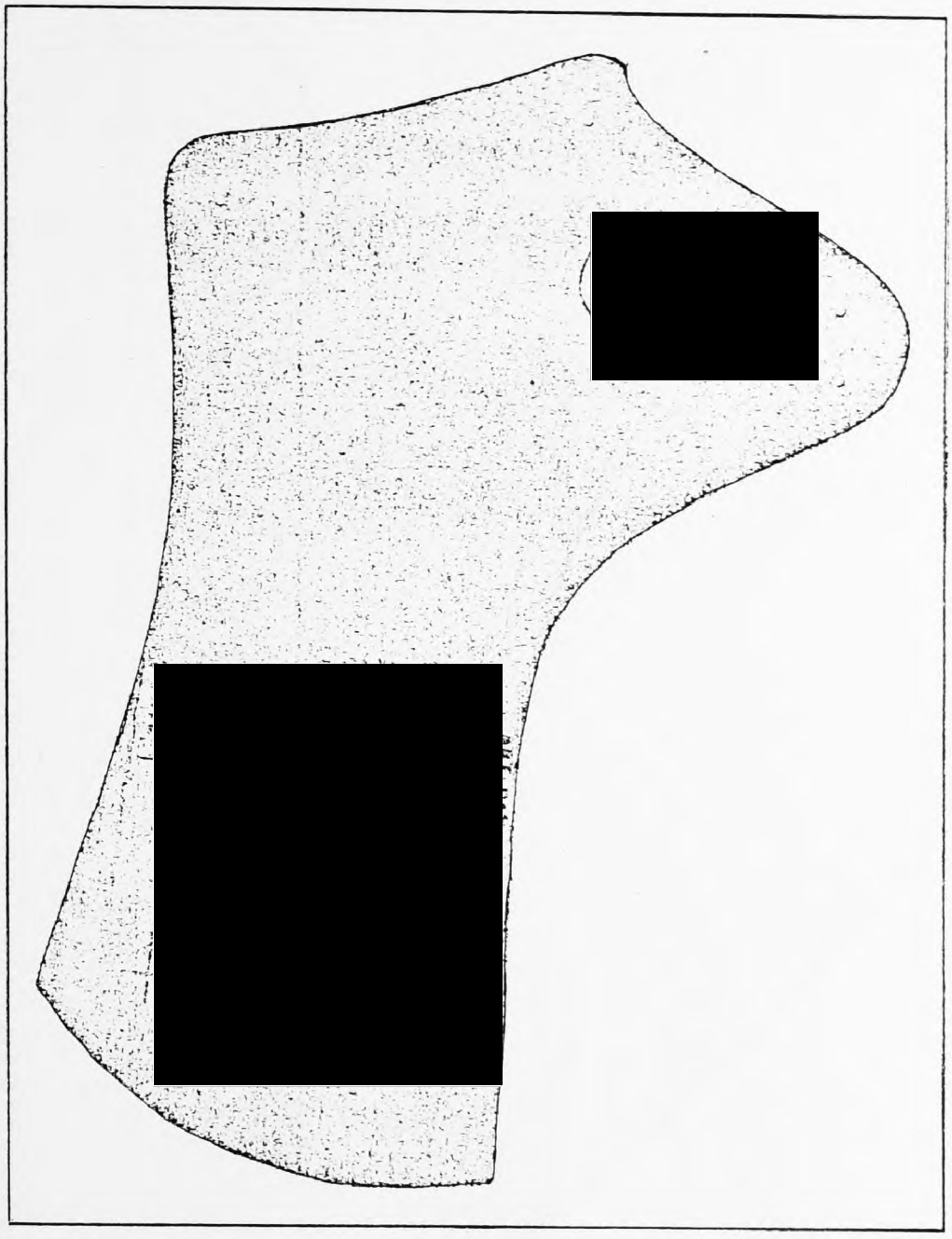


Figure 8.5(d)

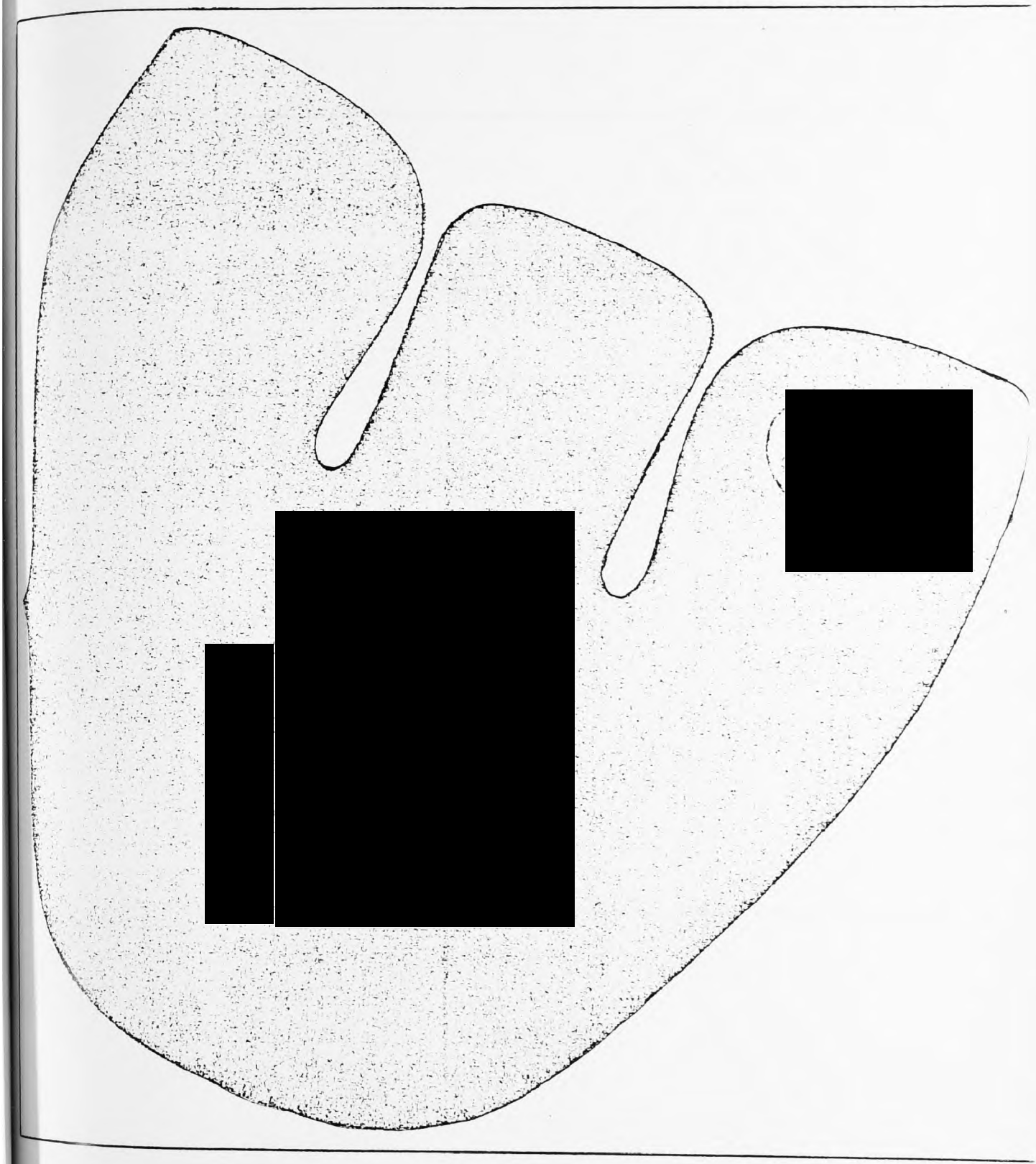


Figure 8.5(e)

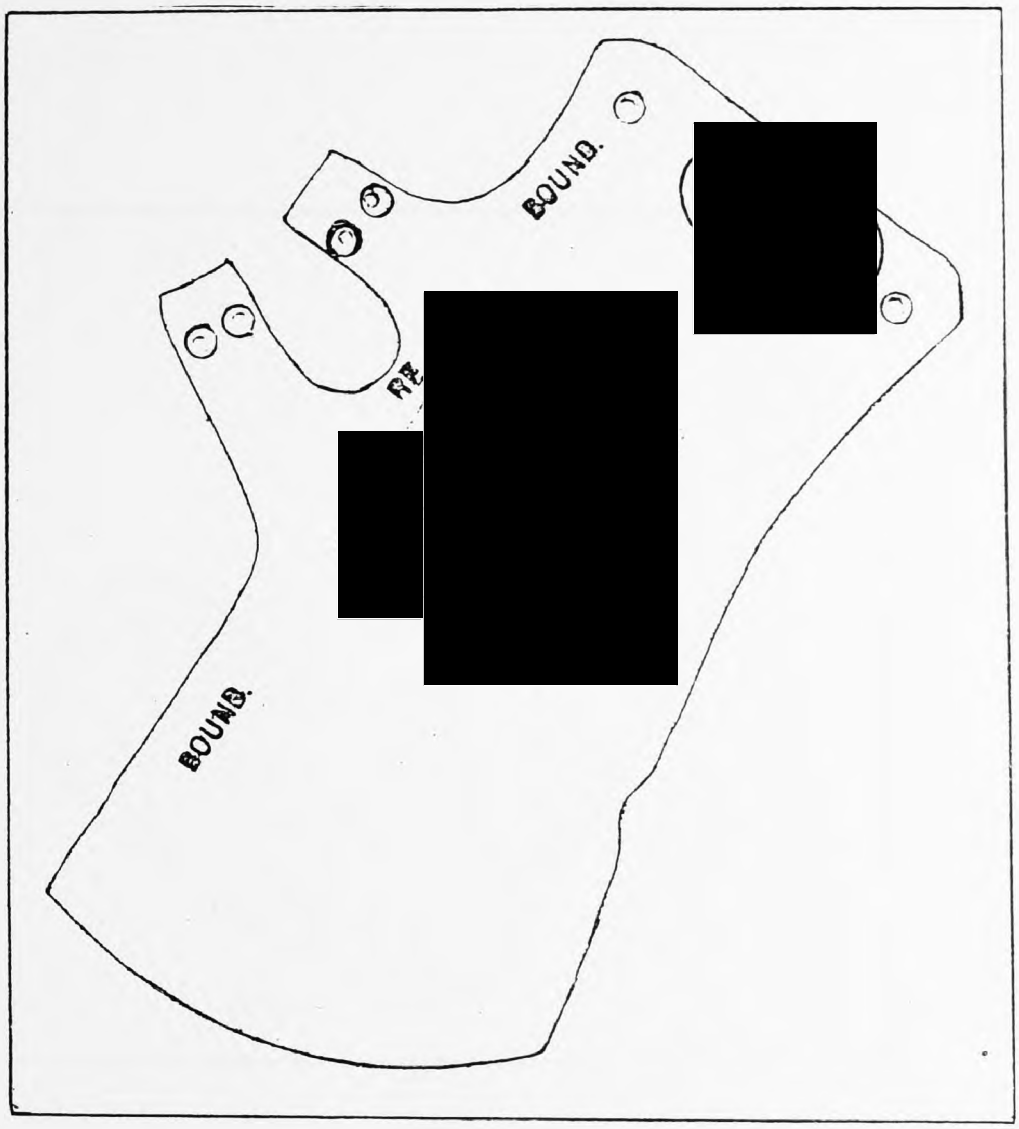


Figure 8.5(f)

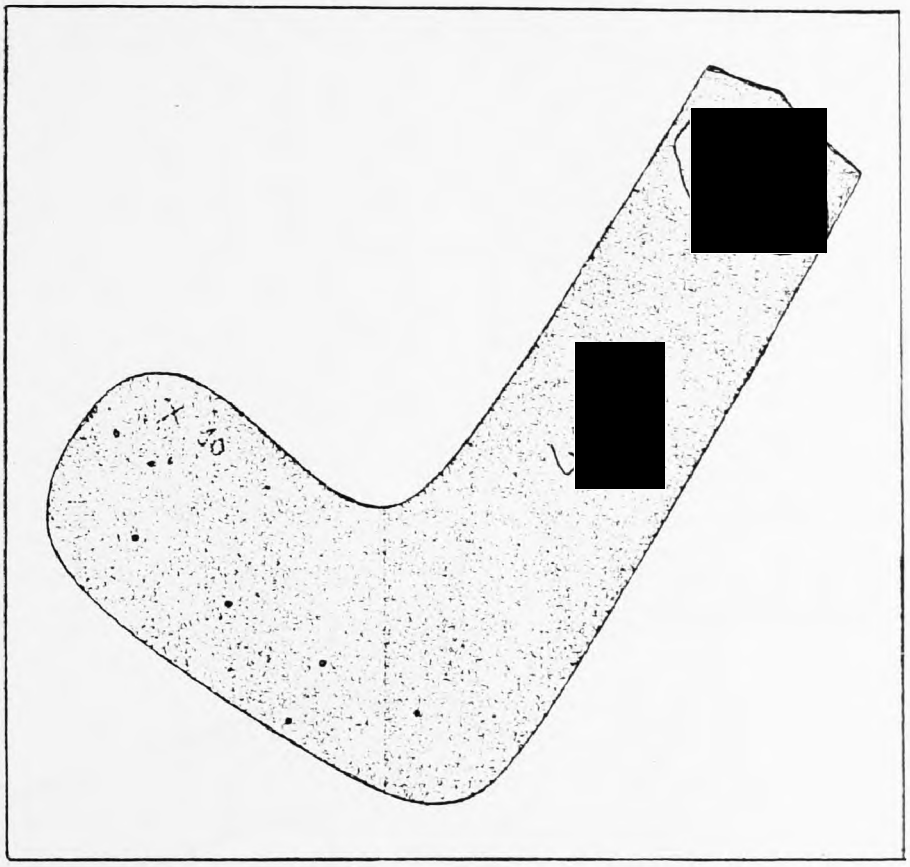


Figure 8.5(g)

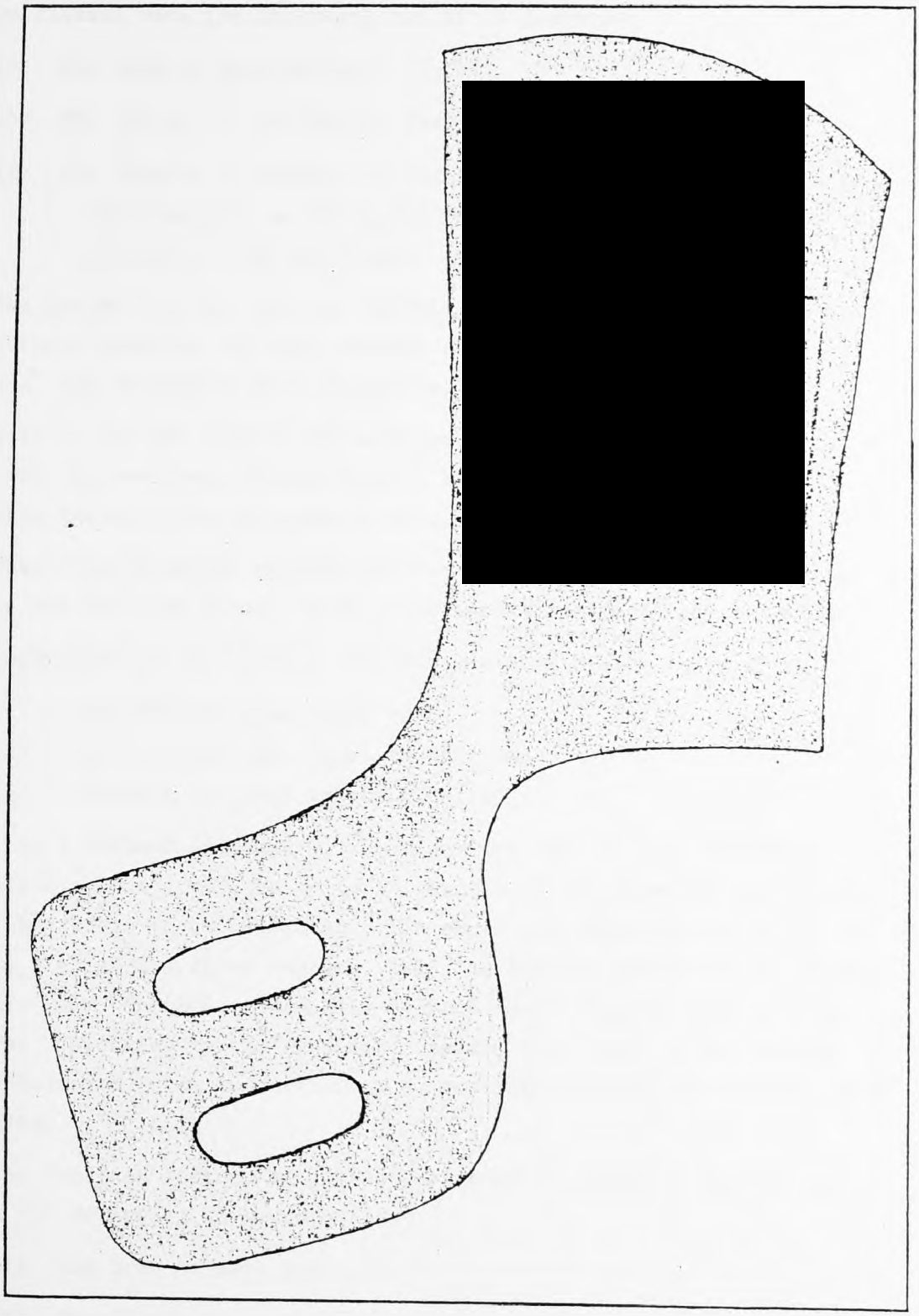


Figure 8.5(h)

The parameters used as the candidate feature set for each of the 100 pattern classes were the following set of 19 features:

(a) The area of each sample,

(b) The length of its longest radius, and

(c) The lengths of another 17 radii measured from the centroid to the boundary, at 20-degree intervals, measured anti-clockwise from the longest radius.

The parameters used for the identification, which, in effect, were the features selected for each pattern class after all classes were "learned" and separated, were determined as follows:

Each of the 100 samples was scanned at random orientation, one at a time, and its candidate feature set of 19 features was extracted. All the above feature sets were stored in a disc backing store.

Then, the selection process was carried out in order to select the feature set for each class, which could separate effectively all classes.

Separation of all classes was achieved effectively by applying:

(1) A minimum distance equal to 6,
(2) Tolerance for area equal to 900, and
(3) Tolerance for each radius equal to 4.

With a minimum distance 6, the algorithm used for the selection evaluation allows errors in up to 5 features of the selected feature set for each pattern to be tolerated. The above selection always starts to contain area as the first feature, then the longest radius as the second feature, and continues to contain as many radii from the rest of the features, belonging to the candidate feature set, until a new feature set is selected which is sufficient to separate all pattern classes from each other.

The selected feature set which separated all pattern classes comprised the following 13 measures:

(i) The area of each pattern,

(ii) The length of its longest radius, and

(iii) The lengths of the next 11 radii measured from centroid to the boundary at 20-degree intervals measured anti-clockwise from the longest radius.

Therefore, the last 6 features from the 19 candidate features for each class were made redundant.

Inspection was performed after the above separation was achieved by scanning each of the 100 shapes, presented at random orientation, and noting the class assigned by the machine for each of them, i.e. being identified. If no class was assigned to any one of the shapes under test, then the message "sample is not identified" was displayed on the display unit of the computer and, hence, a "rejection" was recorded. If, on the other hand, a wrong class was assigned to any of the shapes, then a "substitution" was recorded.

In order to test the efficiency of the system for identification, the above operation was repeated five times in all.

The utility of the figures obtained is reduced somewhat, because sometimes during trials it was necessary to stop the scanning system for short periods (e.g. for other researchers to use the laboratory facilities), or until the following day, thus concealing information concerning errors occurring in the scanning rig due to drifts of temperature, and so on.

However, there is no reason to doubt the measures of performance recorded. The sponsoring company, in fact, provided an assistant, who participated in placing the samples on the scanning table one after the other at random orientation, and witnessed all the tests which were performed.

8.2.1 First experimental evaluation

The first tests for identifying all the 100 samples were performed as described above. The results of those first runs of tests are presented in Table 8.T1.

From this table it can be seen that after the 100 samples were scanned five times, i. e. 500 trials in all, the recognition rate which it yielded was as follows:

(a) 76.2% of the patterns were recognised correctly,

(b) 23.4% of the patterns were rejected, and

(c) 0.4% of the patterns were substituted.

In Table 8.T1 the symbol "-" indicates that the pattern under test was rejected, i.e. it was not identified. Also, in the table the symbol "→" indicates that the pattern under test was substituted with the pattern shown at the tip of the arrow symbol.

Since a much better performance was expected, the above results were considered unsatisfactory. Hence, the causes of poor inspection performance were examined, and ways to eliminate them were considered.

Pattern No.	Class number at 5 different scans				
	1st scan	2nd scan	3rd scan	4th scan	5th scan
1	1	1	1	1	1
2	2	2	2	-	2
3	3	3	3	3	3
4	-	-	4	4	4
5	-	-	5	5	-
6	6	-	6	6	6
7	-	-	-	-	-
8	8	-	8	8	8
9	9	9	9	9	9
10	10	-	10	10	10
11	11	11	11	11	11
12	12	12	12	12	12
13	-	13	13	13	13
14	14	-	14	14	-
15	15	-	15	15	15
16	-	16	16	16	16
17	17	-	17	17	17
18	-	18	18	18	18
19	-	-	19	19	19
20	-	-	20	20	S→10
21	21	-	21	21	21
22	-	22	22	-	S→11
23	23	23	23	23	-
24	24	24	24	24	24
25	25	-	25	25	25
26	26	26	26	26	26
27	-	-	-	-	-
28	-	28	28	-	-
29	-	-	-	-	-
30	30	-	30	30	-
31	-	-	-	-	-
32	-	-	-	32	-
33	33	33	-	33	-
34	-	34	34	34	34
35	35	35	35	35	35

Table 8.T1

Cont...

Pattern No.	Class number at 5 different scans				
	1st scan	2nd scan	3rd scan	4th scan	5th scan
36	-	-	-	-	-
37	37	37	37	37	37
38	38	38	38	38	38
39	39	39	39	39	39
40	40	-	40	40	-
41	-	41	-	41	-
42	42	-	42	42	42
43	-	-	-	-	-
44	-	44	44	44	44
45	45	45	45	45	45
46	46	46	46	46	46
47	-	47	47	47	47
48	48	48	48	48	48
49	49	49	49	49	49
50	50	50	50	-	50
51	51	51	51	51	-
52	52	52	52	52	-
53	53	53	53	53	53
54	54	54	54	54	54
55	55	55	55	55	55
56	56	56	56	56	56
57	57	57	57	57	57
58	58	58	58	-	-
59	59	59	59	59	59
60	60	60	60	60	60
61	61	-	61	61	61
62	-	62	62	62	62
63	63	63	63	63	63
64	64	64	64	64	64
65	-	65	65	-	65
66	66	-	66	-	66
67	67	67	67	67	67
68	68	68	68	-	68
69	69	69	69	69	69

Table 8.T1

Cont...

Pattern No.	Class number at 5 different scans				
	1st scan	2nd scan	3rd scan	4th scan	5th scan
70	70	70	70	70	70
71	71	71	71	71	71
72	72	72	72	72	72
73	73	73	73	73	73
74	74	74	74	74	74
75	-	75	75	75	75
76	-	76	76	-	-
77	-	77	77	-	-
78	-	78	-	-	78
79	79	-	-	79	79
80	80	80	-	-	-
81	-	81	81	-	81
82	82	82	82	-	82
83	83	83	83	83	83
84	84	84	84	84	84
85	85	85	85	85	85
86	86	86	86	86	86
87	87	87	87	87	87
88	88	88	88	88	88
89	89	89	89	89	89
90	90	90	90	90	90
91	91	91	91	91	91
92	-	92	92	-	92
93	-	93	-	93	-
94	94	94	-	94	94
95	-	95	95	-	95
96	-	96	96	-	96
97	97	97	97	97	97
98	98	-	98	98	98
99	-	99	99	99	99
100	100	100	100	100	100
NOT IDENTIFIED	32%	26%	14%	23%	22%
SUBSTITUTION	0%	0%	0%	0%	2%
OVERALL CORRECT IDENTIFICATION: 76.2%					

Table 8.T1

8.2.1.1 Causes of poor inspection performance

Examination in detail of the results taken during the first identification tests shows that mis-identifications occurred preferentially with certain shapes, whilst others were almost invariably recognised correctly.

Five causes were identified for the poor inspection performance of the system, namely:

(a) Shape having two (or more) distinct equal longest radii at no neighbouring points on the boundary, as shown in Figure 8.6 below:

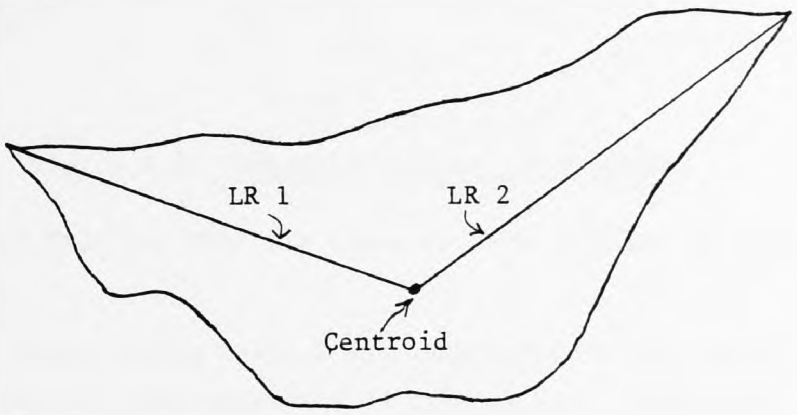


Figure 8.6: Two (or more) longest radii at no neighbouring points

Shapes CK7 and CK43 are shown as such examples in Figures 8.5(e) and 8.5(f).

(b) Shape having uncertain longest radius, owing to the boundary at the longest radius being not a sharp corner but an arc of a circle, the centre of which was the centroid of the shape, which is shown in Figure 8.7:

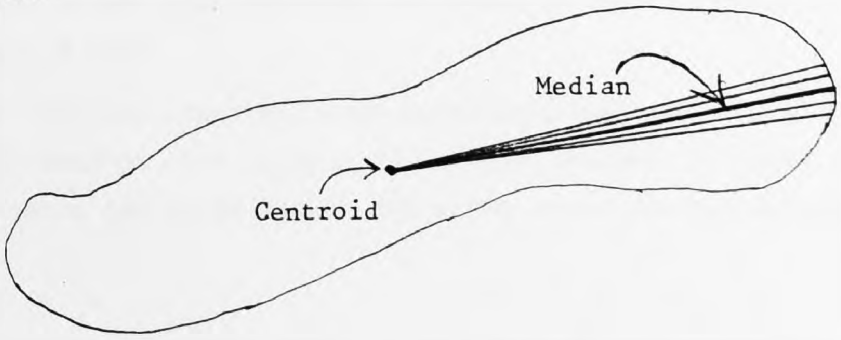


Figure 8.7: More than one longest radii at neighbouring points

Shapes CK41 and CK78 are shown as such examples in Figures 8.5(c) and 8.5(d).

(c) Shape having its centroid outside the boundary, as is shown in Figure 8.8 below:

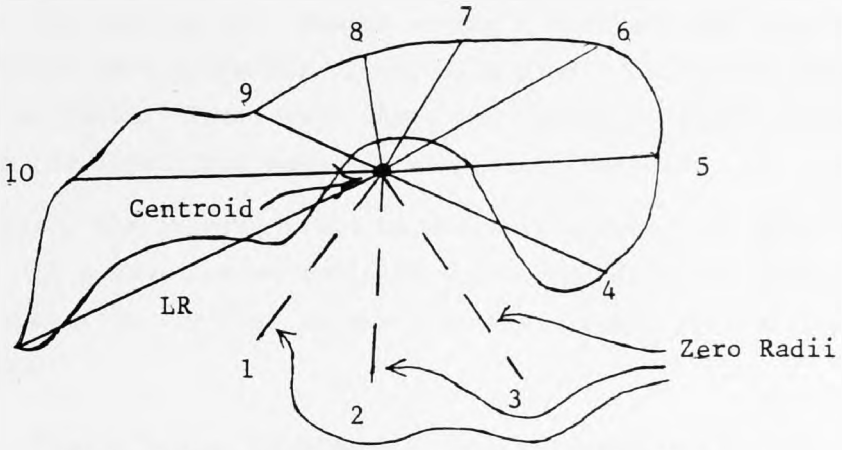


Figure 8.8: Centroid outside the boundary

Shapes CK35 and CK59 are shown as such examples in Figures 8.5(a) and 8.5(b).

(d) Shape having several very short radii, the lengths of which as measured were very sensitive to orientation. Such small radii are shown in Figure 8.9 below:

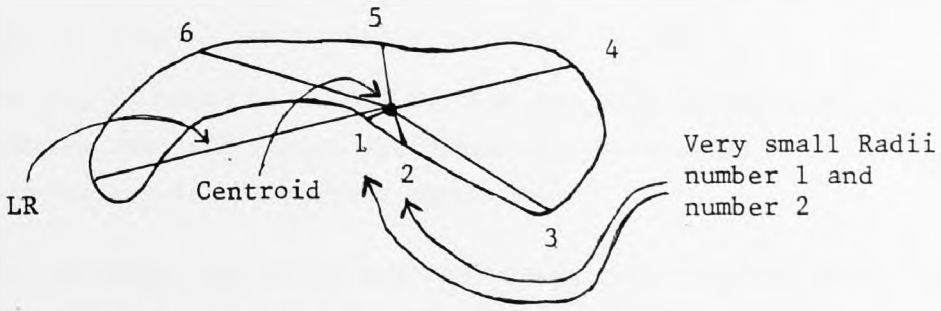


Figure 8.9: Very short in length radii as features

Also, shapes CK20 and CK83 are shown as such examples in Figures 8.5(g) and 8.5(h).

(e) Lastly, error had been introduced because several patterns tended to bend or curl during, or between, periods of their scanning. In some cases the error due to the above cause was indeed considerable.

8.2.1.2 Elimination of causes of bad inspection performance

To eliminate the problems created during the inspection procedure, modifications were made to the identification programs, and a way to protect the samples during scanning was introduced, as follows:

(i) For problem (a), shapes having K distinct and non-contiguous longest radii were scanned K times, each time a different longest radius was used as datum. Thus, each shape was stored as K different classes, but given the same class name for simplicity reasons.

However, the program could be modified further, so that feature sets for all K sub-classes could be determined from one scanning. This problem should in any case decrease as the scanner spatial resolution is improved.

(ii) Shapes having problem (b) were accommodated by taking as longest radius the middle one between several neighbouring ones terminated at points on the boundary being at the same longest distance from the centroid, i.e. the median longest radius was selected. Any identical longest radii falling within an arc 45 degrees in size were dealt with in this way, otherwise the problem was treated also as problem (a).

(iii) For shapes, the centroid of which lay outside the boundary, i.e. problem (c), radii which did not exist at certain angles, because there was no boundary, were simply set equal to zero.

However, it might be useful to have two sets of features, for shapes having centroid inside the boundary, and outside it, although this was not necessary in the present application.

(iv) In every case where any radius was less than 50 cells long, it was set equal to zero, to eliminate problem (d).

(v) To eliminate problem (e), a clean glass sheet was obtained to lay on top of the shapes, and press them flat. Of course, some distortion is introduced because of refraction, as was examined in chapter 6 of this thesis, which distortion has been considered not critical enough for our application.

The effectiveness of the above solutions was established by repeated experimentation on a small number of troublesome shapes.

Further, the filament illumination used was unsatisfactory, since it was not sufficiently bright and individual bulbs kept failing, leading to

varying and poor illumination, which made it difficult to establish a satisfactory threshold for signal binarisation. Hence, the tungsten lights originally used were replaced by more suitable fluorescent illumination, as described also in chapter 6 of this thesis.

8.2.2 Second experimental evaluation

When all the above modifications were incorporated into the inspection system, the previous three-stage procedure was repeated once more, i.e. the scanning and extracting of feature sets of all classes, the separation of the classes, and the scanning and extracting of feature set of each pattern five times for identification purposes.

8.2.2.1 Learning the patterns as classes

During the first stage of the experimental evaluation, scanning and extracting of the candidate feature set for each class (i.e. learning) was performed.

In order to accept effectively all the 100 chosen samples as separate pattern classes for the inspection operation, the following results were recorded:

(a) 100% of the patterns used had been learned once as separate classes.

(b) 24% of the patterns used had to be re-scanned and re-learned once more in new orientations to accommodate ambiguous longest radius problems.

(c) 1% of the patterns used had to be re-scanned and re-learned even twice in new orientation, in order to accommodate extra ambiguity in the location of the longest radius.

Hence, 125 separate classes were, in effect, created out of the 100 samples selected for the present inspection operation.

8.2.2.2 Separation of the classes

During the second stage, i.e. separation of the classes, much valuable information was provided by the evaluation experiments regarding the statistical characteristics of the inspection system.

Thus, a statistical evaluation was performed as follows:

Firstly, the variation in the number of features required in the

selection of the feature set for the separation of all classes in order to maintain a specified minimum distance was determined as a function of the number of patterns to be used as classes.

Secondly, the variation in feature set size with tolerance was determined.

In order to perform the above evaluation, the classes stored in the disc file were grouped in groups of 25, 50, 75, 100 and 125 classes, respectively.

All the recorded results of the evaluation are presented in table form and are shown in Tables 8.T2 to 8.T6.

In these tables the minimum distance is varied from 1 to 8 and sets of three different radius tolerances: 3, 4 and 5, for each of three different area tolerances, 600, 900 or 1200, respectively, are used.

In every case the number of features necessary for separation is recorded, as shown in the above tables.

Also, the graphs which resulted for each of the three different area tolerance conditions are presented in Graphs 8.G1, 8.G2 and 8.G3, respectively.

From these graphs, it can be seen that the feature set size increases very slowly with the number of pattern classes, which suggests that the number of candidate features required to separate a greater number of pattern classes, as required in an industrial application, will not be excessively large.

For the present application, the parameters chosen are minimum distance 6, area tolerance 900 and radius tolerance 4, since they were considered appropriate enough for an acceptable performance of the inspection system used.

Minimum Distance	Area tolerance: 600 Radii tolerances: 3	4	5	Area tolerance: 900 Radii tolerances: 3	4	5	Area tolerance: 1200 Radii tolerances: 3	4	5
1	3	3	3	5	3	3	5	3	3
2	5	7	4	9	7	9	9	7	5
3	9	9	9	10	9	11	10	9	9
4	10	10	11	11	10	19	11	10	11
5	11	11	19	12	11	-	12	11	19
6	12	12	-	14	12	-	14	12	-
7	14	14	-	19	14	-	19	14	-
8	19	19	-	-	19	-	-	19	-

Total number of candidate features per class: 19, and "-" means that no separation is found at the stated conditions.

Table 8.T2: Separation vectors for 25 classes

Minimum Distance	Area tolerance: 600 Radii tolerances: 3	4	5	Area tolerance: 900 Radii tolerances: 3	4	5	Area tolerance: 1200 Radii tolerances: 3	4	5
1	3	3	3	5	3	3	5	3	3
2	5	7	4	9	7	9	9	7	5
3	9	9	9	10	9	11	10	9	9
4	10	10	11	11	10	19	11	10	11
5	11	11	19	12	11	-	12	11	19
6	12	12	-	14	12	-	14	12	-
7	14	14	-	19	14	-	19	14	-
8	19	19	-	-	19	-	-	19	-

Total number of candidate features per class: 19, and "-" means that no separation is found at the stated conditions.

Table 8.T3: Separation vectors for 50 classes

Minimum Distance	Area tolerance: 600 Radii tolerance: 3	4	5	Area tolerance: 900 Radii tolerance: 3	4	5	Area tolerance: 1200 Radii tolerance: 3	4	5
1	3	3	3	5	3	3	5	3	3
2	6	7	5	9	7	9	9	7	5
3	9	9	9	10	9	11	10	9	9
4	10	10	11	11	10	19	11	10	11
5	11	11	19	12	11	-	12	11	19
6	12	12	-	14	12	-	14	12	-
7	14	14	-	19	14	-	19	14	-
8	19	19	-	-	19	-	-	19	-

Total number of candidate features per class: 19, and "-" means that no separation is found at the stated conditions.

Table 8.T4: Separation vectors for 75 classes

Minimum Distance	Area tolerance: 600 Radii tolerances: 3	4	5	Area tolerance: 900 Radii tolerances: 3	4	5	Area tolerance: 1200 Radii tolerances: 3	4	5
1	3	3	3	5	3	3	5	3	3
2	6	7	5	9	7	9	9	7	5
3	9	9	9	10	9	11	10	9	9
4	10	10	11	11	10	19	11	10	11
5	11	11	19	12	11	-	12	11	19
6	12	13	-	14	13	-	14	13	-
7	14	14	-	19	14	-	19	14	-
8	19	19	-	-	19	-	-	19	-

Total number of candidate features per class: 19, and "-" means that no separation is found at the stated conditions.

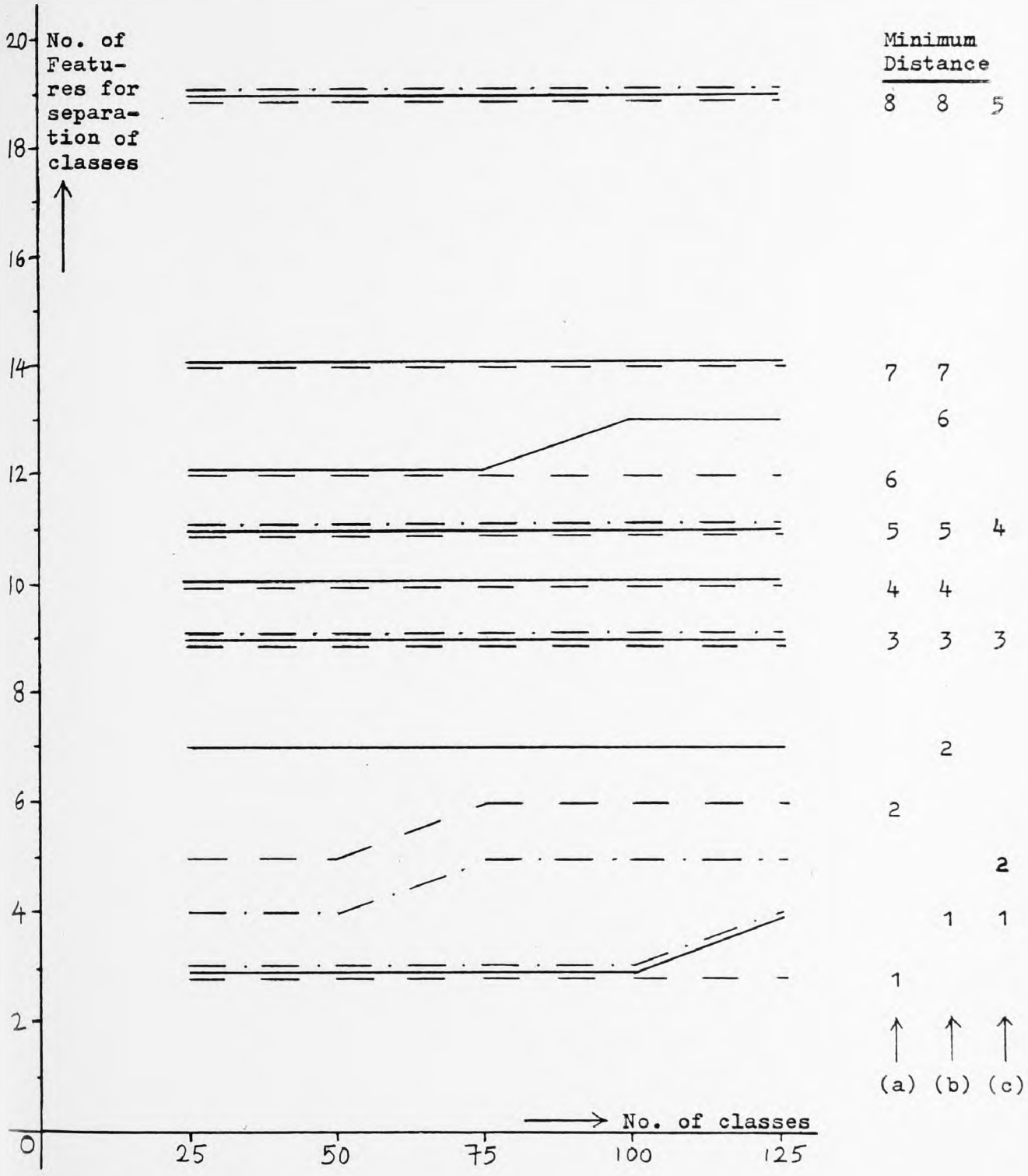
Table 8.T5: Separation vectors for 100 classes

Minimum Distance	Area tolerance: 600 Radii tolerances:			Area tolerance: 900 Radii tolerances:			Area tolerance: 1200 Radii tolerances:		
	3	4	5	3	4	5	3	4	5
1	3	4	4	5	4	4	5	4	4
2	6	7	5	9	7	9	9	7	5
3	9	9	9	10	9	11	10	9	9
4	10	10	11	11	10	19	11	10	11
5	11	11	19	12	11	-	12	11	19
6	12	13	-	14	(13)	-	14	13	-
7	14	14	-	19	14	-	19	14	-
8	19	19	-	-	19	-	-	19	-

Total number of candidate features per class: 19, and "-" means that no separation is found at the stated conditions.

The operating point for the present evaluation is marked by a circle, i.e. where minimum distance is 6, area tolerance is 900 and radius tolerance 4, giving a feature set of 13 features in order to have separation of all the 125 classes used.

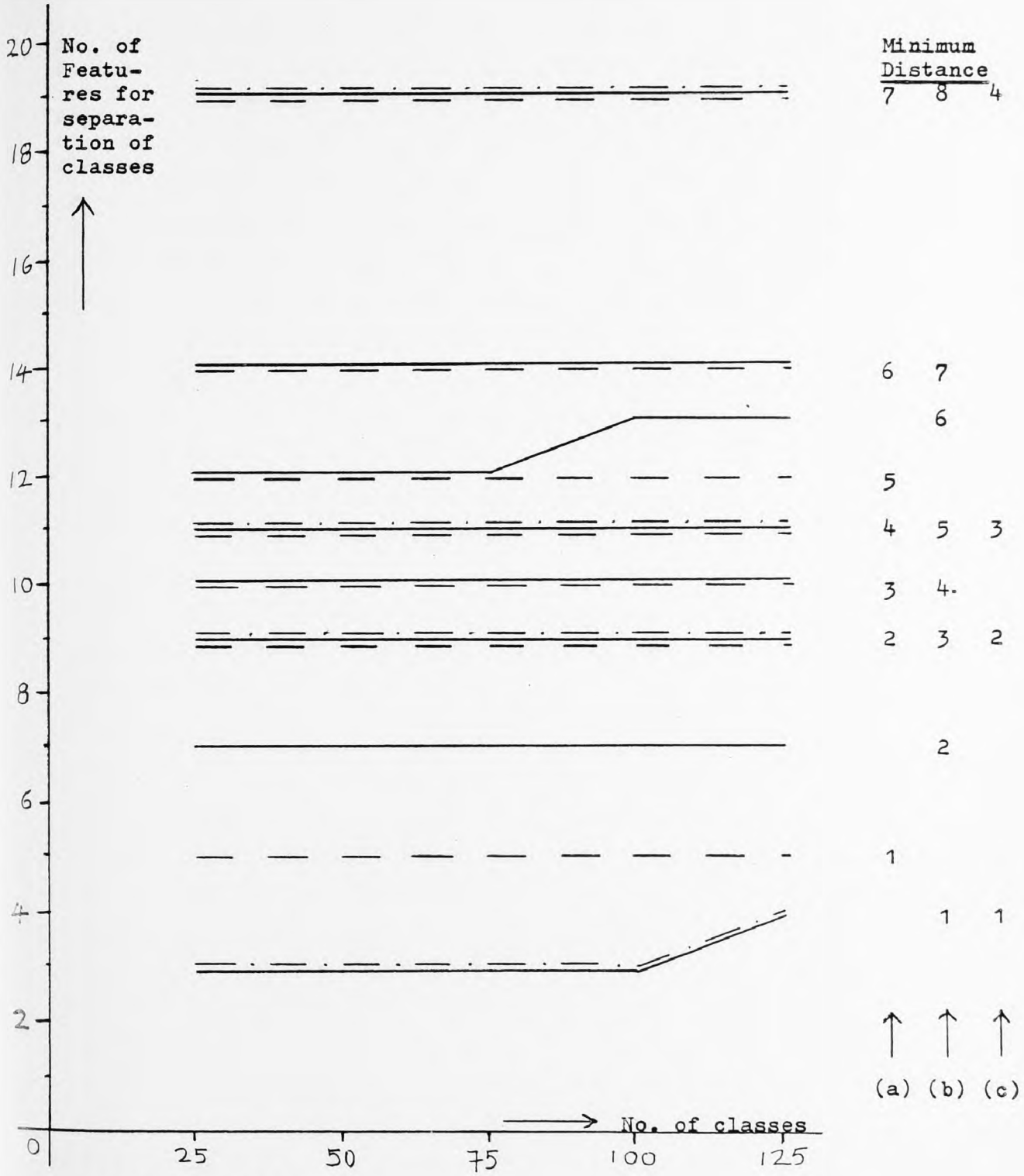
Table 8.T6: Separation vectors for 125 classes



Area Tolerance : 600

Radii Tolerance : (a) – – – 3, (b) ——— 4, (c) – . – . – 5

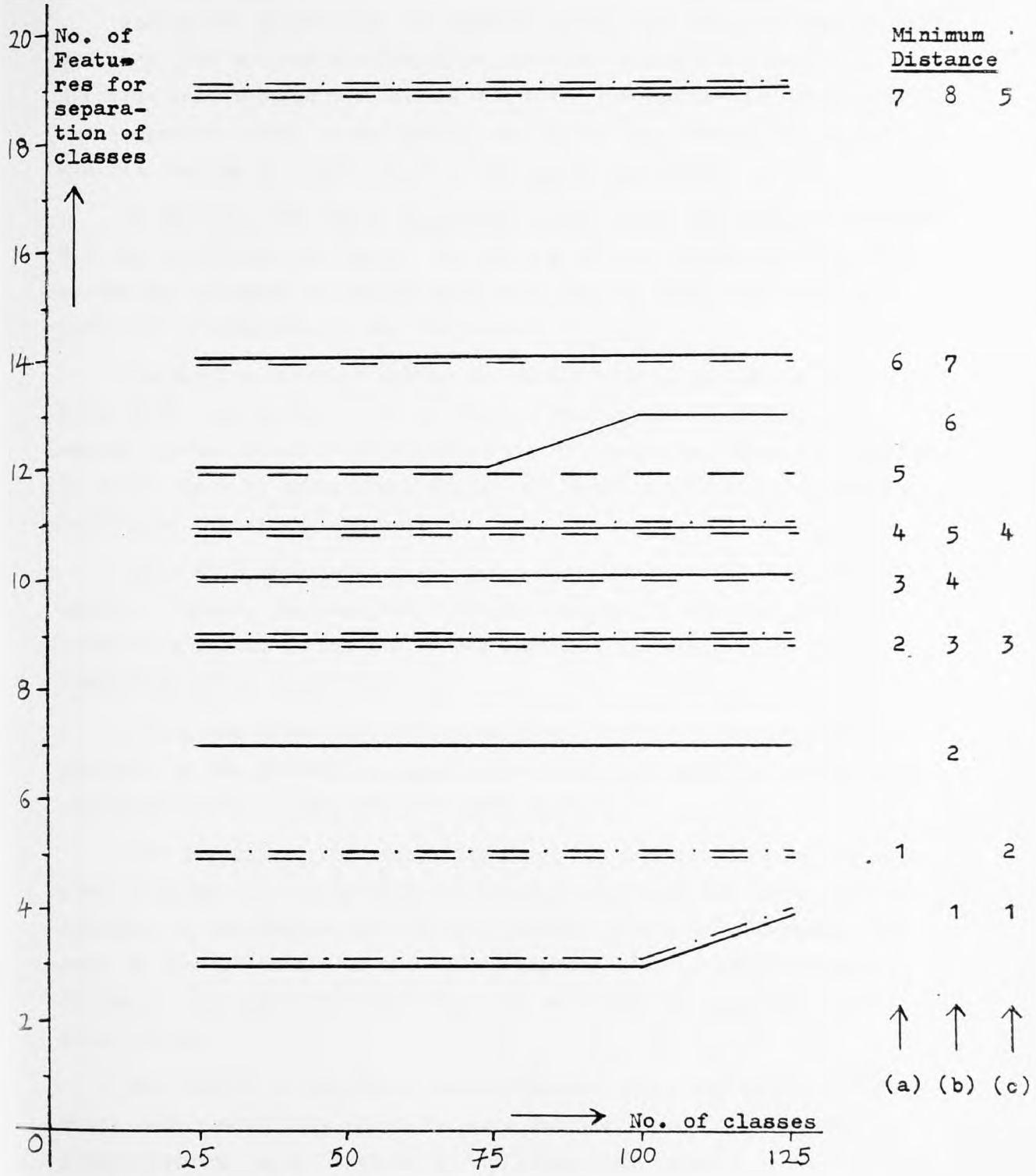
Graph : 8. G1



Area Tolerance : 900

Radii Tolerance : (a) - - - 3, (b) ——— 4, (c) - . - . - 5

Graph : 8. G2



Area Tolerance : 1200

Radii Tolerance : (a) – – – 3, (b) ——— 4, (c) – . – . – 5

Graph : 8. G3

8.2.2.3 Identification tests

During the third stage of identification, all patterns were scanned, one at a time at random orientation, in five consecutive runs, i.e. the patterns were scanned once during the first run before the second run was commenced, then, at the end of the second run, the third run was started, and so on until all five runs were completed.

In this way 500 trials were performed for the 100 patterns selected for the identification tests. At the end of each trial the inspection system was assigned to the pattern under test a class name if it was identified, otherwise it was "rejected".

The results obtained during the above third stage are shown in Table 8.T7. Here, the trial at which a sample was identified as a member of the second or third sub-class of a pattern, which was learned as class twice or three times during the first stage of the evaluation by keeping the same class name, is marked by a star (*) symbol.

Table 8.T7 shows also that the recognition reported was 100% correct. Hence, the identification performance of the modified inspection system during the second experimental evaluation was considered quite acceptable.

During the above identification tests, extra information was provided by the evaluation experiments regarding again the statistical characteristics of the identification system.

Most important, the probabilities of the various features of each pattern under test being measured erroneously, from the corresponding features of the feature set of each pattern class, was determined for each of the 500 trials of the identification tests, where a minimum distance 6 was used and the tolerances were 900 for area and 4 for each radius.

The results of the above evaluation are presented in Table 8.T8, where each probability error for each feature used in the second identification tests is shown at its percentage value.

Pattern No.	Class number at 5 different scannings				
	1st scan	2nd scan	3rd scan	4th scan	5th scan
1	CK 1	CK 1	CK 1	CK 1	CK 1
2	CK 2	CK 2	CK 2	CK 2	CK 2
3	CK 3	CK 3	CK 3	CK 3	CK 3
4	CK 4	CK 4	CK 4	CK 4	CK 4
5	CK 5	CK 5	CK 5	CK 5	CK 5
6	CK 6	CK 6	CK 6	CK 6	CK 6
7	CK 7	CK 7	CK 7	CK 7	CK 7*
8	CK 8	CK 8	CK 8	CK 8	CK 8
9	CK 9	CK 9	CK 9	CK 9	CK 9
10	CK10	CK10	CK10	CK10	CK10
11	CK11	CK11	CK11	CK11	CK11
12	CK12	CK12	CK12	CK12	CK12
13	CK13	CK13	CK13	CK13	CK13
14	CK14	CK14	CK14	CK14	CK14
15	CK15	CK15	CK15	CK15	CK15
16	CK16	CK16	CK16	CK16	CK16
17	CK17	CK17	CK17	CK17	CK17
18	CK18	CK18	CK18	CK18	CK18
19	CK19	CK19	CK19	CK19	CK19
20	CK20	CK20	CK20	CK20	CK20
21	CK21	CK21	CK21	CK21	CK21
22	CK22	CK22	CK22	CK22	CK22
23	CK23	CK23	CK23	CK23	CK23
24	CK24	CK24	CK24	CK24	CK24
25	CK25	CK25	CK25	CK25	CK25
26	CK26	CK26	CK26	CK26	CK26
27	CK27	CK27	CK27	CK27	CK27
28	CK28	CK28	CK28	CK28	CK28
29	CK29	CK29	CK29	CK29	CK29
30	CK30	CK30	CK30	CK30	CK30
31	CK31	CK31	CK31	CK31	CK31
32	CK32	CK32	CK32	CK32	CK32
33	CK33	CK33*	CK33	CK33	CK33
34	CK34	CK34	CK34	CK34	CK34*

Table 8.T7

Cont...

Pattern No.	Class number at 5 different scannings				
	1st scan	2nd scan	3rd scan	4th scan	5th scan
35	CK35	CK35	CK35	CK35	CK35
36	CK36	CK36	CK36	CK36	CK36
37	CK37	CK37	CK37	CK37	CK37
38	CK38	CK38	CK38	CK38	CK38
39	CK39	CK39	CK39	CK39	CK39
40	CK40	CK40	CK40*	CK40	CK40
41	CK41	CK41	CK41	CK41	CK41
42	CK42	CK42	CK42*	CK42	CK42
43	CK43	CK43	CK43*	CK43	CK43
44	CK44*	CK44*	CK44	CK44*	CK44
45	CK45*	CK45	CK45	CK45	CK45*
46	CK46	CK46	CK46	CK46	CK46
47	CK47	CK47	CK47	CK47	CK47
48	CK48	CK48	CK48	CK48	CK48
49	CK49	CK49	CK49	CK49	CK49
50	CK50	CK50	CK50	CK50	CK50
51	CK51	CK51	CK51	CK51	CK51
52	CK52	CK52	CK52	CK52	CK52
53	CK53	CK53	CK53	CK53	CK53
54	CK54	CK54	CK54	CK54	CK54
55	CK55	CK55	CK55	CK55	CK55
56	CK56	CK56	CK56	CK56	CK56
57	CK57	CK57	CK57	CK57	CK57
58	CK58	CK58	CK58*	CK58	CK58
59	CK59	CK59	CK59	CK59	CK59
60	CK60	CK60	CK60	CK60	CK60
61	CK61	CK61	CK61	CK61	CK61
62	CK62	CK62	CK62	CK62	CK62
63	CK63	CK63	CK63	CK63	CK63
64	CK64	CK64	CK64*	CK64	CK64
65	CK65	CK65	CK65	CK65	CK65
66	CK66	CK66	CK66	CK66	CK66
67	CK67	CK67	CK67	CK67	CK67
68	CK68*	CK68	CK68	CK68	CK68
69	CK69*	CK69	CK69	CK69	CK69

Table 8.T7

Cont...

Pattern No.	Class number at 5 different scannings				
	1st scan	2nd scan	3rd scan	4th scan	5th scan
70	CK70	CK70	CK70*	CK70	CK70
71	CK71*	CK71	CK71	CK71	CK71
72	CK72	CK72*	CK72	CK72	CK72
73	CK73	CK73	CK73	CK73	CK73
74	CK74	CK74	CK74	CK74	CK74
75	CK75	CK75	CK75	CK75	CK75
76	CK76	CK76	CK76	CK76	CK76
77	CK77	CK77	CK77	CK77	CK77
78	CK78	CK78	CK78*	CK78	CK78
79	CK79	CK79*	CK79	CK79	CK79
80	CK80	CK80	CK80	CK80	CK80*
81	CK81	CK81	CK81	CK81	CK81
82	CK82	CK82	CK82	CK82	CK82
83	CK83	CK83	CK83	CK83	CK83
84	CK84	CK84	CK84	CK84	CK84
85	CK85	CK85	CK85	CK85	CK85
86	CK86	CK86	CK86	CK86	CK86
87	CK87	CK87	CK87	CK87	CK87
88	CK88	CK88	CK88	CK88*	CK88
89	CK89	CK89	CK89	CK89*	CK89
90	CK90	CK90	CK90	CK90	CK90
91	CK91	CK91	CK91	CK91	CK91
92	CK92	CK92	CK92	CK92	CK92
93	CK93	CK93	CK93	CK93	CK93
94	CK94	CK94	CK94	CK94	CK94
95	CK95	CK95	CK95	CK95	CK95
96	CK96	CK96	CK96	CK96	CK96
97	CK97	CK97	CK97	CK97	CK97
98	CK98	CK98	CK98	CK98	CK98
99	CK99	CK99	CK99	CK99	CK99
100	CK100	CK100	CK100	CK100	CK100

Table 8.T7

Feature vector	Percentage probability of error
Area	19.8
Longest radius	12.4
2nd radius	31.4
3rd radius	8.8
4th radius	13.6
5th radius	21.0
6th radius	14.0
7th radius	25.4
8th radius	27.2
9th radius	18.2
10th radius	13.8
11th radius	25.4
12th radius	20.4

Table 8.T8: Probability of measuring features erroneously

The average percentage probability of error per feature, taken over all 13 features (N) of the feature vector is, therefore, 19.3% (= p(e)).

The percentage probabilities of the features being in error (i.e. out of tolerance) simultaneously during the 500 scanning trials, involving the 100 patterns, were also recorded and are shown in Table 8.T9 below:

Number of features out of tolerance simultaneously	Percentage of tests
0	8.4
1	17.4
2	25.6
3	20.4
4	19.4
5	8.8
6 (or more)	0.0 ↘ (no rejection)

Table 8.T9: Percentage of tests for which stated number of features were out of tolerance simultaneously

The information shown in Tables 8.T8 and 8.T9 is essential in the provision of a prediction and optimisation to the identification performance of the present system. This is presented in the following paragraph.

8.3 Performance Prediction and Optimisation

The performance of the inspection system with given measurement resolution depends on the minimum distance (D), the number of features in the feature set for each pattern under test (N), and the average probability of each feature being measured erroneously (p(e)). This last quantity is reasonably conjectured to be inversely proportional to cell size.

The number of features needed to provide a specified minimum distance depends on the number and variety of the shapes, and also on the method used for selecting the feature subset for each shape (N). An optimal combination is possible, given parameters obtainable by measurement, but may be computationally not feasible.

The analysis and prediction of performance, and its possible optimisation, also form the subject of the present chapter.

8.3.1 Statistical analysis for performance prediction

The identification process may be described statistically as follows:

N measurements are made of features, in sequence. The measures are quantised into tolerance cells, by dividing them by a number representing the size of a tolerance cell. The quantised measures are then compared with stored measures, similarly quantised, representing pattern classes.

Suppose that a vector of quantised measures is compared with the vector of stored measures representing its true class. Each measure in the vector may be either within tolerance, in which case it provides a "fit", or outside tolerance, in which case it is in error.

Identification comprises comparing the measures in the feature to be identified, one by one, with corresponding measures for learnt features typifying stored classes, until a prototype is found from which the vector being identified differs here in (D - 1) measures or fewer.

D is the "minimum distance" for the feature set, the significance of which was explained earlier in this thesis.

Rejection occurs when D or more measures in the vector being identified fall outside their specified tolerance cell, due to errors of various kinds which are considered random and mutually independent.

Thus, the identification is effectively a sequence of N experiments having a binomial outcome.

The probability of K or more features falling outside their respective tolerance cells is given by the cumulative binomial distribution below, in which p(e) is the average probability of measurement error taken over all features of the selected feature set (N):

$$P(K) = \sum_{r=K}^{N} N!/r!(N - r)!\left[p(e)\right]^{r}\left[1 - p(e)\right]^{N-r} \qquad (8.E1)$$

Further, the probability of 1, 2, 3, ..., K features exactly being in error, is given by the individual terms in the binomial distribution:

$$p(r) = N!/r!(N - r)!\left[p(e)\right]^{r}\left[1 - p(e)^{N-r}\right] \qquad (8.E2)$$

During the second identification tests, which were described in the previous paragraph, the probabilities of 1, 2, 3, ..., etc. features being in error simultaneously during the 500 trials of the 100 patterns, are shown in Table 8.T9.

These results are plotted in Figure 8.10 below, in a histogram form. In this figure, the dotted bars on the histogram superimposed on the measured values, which are drawn as solid lines, show the probability of the specified number of errors occurring simultaneously as predicted by the binomial distribution, given the measured percentage probability of error per feature, p(e).

It is seen that each measured bar height is closer to its predicted value by less than the expected error of measurement (the heights of the measured bars obey a Poisson distribution), thus confirming that the binomial model for the identification process is valid.

However, the 19.3% result measured from Table 8.T8 for the average percentage probability of error per feature, taken over all 13 features of the feature vector, suggests a probability of rejection of 2%, as is

also shown in the histogram of Figure 8.10 for the case of more than five features in error.

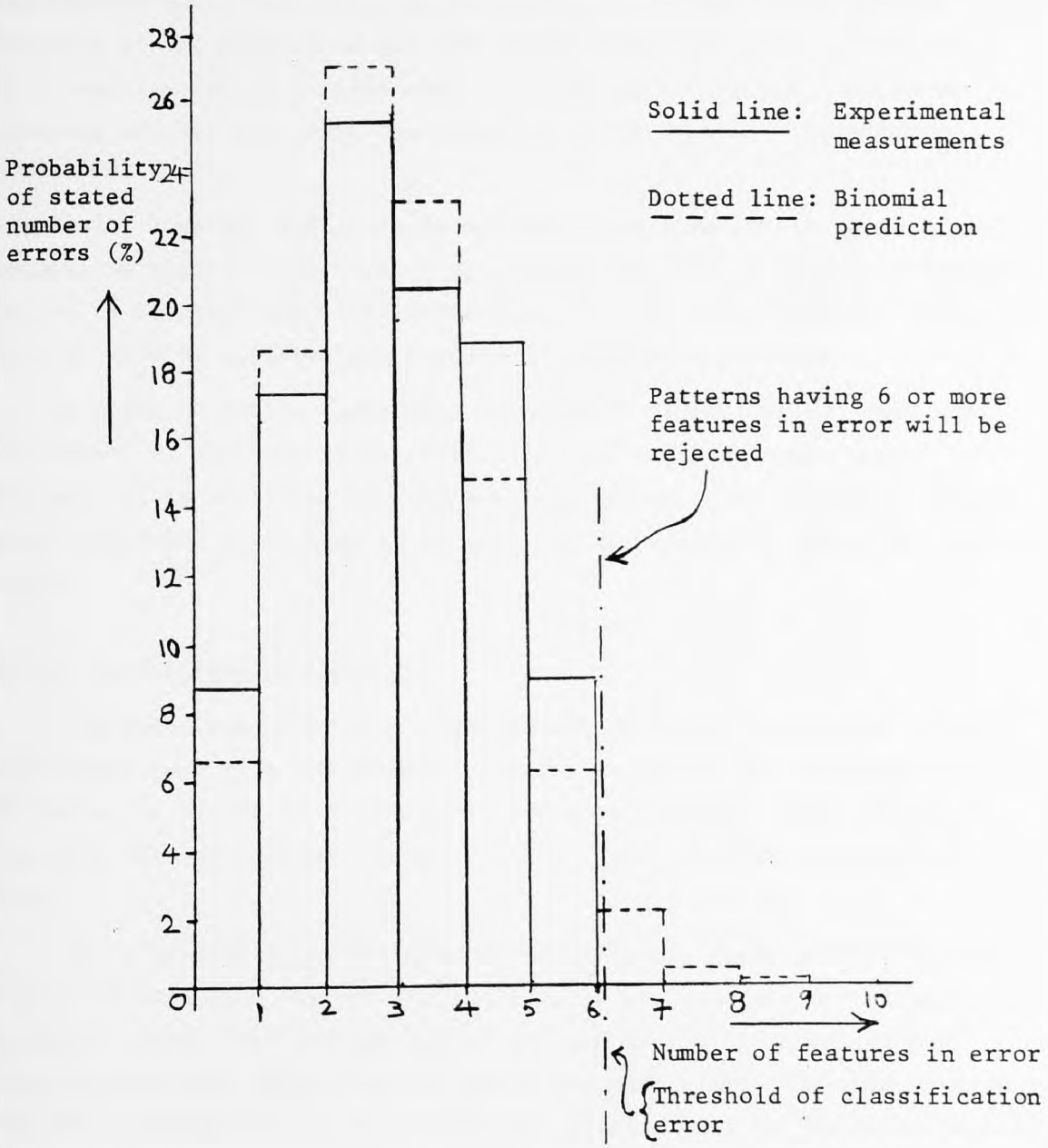


Figure 8.10: Error histogram for features measured and predicted

This is found from equation (8.E2), by substituting r by 6, N by 13 and p(e) by 0.193, respectively, giving indeed a 2% result, or 10 rejections out of the 500 trials.

However, no rejections were recorded during the second identification tests.

The above discrepancy is due mainly to temperature drift of the

scanning system, particularly at the time the system is restarted after an interval of being switched off. At such scanning time, features of the pattern under test could be influenced differently from the way features of the pattern of the same class could have been influenced if it was scanned as a class previously during a long and continuous scanning period, i.e. with the scanning system at stable temperature conditions.

It is, however, difficult to achieve every time stable temperature conditions before tests have to be carried out, when a long experimental period is involved and other researchers require the laboratory facilities at several intermediate periods for different purposes.

In order to avoid, therefore, unnecessary rejections, or even identification of patterns which could have been rejected under normal conditions, it is desirable for the scanning system to be at stable temperature conditions every time it is used for an industrial inspection application.

8.3.2 Performance optimisation

The selection of a "best" feature set involves a selection of the parameters cell size and minimum distance to obtain the required performance, or better if possible, in terms of rejection rate, whilst at the same time minimising storage, look-up time and feature selection time.

It is generally not possible to minimise all these simultaneously. Since the prototype vectors for the various patterns are distributed sparsely within the "feature space" defined by the features, it has been assumed that rejection will occur far more often than substitution. The above assumption has been confirmed particularly by the first experimental results.

The procedure for selecting a feature set from the candidates measured during the learning stage is sub-optimal, in that more features than are really necessary have been included. This can be seen clearly from the example shown in Figure 7.3(d) of chapter 7 of this thesis, where the second feature of each of the four pattern classes used (i.e. 15, 15, 15 and 14) does not contribute to the improvement of the inspection performance and it may be discarded without any real loss. The algorithm used is, however, attractive in being fast and not very difficult to program

and mainly fast to run. It is important, though, to know the extent of sub-optimality of the feature sets used.

Also, the analysis using the cumulative binomial distribution shows that for typical parameter values, reducing the average probability of error per feature by a factor 2, e.g. from about 20% (since 19.3% is measured) to 10%, reduces the average rejection rate from about 2% to 0.05%, i.e. by a factor of 40.

Hence, the problem of optimal design should have as its objective the minimisation of the rejection rate with specified performance requirements.

8.3.2.1 Optimal feature set

As has already been seen, the first sub-optimal feature selector builds up the set by starting with area and longest radius, then adding further radii to the set, in anti-clockwise order, until the required minimum distance is achieved. Radii which contribute no improvement in performance during the above operation are not discarded.

However, if a subset providing a specified minimum distance actually exists within a set of candidate features, it is certain to be found. Almost invariably, it will contain more features than are necessary.

In general, there will be several distinct combinations containing N', say, features which are optimal. A stronger optimality can then be obtained by taking into consideration the probabilities of erroneous measurement for the various features, where this is known.

Thus, one takes as optimal the set containing N' features for which the sum S(P), defined below, is minimum:

$$S(P) = \sum_{P} p(k)$$

where p(k) is the probability of error for the k^{th} feature, and the summation includes only the features within the set selected.

The only obvious way to select an optimal feature set is to examine all combinations of the candidate features singly, in pairs, in triples, and so on, until a combination having the required minimum distance is obtained. This process is very expensive computationally. Some economy is obtainable by not examining subsets containing fewer features than the minimum distance.

If the candidate set contains M features, then there will be M individual features, M.(M - 1) pairs, M.(M - 1).(M - 2) triples, and so on. The number of combinations which have to be searched for M = 19 is given in Table 8.T10 below:

Number of features in subset	Number of searches (cumulative)
1	19
2	361
3	6,175
4	99,199
5	1,494,559
6	21,029,599
7	274,985,119
8	3,322,451,359

Table 8.T10: Number of combinations to be searched for optimal feature selection with M = 19

To investigate the extent to which the current feature selection is sub-optimal, and to determine more precisely the cost in additional computing time if optimal selection is used, an evaluation was carried out to select features optimally, by using the data acquired during the inspection tests for the 125 separate pattern classes.

It has been assumed here, as was also mentioned at the beginning of paragraph 8.3, that the average error p(e) in matching features between sample and prototype is inversely proportional to tolerance cell size.

The results obtained from the above evaluation are shown in Table 8.T11 below for p(e) = 20% (which is very close to the one measured, i.e. 19.3%, with area tolerance 900 and radius tolerance 4), with equivalent results obtained using the sub-optimal selection shown side by side for comparison. Thus, two extra sets of results have been included, firstly by using p(e) = 10% when the area cell size is doubled to 1800 and the radius cell size is doubled to 8, and, secondly, by using p(e) = 40% when area cell size is halved to 450 and radius cell size is halved to 2, respectively.

Also, in Table 8.T11 are included the computed optimal sets of features for minimum distance 1 and 2, respectively, since there was time

to select them optimally only up to minimum distance 2. This is because the optimal selection is expensive in computer time. On the PRIME 550 computer used for optimal selection, runs per program of greater than 3 hours are prohibited by the operating system.

Minimum Distance	Number of features for stated misfit probability			
	p(e) = 10%	p(e) = 20%	p(e) = 40%	Optimal (with p(e) = 20%)
1	9	4	3	2
2	11	7	4	4
3	14	9	7	-
4	15	10	9	-
5	19	11	10	-
6	-	13	11	-
7	-	14	12	-
8	-	19	14	-
9	-	-	15	-
10	-	-	19	-

Table 8.T11: Size of optimal and sub-optimal feature sets

It is seen from the above table that by selecting the set having the minimum distance 2 optimally, it contained only 4 measures, whereas the sub-optimal set with p(e) = 20% contained 7, also with minimum distance 2. For minimum distance 1, the optimal set contained 2 features, whilst the sub-optimal set with p(e) = 40% included 3 with minimum distance 1.

The consequences of this in terms of classifier performance are discussed in the following section, 8.3.2.2.

8.3.2.2 Optimal design for minimisation of rejection rate

The starting point for optimal design for minimisation of rejection rate is the specification of the performance requirements.

Given that the largest acceptable probability of rejection per pattern is p(1), and that L patterns must be stored as classes, experiments give the number of features required per feature set for a specified minimum distance D for the following two categories:

(a) Using the fast feature selection method adopted for the present work, and

(b) Using the optimal, but slow, feature selection process described in the previous section, 8.3.2.1.

Increasing the size of the tolerance cells increases, on average, the number of features required to maintain a specified minimum distance, but reduces the probability of features being measured erroneously. This can be seen more clearly in the graphs of Figure 8.11 below, which is obtained from the results presented in Table 8.T11.

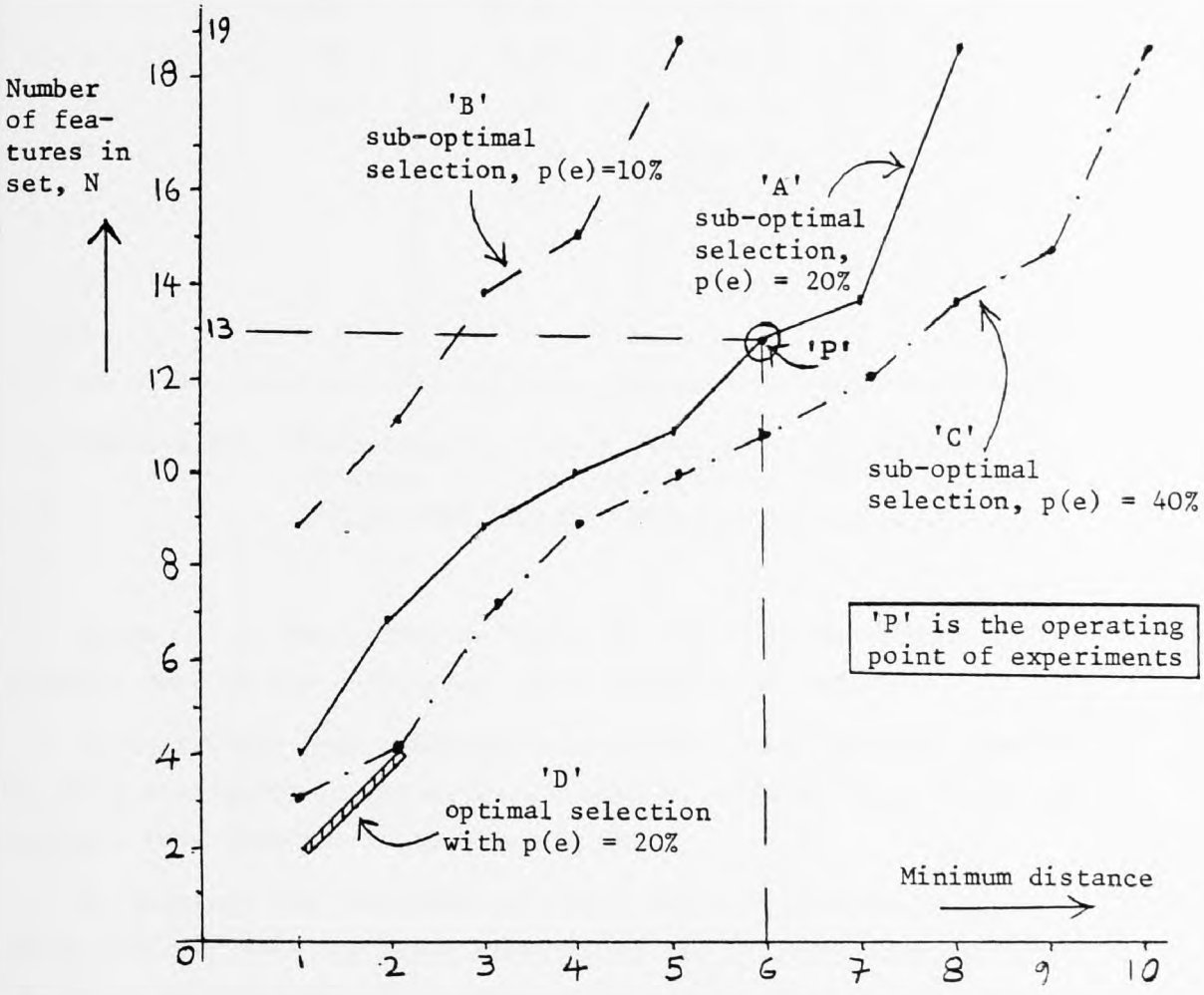


Figure 8.11: Number of features in feature set vs. minimum distance, with average probability error per feature as parameter

Thus, the use of large tolerance would seem beneficial, although the larger feature set size would increase the storage requirement and look-up time by about 17%.

The problem of optimal design is summarised by the graphs in Figure 8.12, in which the various curves represent design alternatives, with the objective of minimising the rejection rate, p(1), which is computed by using the equation (8.E2) and is presented in Table 8.T12 below for each

individual case, which was previously considered.

Minimum distance	Reject rate (%) with the following p(e) as parameter:			
	p(e) = 10%	p(e) = 20%	p(e) = 40%	Optimal (with p(e) = 20%)
1	38.74	40.96	43.20	32.00
2	21.31	27.52	34.56	15.36
3	11.42	17.62	29.03	-
4	4.28	8.81	25.08	-
5	2.66	3.87	20.06	-
6	-	2.30	14.71	-
7	-	0.99	10.09	-
8	-	1.66	9.18	-
9	-	-	6.12	-
10	-	-	9.76	-

Table 8.T12: Percentage of reject rate, p(1), vs. minimum distance with p(e) as parameter for sub-optimal and optimal feature sets

Curve 'A' of the graphs in Figure 8.12 is for the feature set actually used in the evaluation, with operation at point 'P'.

It is evident that performance could have been improved somewhat by using a slightly larger minimum distance, although this would have retained more features in the feature set.

By doubling the tolerance cell size for each feature, e.g. by using 1800 for area and 8 for each radius length, operation on curve 'B' could be obtained. The consequence, again, would be improved performance, but the candidate feature set used did not include sufficient features to achieve a minimum distance greater than 5.

Halving the tolerance cell size for each feature would have produced operation in curve 'C', which is clearly far inferior to that of curves 'A' and 'B'.

The graphs of Figure 8.11, where A, B, C, D and P have the same significance as in Figure 8.12, show how the feature set size varies with minimum distance, with tolerance cell size (i.e. probability of mis-match) as parameter. The storage required to hold a given number of

shapes, and the average time required to identify a shape, both increase linearly with feature set size.

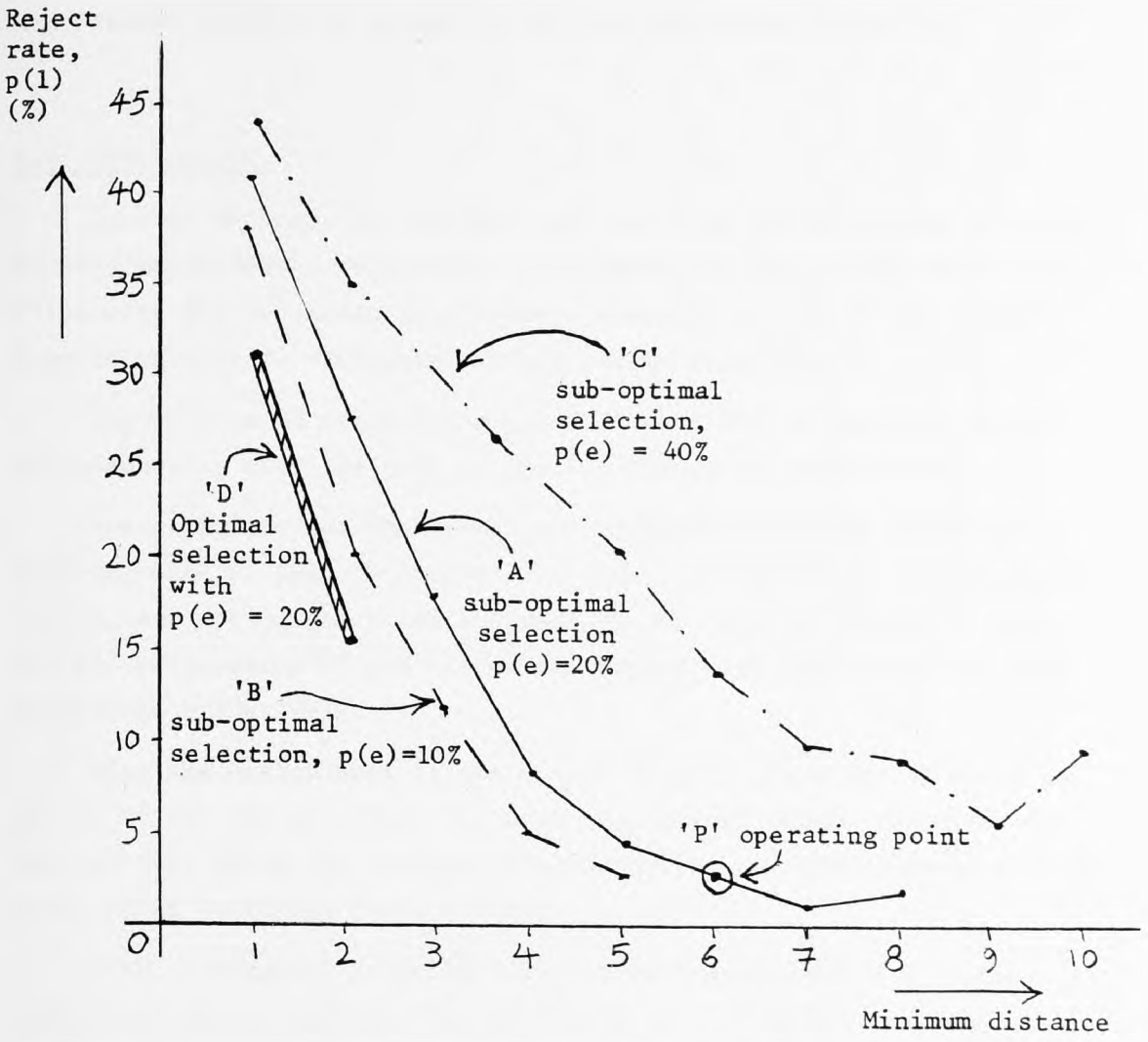


Figure 8.12: Percentage reject rate, p(1), plotted against minimum distance with p(e) as parameter for sub-optimal and optimal feature sets

The best performance of all is obtained by operating on curve 'D', which has the same tolerance cell size as for curve 'A', but with features selected optimally. This has the consequence that the number of features required to achieve a given minimum distance is almost halved. This minimises the data storage requirement and look-up time.

However, situation 'B', in which improved identification performance is obtained using larger tolerance cells, imposes a penalty, requiring more store and a longer look-up time.

The optimal feature selection of 'D' is, on the other hand, expensive in computer time. This has led to the decision of adopting the situation 'A', and in particular the operating point 'P', as the best solution to the present problem of automatic on-line inspection operation.

8.4 Conclusions

In this chapter, the primary task has been the provision of a convincing experimental evaluation, by examining a set of 100 different silhouette shapes chosen to provide a reasonable test of the present inspection system. This was carried out successfully.

The problem of orientation has been examined further and solved satisfactorily with the help of special simulation experiments.

The efficiency of the system for identification was tested by scanning each of the 100 shapes five times, presented at random orientation, and noting the class assigned by the machine. Several causes for bad performance of the inspection system were identified and then eliminated effectively.

When the performance of the system reached the state at which it fulfilled all the necessary requirements, a statistical analysis was carried out, using the results obtained during the experimental evaluation, which confirmed their validity.

Also, a possible solution for performance optimisation has been considered which, however, was proved to be too expensive computationally with the computer unit used for the on-line operation.

CHAPTER 9: OVERALL CONCLUSIONS AND RECOMMENDATIONS

9.0 INTRODUCTION

In the present thesis, all the stages of progress and attainments are described for a research project aimed at providing an instrument which may be trained to identify automatically silhouette shapes.

The above project was successful insofar as the methodology developed and the design details, which are presented in this thesis, represent a physically realised automatic on-line inspection system that works fast and performs the functions for which it was originally intended. Further, it provides an important basis for future expansion of its present capabilities.

It is also fair to point out the important contribution which this work has made to the solution of the wider problem of silhouette image processing, by providing a completely new approach and a new design, which actually produced the desired results as they were expected at the time the present project was initiated.

9.1 Methodology

The methodology selected and developed during the present work, which is based on feature extraction from silhouette shapes imposed on a binary grid of cells, has been shown to work successfully.

Since orientation as well as high accuracy were important requirements in the development of this work, and since other methods available, such as Fourier or density function methods, could not adequately fulfil all the above requirements, it was decided that the present, i.e. area and radius from the centroid, approach should bring the best possible solution.

Indeed, this has been achieved, something which shows distinctly the contribution of the method developed here to the solution of the above requirements and, to a considerable degree, of the orientation problem.

The biggest problem, however, which was phased from the very beginning, was the amount of computation involved, and in particular during the initial measurements for extracting the boundary points from a

silhouette shape under test.

It was decided to solve the above problem by designing and constructing special-purpose hardware to be incorporated in the scanning system developed for the present work. This proved to be a wise decision since it improved the performance in two ways - very high speed and higher accuracy - something which undoubtedly gave an additional and very important capability to the whole system as originally intended.

The automatic identification process which was used involved recognition of each shape under test from a selection of several shapes representing pattern classes, which were mutually separated by selecting appropriately a feature set, with the help of the "minimum distance" concept, which proved very valuable and provided a powerful immunity to error in parameter measurements.

Since some of the above ways and techniques were original, patent cover has been applied for.

9.2 The Practical Area of the Automatic On-line Inspection

A one-dimensional photodiode array camera, of the 1728 CCD type, was used as the image-sensing device.

Television cameras were considered to be unacceptable for the present work because of their low performance, and laser scanners too expensive.

For flat, opaque objects which were used a precision of $\pm$ 0.05 mm was attainable with the present scanning system. There is no theoretical limit to the number of measurements which can be taken with the resolution obtainable.

Some errors could be eliminated by increasing the resolution (e.g. tolerance), others by improving the scanning conditions where it was necessary, due to mechanical imperfections of the scanning table drive, and to the separation between camera and moving table not being always exact.

The experimental evaluation which was carried out proved that the method adopted here was feasible and acceptable for industrial automatic inspection applications.

In particular, awkward shapes could be accommodated without intervention by the operator of the on-line operational system, and

extrapolation from data obtained experimentally reasonably predicted that the quantity of data which would have to be stored for large numbers of pattern classes would not be excessively large.

A statistical analysis of the performance and its prediction with the system used indicated the validity of the results obtained in the above evaluation.

Although an optimal performance design was also produced for the present inspection system, it was considered to be not implementable because of the computation involved with the small computer used in the laboratory where the on-line operation was performed.

9.3 Cost Considerations

During the development and practical application of the present work the cost factor was treated with extra care.

Here, cost was considered in connection with the quality of the work produced during each stage of the present research project. By "quality" we mean fitness for the purpose intended.

The cost of quality was connected mainly with the following three components:

(a) Preventive costs, associated with designing for ease of assembly and reducing the number of parts before constructing.

(b) Costs of appraisal, e.g. inspection and testing to be performed fast and accurately enough so as to avoid unnecessary repetitions, and

(c) Costs of failures, i.e. the consequential cost arising from failure of the preventive and appraisal activities was kept as low as possible.

Although it was not possible to calculate precisely the above costs involved during the present work, an estimate is given below of the relative proportions of the costs similar to the ones mentioned above, which occur generally in industrial environments today:

(i) Prevention costs: 10%

(ii) Appraisal costs: 25%

(iii) Failure costs: 65%

The cost of failures and their rectifications is the largest fraction and is attributable to deficiencies in the quality assurance activities.

9.4 Recommendations for Future Work

Although the system works as required, it would be useful to conduct some experiments to assess further the dynamic performance of the driving system and to investigate the effects of varying degree of mechanical damping upon the mechanical resonance, which could become critical if very long periods of use of the present scanning system became necessary in future.

It is probably worth devoting some further study in the task of feature selection, since there may be sub-optimal methods which provide feature sets which are almost optimal, but which can also be located reasonably quickly.

The optimal method which was examined here requires a faster general-purpose computer for the on-line operation, or a special-purpose architecture, which might well render optimal selection practicable.

The company sponsoring this present work has already started to build its own scanning system in its plant, which will initially be almost identical to that built at the laboratory where this research project was developed.

The new scanning rig should, however, have a faster and more robust mechanical drive, and a higher resolution. Thus, the new system should then be used to confirm predictions of performance by operating over even larger data bases.

9.5 Summary

The geometric inspection procedure allows items to be inspected and their orientation and position determined.

An operational automatic on-line inspection system would have a repertoire of basic property measuring algorithms, which would obtain the nominal object property values in a "learn" phase and compare them with all subsequent presentations in an "inspect" phase. Several different objects could be learnt so that mixed items could be inspected and identified simultaneously.

This technology is no longer too expensive, and application is as much limited by imagination as by any other factor.

APPENDIX

A.0 INTRODUCTION

In this appendix the complete design description is given of the special-purpose hardware interface system for preprocessing.

Also, the multiplexing system, which was designed especially so that the fast preprocessor could operate completely independently of the original slow-speed preprocessor of the scanning rig, is included.

The present design is a proposed functional block of an experimental system for obtaining digital descriptions of the visual aspects of physical objects.

The function of this system is to identify the position and nature of the edge of objects, i.e. to obtain a silhouette description, and to transmit this information to a computer for further processing as required.

The circuit is designed to work with the Fairchild SL62912 "CCD 121 HC design development board", from which it receives video signals for preprocessing.

The developed system is designed to communicate with a D.E.C. PDP 11/10 minicomputer through a "DR11-B" direct memory access interface, through which control signals are received and through which control signals and data are transmitted to the minicomputer.

A.1 System Configuration

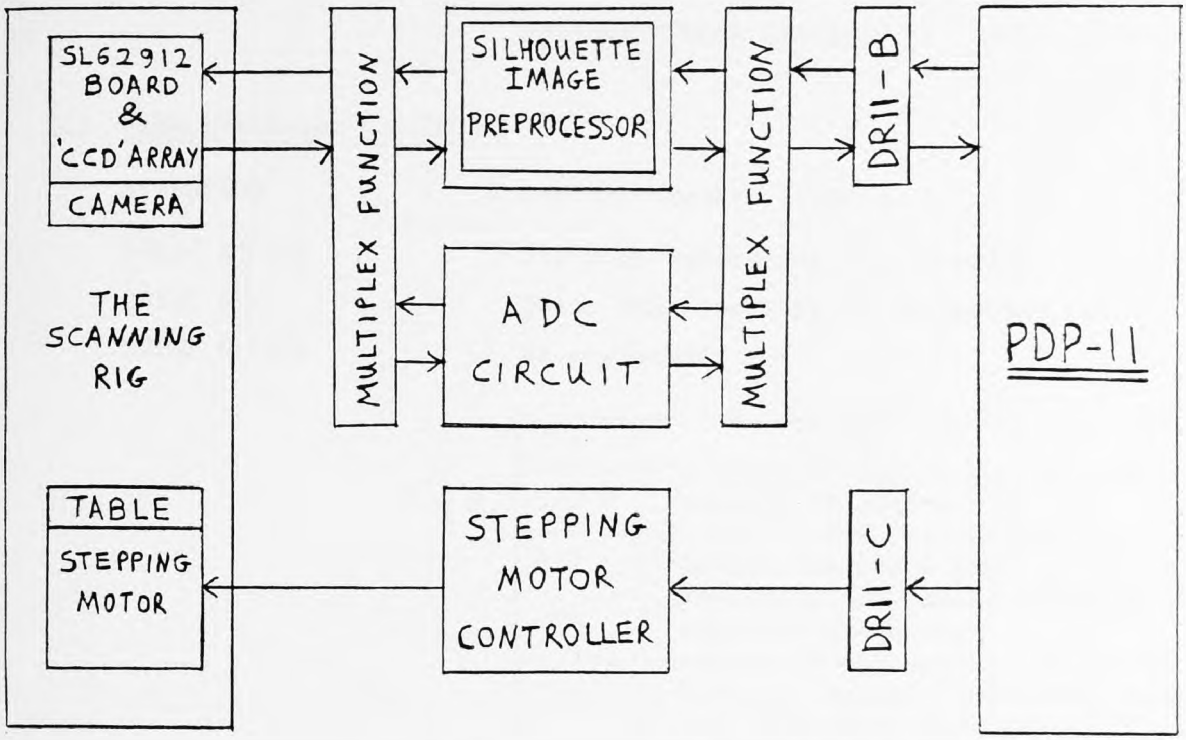


Figure A.1: Experimental scanning system

The silhouette image preprocessor can be selected to operate in the system through the multiplexing function. When this is the case the following signals apply: (see Block diagram: silhouette image preprocessor)

(a) From processor to SL62912 board

CCDCK	Clock signal driving SL62912

(b) From SL62912 to processor

ΦXB	Scanning line sync signal
VIDEO	Analog line scan video signal

(c) From PDP11 to processor

FNCT 1	Used as computer interface control enable and interrupt acknowledge.
FNCT 2	Used as the master enable signal for the processor.

END CYCLE — Signal occurring when a D.M.A. cycle is complete, i.e. when one 16 bit word has been transferred to computer memory. Follows "CYCLE REQUEST".

(d) From processor to PDP 11

ATTENTION — Interrupt request signal.

DSTAT A (LSB), DSTAT B, DSTAT C (MSB) — 3 bit code indicating the function which the interrupt is requesting to be performed:

C B A	
0 0 0	Request to step motor
0 0 1	Informs computer that a scan has just been processed, in which no edges were detected
0 1 1	Informs computer that some data have just been sent to it through D.M.A.
All others	Invalid - fault

SINGLE CYCLE — This signal prevents the computer from using the multibus while a D.M.A. transfer is in progress.

CYCLE REQUEST — This signal requests the DR11-B to transfer a word of data to the computer memory through a D.M.A. operation. Answered by "END CYCLE" on completion of transfer.

C1 = 1), C0 = 0) Hardwired — 2 bit D.M.A. control word. In this case specifying that data should be transferred from the user device to the computer.

BA INC ENB = 1} Hardwired — Enables the DR11-B to increment the bus address afrer each D.M.A. cycle.

WC INC ENB = 0} Hardwired — Disable D.M.A. transfer word count. This helps to ensure that errors do not arise because of word count overflow.

A00 = 0}Hardwired — Bus address L.S.B. tied low because word transfer only is used in D.M.A. and words are stored at even addresses.

A.2 External Operation of Processor

A.2.1 Processing initiation

Before commencing operation, the computer must initialise the processor and DR11-B interface. It does this by:

(a) Setting signals FNCT 1 and FNCT 2 low, which disables and initialises the processor.

(b) Loading the DRBA (bus address) register, in the DR11-B interface, with the start address of an area in computer memory, which will be used as the buffer stack for D.M.A. data transfers.

When the PDP 11 is ready to proceed, it sets both FUNCT 1 and FNCT 2 high, which sets the processor going.

During operation the processor communicates with the PDP 11 in order to request various actions and to transmit processed data. These operations are detailed in the following paragraphs.

A.2.2 Request to operate stepping motor

At the beginning of each processing cycle (i.e. at the beginning of each scan) the processor issues an interrupt request with DSTAT = 0. This indicates that the SL62912 has acquired a set of data in the CCD shift register, from the photodiode array, at the current position of the table, and is transmitting it to the processor. The computer can now step the table to the next position in preparation for the next scan.

The computer clears the interrupt request by pulsing FNCT 1 low and operates the stepping motor, as appropriate, via a DR11-C interface and the stepping motor controller (see Figure A.1).

A.2.3 Transmission of processed data

If the processor detects edges during a scan, the associated data (i.e. x co-ordinate and nature of edge) have subsequently to be sent to the computer.

When transmission is to take place, after processing, the processor sets signal "SINGLE CYCLE" low, which forces the DR11-B to keep control of the computer bus while data transfer takes place. The processor then pulses signal "CYCLE REQUEST" to request that a data word be transferred via the D.M.A. When a word has been sent the DR11-B pulses signal "END CYCLE" which primes the processor to set up another word for transmission. When all data have been sent, the final "END CYCLE" causes "SINGLE CYCLE" to be set high again, which terminates the transmission.

Shortly following the end of transmission the processor issues an interrupt request (details in following paragraphs).

A.2.4 End of scan status report

When the processor has completed processing the video data for any scan, it will have reached one of two conclusions:

(a) There were no edges in that scan, or

(b) There were edges in that scan.

In either case the processor issues an interrupt request. In case (a) DSTAT = 1. In case (b) DSTAT = 3. The computer then takes one of the following courses of action:

(a) If DSTAT = 1 and the computer has not, since enabling the circuit, been informed that edges have been detected, it means that it is still the start of the frame because the table has not yet been stepped up to the object. In this case the computer clears the interrupt request by pulsing FNCT 1 low and takes no further action.

(b) If DSTAT = 3, it means that edges were detected in the current scan and the data pertaining to those edges have been added to the data buffer stack in the computer memory, via a D.M.A. In this case the computer clears the interrupt request by pulsing FNCT 1 low, notes the fact that edges have been found and takes appropriate action with the new data.

(c) If DSTAT = 1 and the computer has previously noted that edges were found, it means the whole object has been traversed by the stepping process and a complete set of data is now present in the buffer stack. In this case, since no further operation is required of the processing circuit, the computer sets both FNCT 1 and FNCT 2 low, thus disabling the circuit.

A.3 Principle of Operation

There are two processing functions within the silhouette image pre-processor. These are:

(a) Conversion of the analog video signal into a binary signal producing a silhouette representation.

(b) Identification of the position and nature of the edges of the object under inspection, from the binary signals.

A.3.1 Derivation of binary signal

The binary signal is derived very simply (in principle) by comparing the video signal voltage with a predetermined reference voltage.

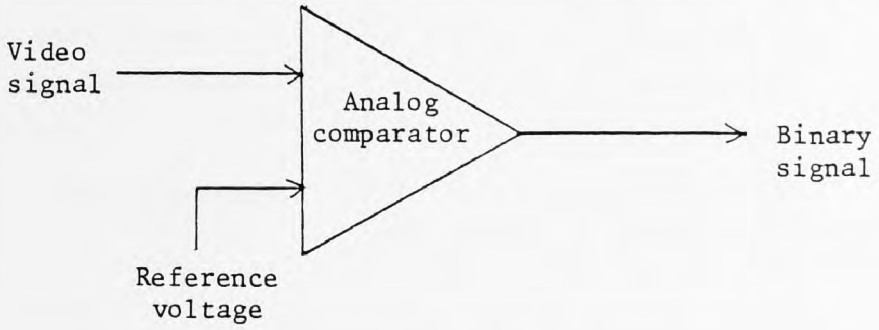
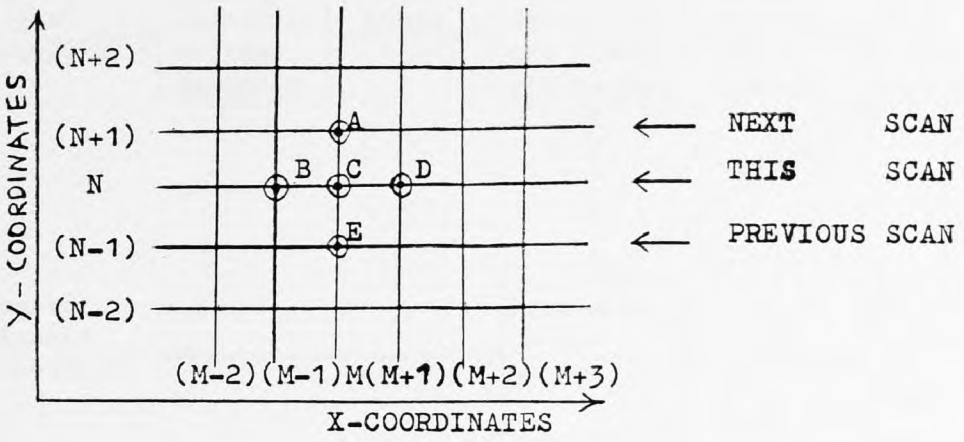


Figure A.2: Derivation of binary signal

A.3.2: Identification of edges

The method used to specify the edge points of the object is the three scan system (see chapter 4), which is summarised briefly in Figure A.3.

In Figure A.3 the distance between x co-ordinate points corresponds to the resolution distance between adjacent photodiodes in the array. The distance between y co-ordinate points corresponds to the distance which the table is stepped by the computer, between scans.



Point C at co-ordinate (M, N) is the point under consideration.

"X" = don't care condition.

A B C D E	Nature of point C
0 1 1 1 1	Up point
1 1 1 1 0	Down point
X 0 1 1 X	Left point
X 1 1 0 x	Right point
All others	Not a valid edge

Figure A.3: Summary of three scan method

A.4 Circuit Description

The silhouette image preprocessor consists of five principal functional areas, which are:

(1) The analog threshold voltage detector, i.e. the circuit which derives the binary signal from the video signal.

(2) The edge detection and classification circuit.

(3) The processed data buffer storage function. The data relating to any edges detected in a scan (i.e. x co-ordinate and nature of edge) are assembled here during processing, prior to transmission to the computer.

(4) The processor control function, including clock generator.

(5) The computer interface control logic.

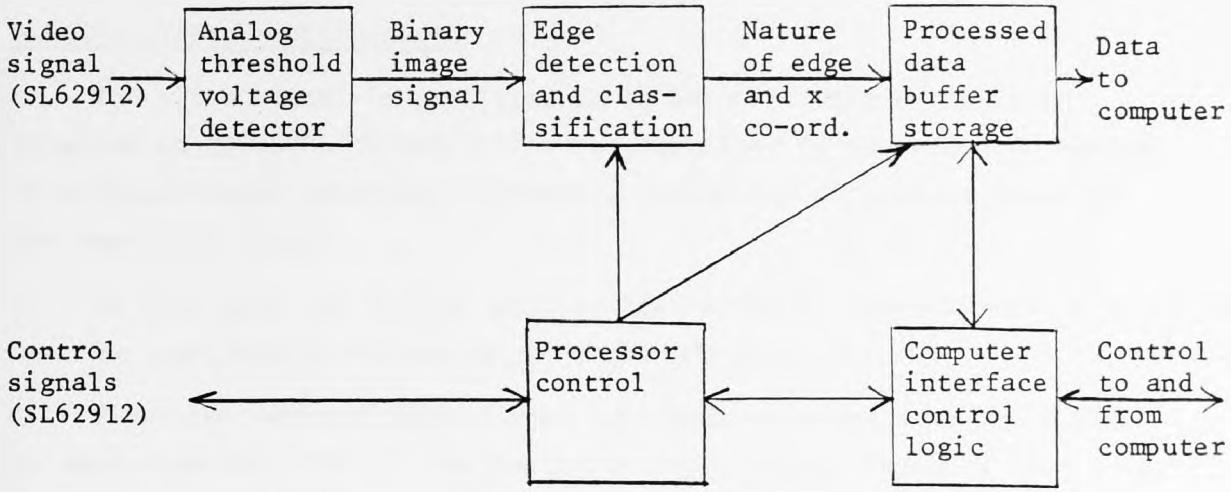


Figure A.4: Main functional areas of silhouette image preprocessor

A.4.1: Analog threshold voltage detector

Circuit diagram: Sheet 1

A.4.1.1 Video input

The video signal from the SL62912 is output via a coaxial cable of 50Ω impedance and is connected to both experimental circuits on the scanning system, as shown in Figure A.5.

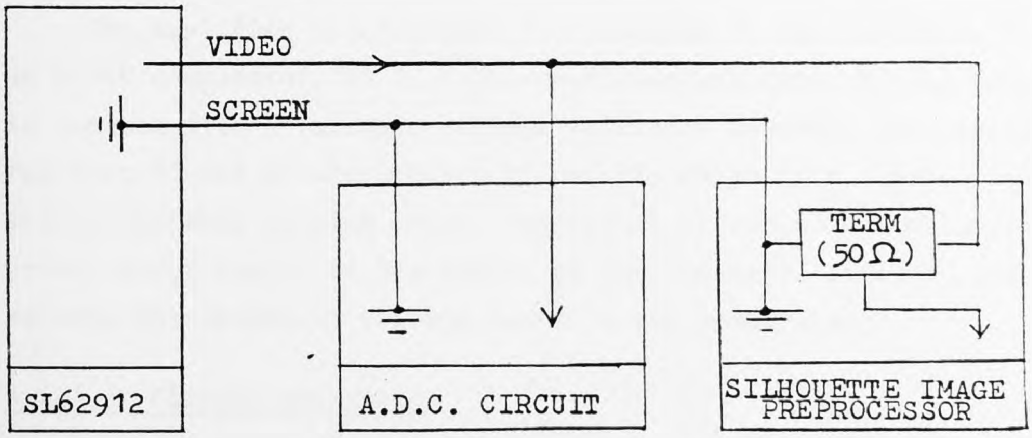


Figure A.5: Connection of video signal

The termination network, located in the silhouette image preprocessor, consists of resistors R1 and R2, and capacitor C1, on Sheet 1.

A.4.1.2 Video amplification

The video signal is amplified by an RS1445 wideband amplifier (similar to MOTOROLA MC1445) IC7. The amplifier is connected to operate as a single-ended inverting amplifier. Capacitor C4 reduces noise at the amplifier output.

In this mode the voltage gain of the RS1445 is approximately x 10 (20 dB) and the peak output voltage is approximately 2.5V.

Since the maximum video signal amplitude expected from the SL62912 is approximately -700 mV, the amplifier input signal is taken from a tap of the 55Ω line termination resistance, so that the maximum input to the amplifier is approximately -250 mV. This means that when a maximum signal is received, the amplifier output is clipping.

The lack of linearity which is produced is not a disadvantage in this application, since the end result of this stage of processing is a binary signal obtained by thresholding the amplifier output. The distorted peak signals are not considered in this process.

In addition, the thresholding proces does require the maximum amplification of the middle range of video amplitudes, so that signals of different amplitude within this region can be more easily distinguished.

A.4.1.3 Amplified video thresholding

The amplified video signal is connected to the inverting input of an LM306 comparator, IC 11. The non-inverting input of the comparator is connected to a variable voltage reference network, comprising resistor R3 and potentiometers P1 and P2, which form a potential divider with a variable tapping point, capacitors C2 and C3 which smooth the 5-volt power supply in the region of the potential divider, and C5 which smooths the reference voltage input to the comparator.

A.4.1.4 Circuit operation

When the video signal is at a large negative amplitude (i.e. down to approximately -700 mV max.) it means that there is no object in that position.

In this circumstance the amplifier input is correspondingly negative, which makes the amplifier output correspondingly positive.

The non-inverting input of the comparator will have previously been set, via P1 and P2, to a value more negative than the amplifier output at

this time, so that the inverting pin 4 is more positive than the non-inverting pin 3. The comparator output is thus logic low.

When the video signal is near earth potential it means that an opaque object is in that position in the field of view.

In this circumstance, the amplifier input is near earth and the amplifier output is also near earth.

The voltage on the inverting pin 4 of the comparator should now be more negative than that at the non-inverting pin 3. The comparator output is thus logic high.

The comparator output goes to the edge detection and classification circuit for further processing, as signal "COMPOUT".

The threshold voltage detector operates continuously. The succeeding logic determines at what time data output from the circuit is valid..

A.4.2 Edge Detection and Classification Circuit

Constituent elements: (see block diagram)	x-co-ordinate counter	
	Delay latches	
	This scan store	} Processing stores
	Previous scan store	} Processing stores
	Edge detector	
Circuit diagram:	Part of Sheet 4	
	Sheet 5	

A.4.2.1 x co-ordinate counter (Sheet 4)

The counter consists of four 74LS161, 4 bit binary counter, IC5, IC6, IC7 and IC8.

The counter is enabled when signal "CLEAR COUNT" goes high and counts on the positive edge of clock $\overline{CK/2}$. When "CLEAR COUNT" is low the counter is cleared.

The 11 LSB's of the counter are used to address the processing stores (this scan and previous scan stores) on Sheet 5, and are also clocked through 74LS174 registers, IC32 and IC33, to become the x co-ordinate of the previous video point which was processed.

When signal "EED2" is low, IC32 and IC33 are cleared, i.e. when not processing a scan.

The counter output is also decoded for control purposes (see section A.4.4).

A.4.2.2 Delay latches (Sheet 5)

The latches are 74LS174 registers IC21 and IC38.

They are connected in shift register fashion to make three 3 bit long registers, so that three adjacent video points from each of three adjacent scans and from the same x co-ordinates in each scan are available at the same time for processing.

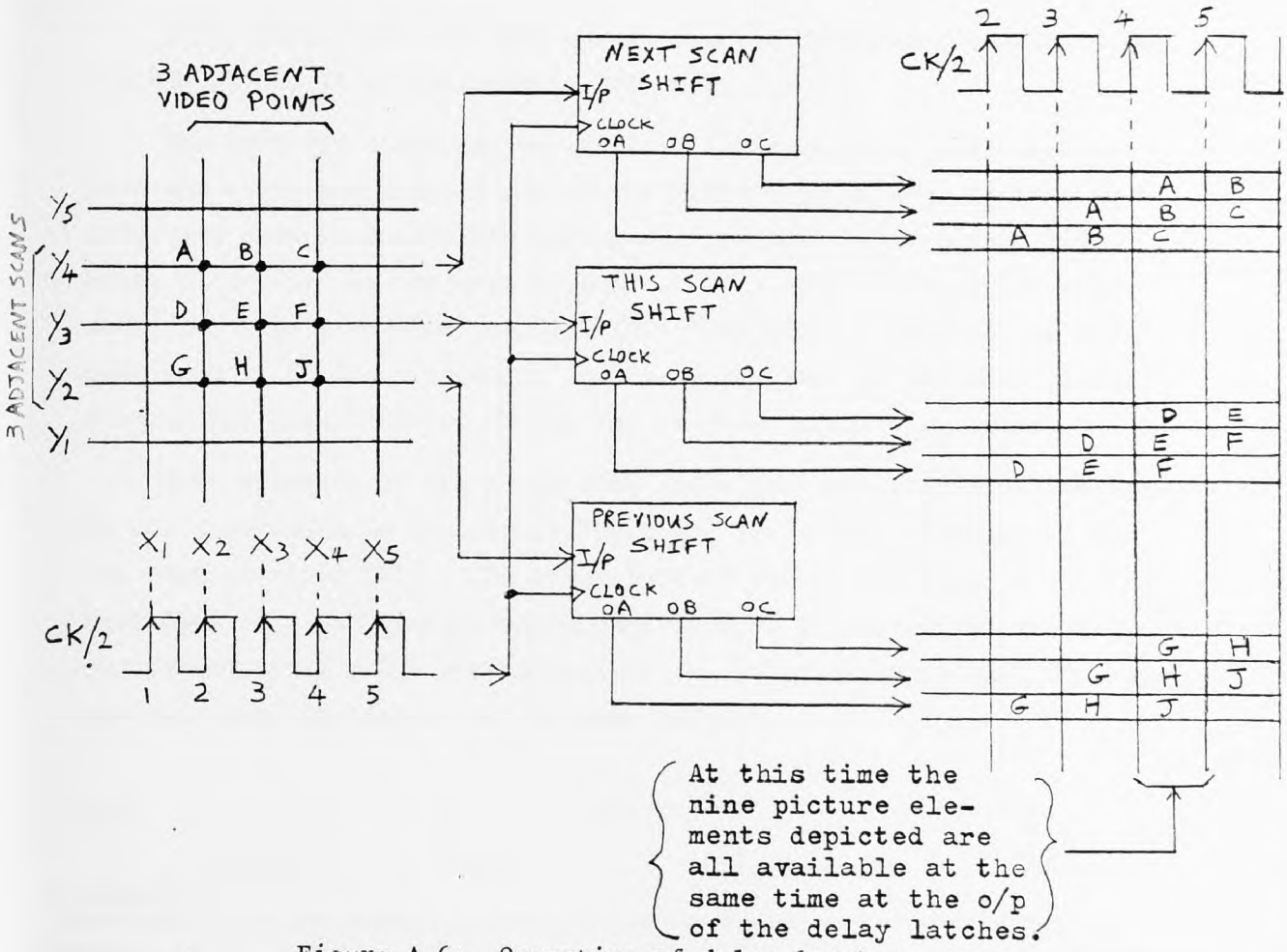


Figure A.6: Operation of delay latches

In Figure A.6, the scan at Y4 is denoted the "NEXT SCAN" and represents the data currently entering the processor from the SL62912. The scan at Y3 is denoted "THIS SCAN" and represents the data which were received from the SL62912 during the last processing cycle (it was then the "NEXT SCAN") and stored in the "THIS SCAN STORE". The scan at Y2 is denoted the "PREVIOUS SCAN" and represents the data which were read from the "THIS SCAN STORE" during the last processing cycle (it was then

"THIS SCAN") and stored in the "PREVIOUS SCAN STORE". It is also the data which were received from the SL62912 two processing cycles ago.

New data appear at the input to the delay latches shortly before the rising edge of clock CK/2 and are clocked into the first stage on that edge. Data are subsequently shifted along on rising edges of CK/2.

The nine binarised video points, representing the element of the picture frame under consideration, which are output during any particular cycle of CK/2, are input to edge detector for processing together.

A.4.2.3 Processing stores (Sheet 5)

IC14, IC15, IC25 and IC26 constitute the processing stores. They are MCM 93425, 1K x 1 bit static RAM chips.

The RAMS are organised as two 2K x 1 bit stores. IC14 and IC15 form one store and this is called the "THIS SCAN STORE", because the data read from it during processing are the scan whose points are being classified by the edge detector. IC25 and IC26 form the other store and this is called the "PREVIOUS SCAN STORE", because the data read from it during processing are the scan whose points were classified by the edge detector during the previous processing cycle.

Both elements of the processing store are addressed simultaneously by the x co-ordinate counter on Sheet 4. The address changes on the -ve edge of clock CK/2. The first half of the clock cycle is used to read data from the stores, which were written in during the previous processing cycle. The second half of the clock cycle is used to write new data into the stores at the same address.

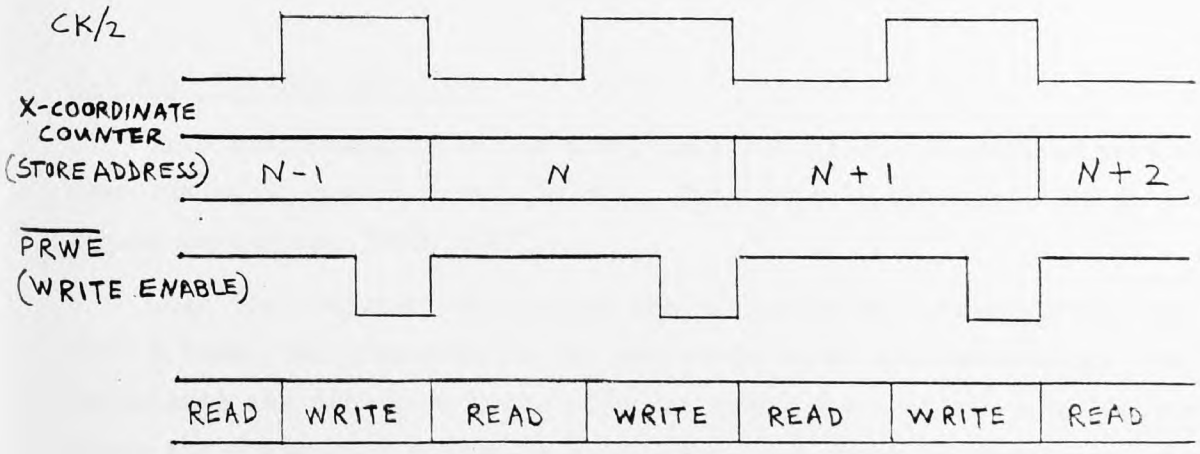


Figure A.7: Processing stores cycle timing

The data which are output from the "THIS SCAN" store are latched into the first stage of a delay latch on the rising edge of CK/2. The latched data are then input to the "PREVIOUS SCAN" store where they are written in during the next half clock cycle. In this way the "THIS SCAN" data are transferred to the "PREVIOUS SCAN STORE" for use as "PREVIOUS SCAN" data during the next processing cycle.

Similarly, the data being input from the SL62912 are latched through a delay latch on the rising edge of CK/2 and written into the "THIS SCAN" store during the second half clock cycle. In this way the "NEXT SCAN" data are stored in the "THIS SCAN" store for use as "THIS SCAN" data during the next processing cycle.

A.4.2.4 Edge detector (Sheet 5)

IC20, which is an N82S100 F.P.L.A. (field programmable logic array), is the edge detector.

The 9 bits of data corresponding to three adjacent video points from three adjacent scans, which were described in section A.4.2.2 and Figure A.6, are input to the N82S100 chip for processing. However, since the three scan method, described in section A.3.2, is used for processing, only five of the nine points which are input are considered.

The N82S100 chip is enabled to process data by signals "EED1" and "EED2". Both signals must be high to enable the chip.

Data to the N82S100 change on the +ve edge of CK/2 and the results of comparison are output to IC31, a 74LS174 register, where they are latched out on the -ve edge of CK/2.

The F.P.L.A. program is shown on the following page.

A.4.2.5 Circuit operation

Each processing cycle (or scan) consists of three distinct periods (see timing diagram - frame timing). This circuit operates only in the region called the "REAL SCAN".

When the computer has enabled the processor by setting FNCT 1 and FNCT 2 high, the processor is not yet ready to do any processing. This is because the data from three adjacent scans are required simultaneously, since the three scan method is being used. In practice this means that there must be a valid scan stored in the "PREVIOUS SCAN" store, the following scan stored in the "THIS SCAN" store and the next scan must

16X48X8 FPLA PROGRAM TABLE

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) _____

CUSTOMER SYMBOLIZED PART # _____

DATE RECEIVED _____

COMMENTS _____

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____

REV _____ DATE _____

PROGRAM TABLE ENTRIES

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	$\overline{Im}$	Don't Care	Prod. Term Present in Fp	Prod. Term Not Present in Fp	Active High	Active Low
H	L	— (dash)	A	• (period)	H	L

NOTE

Enter (—) for unused inputs of used P-terms.

NOTES

1. Entries independent of output polarity.
2. Enter (A) for unused outputs of used P-terms.

NOTES

1. Polarity programmed once only.
2. Enter (H) for all unused outputs.

PRODUCT TERM*																	ACTIVE LEVEL*								
	INPUT VARIABLE*																								
NO.	15	14	13	12	11	10	E 9	8	7	6	B 5	C 4	D 3	2	A 1	0	OUTPUT FUNCTION* 7	6	5	4	3	2	1	0	
0	H	L	L	L	L	H	H	—	H	—	H	H	H	—	L	—	A	A	A	A	A	A	•	•	U
1	H	L	L	L	L	H	H	—	L	—	H	H	H	—	H	—	A	A	A	A	A	•	A	•	D
2	H	L	L	L	L	H	H	—	—	—	L	H	H	—	—	—	A	A	A	A	A	A	A	•	
3	H	L	L	L	L	H	H	—	—	—	H	H	L	—	—	—	A	A	A	A	A	•	•	A	R
4	H	L	L	L	L	L	—	—	—	—	—	—	—	—	—	—	A	A	A	A	•	•	•	•	} EN
5	H	L	L	L	L	—	L	—	—	—	—	—	—	—	—	—	A	A	A	A	•	•	•	•	
6																									
7																									
8																									
9																									
10																									
11																									
12																									
13																									
14																									
15																									
16																									
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40																									
41																									
42																									
43																									
44																									
45																									
46																									
47																									

* Input and Output fields of unused P-terms can be left blank. Unused inputs and outputs are FPLA terminals left floating.

be entering the processor, before it is valid for the edge detector to look for edges.

Thus, signal EED1, which enables the edge detector on pin 26 of IC20, does not go high until the third valid scan is being input to the processor. Apart from this consideration the operation of each processing cycle is as follows:

Shortly following the start of a "REAL SCAN" period the first video pulse (corresponding to the first call of the photodiode array) arrives at the processor. Its binary equivalent appears at IC21, pin 3, as "COMPOUT".

At this time signal "CLEAR COUNT" is low, clearing the x co-ordinate counter, so that the processing stores are addressed at location zero. The "THIS SCAN" store thus has at its output IC14, pin 7, the binary equivalent of the first video cell of the previous scan. Similarly, the "PREVIOUS SCAN" store has at its output IC25 pin 7, the binary equivalent of the first video cell of the last but one scan. These two signals are input to register IC38.

At the next +ve edge of CK/2 the three items of data are clocked into the first stage of the delay latches (IC21, IC38).

At this time also, signal "CLEAR COUNT" goes high enabling the x co-ordinate counter to count subsequent $\overline{CK/2}$ clock pulses, and signal "$\overline{PRWE}$" becomes active producing write enables for the processing stores.

After the +ve edge of CK/2, IC21 pin 2, which is the first cell of the new scan, is input to the "THIS SCAN" store and IC38 pin 2, which is the first cell of the previous scan, is input to the "PREVIOUS SCAN" store. The "$\overline{PRWE}$" pulse which occurs subsequently writes these data into the stores, replacing the data which were just read out, with the corresponding data from the next scan.

The next -ve edge of CK/2 clocks the x co-ordinate counter to value 1, addressing the next location of the processing stores.

As before, the contents of the new location and the second video of the new scan are now input to the delay latches. On the next +ve clock they are loaded into the first delay stage, replacing the data of the first video location, which are transferred to the second delay stage. The latched data, in the first delay stages, are then written into the processing stores, replacing the old data at location 1 of the stores, which were just read out.

Subsequent video cells are input and transferred to, and between, stores in a similar fashion.

After the third video cell has been input as described above, the outputs of the delay latches always contain nine valid video points which can be processed in the edge detector. However, because of edge effects which cause deterioration of the video signal at each end of a scan, edge detection does not start when three video cells have been input, but is delayed until 256 cells have been input, to avoid the difficult area at the start of the scan. At this time, signal "EED2" goes high on a -ve edge of CK/2 which enables the edge detector. The nine video points are then processed and the results latched out, as previously described, until "EED1" goes low at the end of the "REAL SCAN" period.

The x co-ordinates and the results of edge detection are latched out on -ve edges of CK/2 in registers IC33, IC32 and IC31 and transmitted to the buffer storage function.

A.4.3 Processed Data Buffer Storage Function

Constituent elements	Edge position store
(see block diagram)	Store address counter
	Store write/transmit control
Circuit diagram:	Sheet.6
	Sheet 7
	Sheet 8
	Sheet 10

A.4.3.1 Edge position store (Sheet 10)

This store consists of 28 MCM93425 1K x 1 bit static RAMS arranged as a 2K x 14 bit store.

The store receives 14 bits of data from the edge detection and classification circuit during the "REAL SCAN" processing period. The data consist of 3 bits which indicate the nature of the edge detected, and 11 bits which are the x co-ordinate of the detected edge.

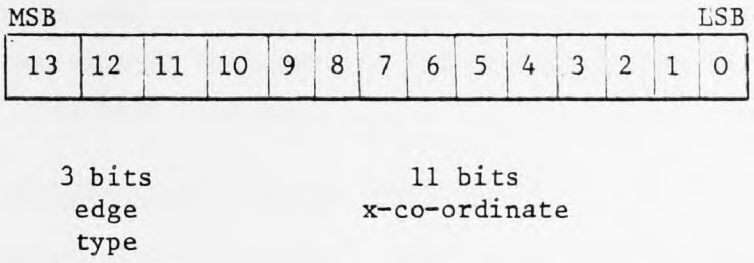


Figure A.8: Processed data format

An 11 bit address is received from the store address counter, together with a write enable signal $\overline{SWE}$ from the store write/transmit control circuit.

The store output is buffered by IC27 and IC28 74LS244 octal buffers before transmission to the DR11B interface.

A.4.3.2 Store address counter (Sheet 7)

The counter consists of 3 74LS161 4 bit binary counters, IC8, IC9 and IC10.

The signal $\overline{ADDCLR}$ controls the counter operation, so that when it is low the counter is cleared and disabled, and when high the counter increments on +ve edges of signal "STORECK".

The 11 bit counter output goes to the store transmit logic and is used to detect the end of transmission when data are being transmitted to the computer.

The 10 LSB's of the counter are used as the chip address for the edge position store. The MSB of the counter is a chip select used to select the least significant half of the store (i.e. 0-1023) and the inverted MSB is a chip select used to select the most significant half of the store (i.e. 1024-2047).

The store address is buffered by IC15 and IC17 before going to the edge position store.

A.4.3.3 Store write/transmit control (Sheet 6 and Sheet 8)

This functional block has two main functions:

(1) To control the writing of processed data into the edge position store, during processing.

(2) To control the output of data from the store during transmission to the computer and to detect when all the stored data have been sent.

A.4.3.3.1 Address counter clock (Sheet 6)

The counter clock, "STORECK", originates from two different places depending on whether data are being input or output, the course being controlled by multiplexer IC16.

During the "REAL SCAN" signal "$\overline{\text{EEDW}}$" is low which selects pin 2 of IC16 as the clock source. When signal "RAM ENABLE" goes high on a -ve edge of clock CK/2, pin 10 of IC12 primes the NAND gate to generate a clock. On the subsequent +ve edge of CK/2, the output of the NAND gate goes low. On the succeeding -ve clock edge both "RAM ENABLE" and the clock go low, so the NAND gate output goes high. This positive edge is sent through IC16 to clock the address counter to the next address (Figure A.9).

The edge data were stored, during the clock cycle preceding the +ve edge of "STORECK", at the address on the counter before it was incremented.

When data are being transmitted to the computer, signal "$\overline{\text{EEDW}}$" is high, which selects pin 3 of IC16 as the clock source. When a word of data has been transmitted by the DR11-B it pulses signal "END CYCLE". The trailing edge of this pulse clocks the address counter, as "STORECK", so that a new item of data is addressed for transmission.

A.4.3.3.2 Store write enable (Sheet 6)

Signal $\overline{\text{SWE}}$ is the store write enable. During the "REAL SCAN" period signal $\overline{\text{STOREWE}}$ pulses during every -ve clock half cycle of CK/2. When "RAM ENABLE" goes high (when an edge is detected) on a -ve clock edge, pin 5 of IC12 primes the NAND gate to produce a write enable. When a $\overline{\text{STOREWE}}$ subsequently occurs, it is inverted by an element of IC13 and makes $\overline{\text{SWE}}$ pulse low, thus writing the data into the address currently on the address counter. The address then increments in preparation for the next data, as previously described.

Pullup resistor R6 increases noise immunity on $\overline{\text{SWE}}$.

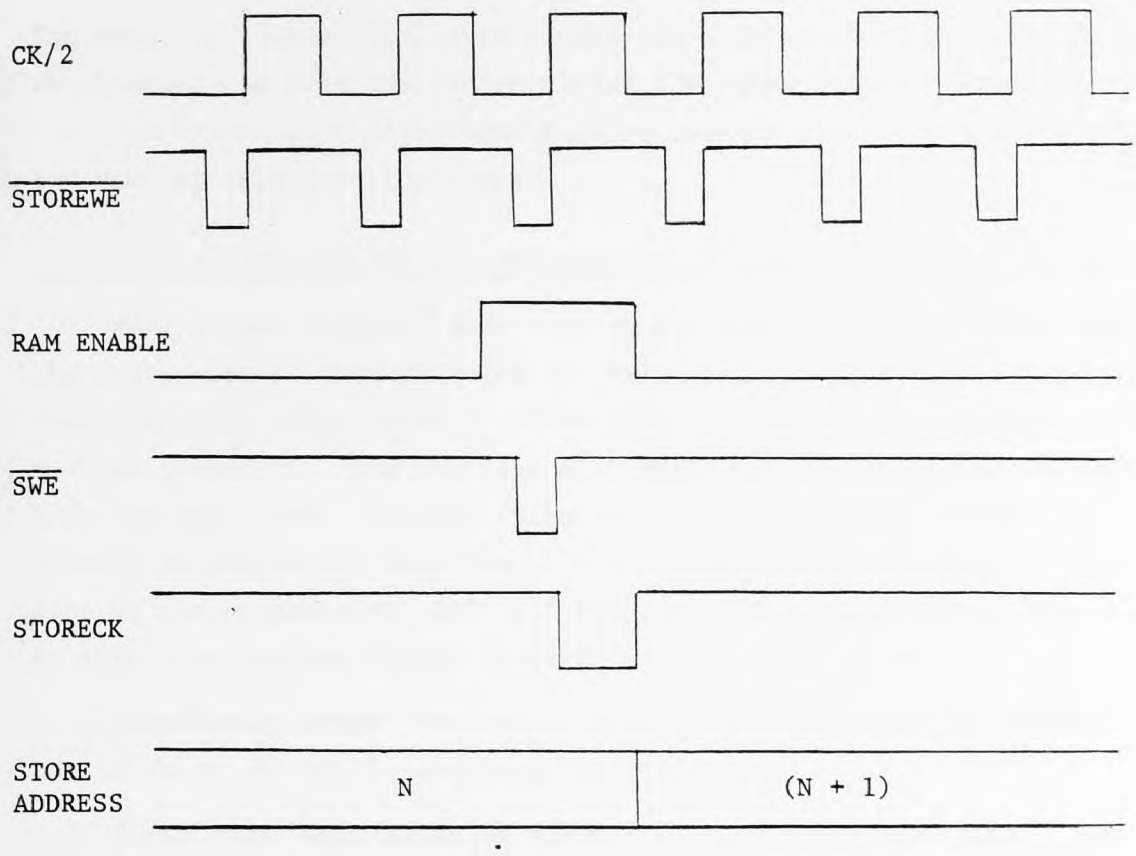


Figure A.9: Writing data to edge position store

A.4.3.3.3 Clearing the address counter (Sheet 6)

Signal $\overline{\text{ADDCLR}}$ clears the address counter. The counter needs to be cleared at two different times: at the beginning of the "REAL SCAN", so that processed data are stored starting at address zero, and also prior to data transmission so that all the data are sent.

Just before the "REAL SCAN" starts, signal $\overline{\text{DSCAN}}$ is low and ΦXB is high (see timing diagram -"START OF REAL SCAN TIMING"). At this time then pins 12 and 13 of IC12 are high which makes $\overline{\text{ADDCLR}}$ low, and the counter is cleared. On the -ve edge of ΦXB, $\overline{\text{DSCAN}}$ goes high. Thus, pins 12 and 13 of IC12 are both low. Also at this time IC1 pin 8 is high, so $\overline{\text{ADDCLR}}$ goes high, removing the counter clear, in preparation for storing processed data.

At the start of the "DUMMY SCAN" period (when data are to be transmitted to the computer) signal "COMPEN" goes high, making pins 2 and 12 of IC1 high. The next +ve edge of $\overline{\text{CK/2}}$ (this is the first clock edge when the clock resumes after the ΦXB sync pulse; see: end of stepping - start of dummy scan timing) clocks the high on pin 12 into the D-type making pin 8 go low. This makes $\overline{\text{ADDCLR}}$ go low, clearing the counter.

The next -ve edge of $\overline{CK/2}$ then clocks the high on pin 2 into the D-type which makes pin 6 go low, which clears the other D-type element at pin 13. Pin 8 thus goes high, removing the counter clear, in preparation for reading data from the store.

A.4.3.3.4 Detecting end of transmission to computer (Sheet 6 and 8)

When signal "COMPEN" goes high at the beginning of a "DUMMY SCAN", the store address counter holds the address of the last store location where data are stored plus 1. This address is input to registers IC5 and IC6 (Sheet 8). The +ve edge of "COMPEN" clocks the store address into the registers. Shortly following this the address counter is cleared as previously described. The stored address is input to one side of the comparator: IC2, IC3 and IC4. The comparator is disabled at this time because signal "COMPOP" is low.

Immediately after the address counter has been cleared "COMPOP" goes high at IC1 pin 5, enabling the comparator.

If no edges were detected during the preceding "REAL SCAN", the address stored in registers IC5 and IC6 will be zero. So, when "COMPOP" enables the comparator a comparison is obtained immediately and signal "ENDOP" goes high, indicating that no data need be transmitted.

If edges were found in the "REAL SCAN", then the address on IC5 and IC6 will be non-zero and transmission of data will proceed as described in section A.4.3.4, until the address counter has been incremented by "END CYCLE" pulses to equal the address in IC5 and IC6. "ENDOP" is then generated to indicate that transmission can stop.

A.4.3.4 Circuit operation

Data are stored in the edge position store during the "REAL SCAN" when a scan is being processed, and data are read from the store and transmitted to the computer during the "DUMMY SCAN".

At the beginning of a "REAL SCAN" the store address counter is cleared. During subsequent processing, when an edge is detected, signal "RAM ENABLE" goes high, which enables data to be stored and the address counter to be incremented.

At the start of the "DUMMY SCAN" the counter is cleared in preparation for reading out the data previously stored. Also, the final address reached during processing is saved, so that end of transmission can be

detected. Transmission then proceeds, the address counter incrementing through the store under the control of the DR11-B, until all the data have been sent. Signal "ENDOP" is then generated to indicate that transmission should cease.

A.4.4 Processor Control

Constituent elements:	Control Logic	
(see block diagram)	Start processing	Control decoders
	Stop processing	
	Delay for stepping	
	Clock generator	
Circuit diagram:	Sheet 2	
	Sheet 3	
	Sheet 4 (part of)	

A.4.4.1 Clock generator (Sheet 2)

This circuit produces clocks to drive the processor and SL62912 development board.

The basic clock is generated by a "Radio Spares" RS 307-430, 10 MHz crystal clock circuit, IC16. The crystal output is buffered, by a NAND buffer, an element of IC11 74LS37, and input to IC2, a 74LS163 4 bit binary counter. The counter is free running and produces two clocks: a 5 MHz clock at pin 14 and a 0.625 MHz clock at pin 11.

The 5 MHz clock is sent as clocks "CK" and "$\overline{CK}$" to drive the processor logic. In addition, both the 5 MHz and 0.625 MHz clocks are input to IC4, a 74LS157 multiplexer. One or other of the two clocks is selected for transmission to the SL62912, by signal "DSCAN". When this signal is low, the fast clock is sent, when it is high the slow clock is sent.

The clock to the SL62912 can be switched off at pin 2 of IC10. When signal "$\overline{ESCAN}$" goes low the D-type, an element of IC18, is cleared so that pin 8 goes high and forces the output of the NOR gate low. The clock is restarted when signal "$\overline{ESTEP}$" occurs. The -ve edge of this pulse, inverted at IC29 pin 6, switches the D-type so that pin 8 goes low, enabling clock transmission. Signal "SSYNC" which is low when the processor is disabled, from the computer, ensures that "CCDCK" is running

when the processor is first enabled.

A.4.4.2 Decode start processing (Sheet 4)

As described in section A.4.2.5, processing of video data does not start until the 256th video cell is received from the SL62912.

This is achieved by decoding count 256 of the x co-ordinate counter at pin 9 of IC37. The decode is enabled by signal "ENWE" which is high only during the "REAL SCAN" period. The signal "SSCAN", resulting from the decode, initiates scan processing.

A.4.4.3 Decode stop processing (Sheet 4)

There are 1728 cells of video output by the SL62912, but this circuit only processes up to cell 1598 because of degradation of the video signal at the end of the scan. (N.B. cell No. 1599 is also accepted but used only as part of the processing of cell 1598.)

The end of processing is decoded in a similar manner to the start. A count of 1598 is decoded from the output of the x co-ordinate counter in elements of IC30, IC24 and IC19, to produce signal "$\overline{\text{ESCAN}}$". The decode is enabled by two signals, "$\overline{\text{DSCAN}}$" which is high when it is not the dummy scan and "ENESCAN" which is high during a "REAL SCAN" and goes low when "$\overline{\text{ESCAN}}$" goes low, terminating the "REAL SCAN".

A.4.4.4 Decode delay for stepping (Sheet 4)

At the end of the "REAL SCAN" period the computer is still in the process of stepping the table to a new position. This was started at the beginning of the real scan and, since the stepping is slow compared to processing, it continues for several milliseconds beyond the end of the "REAL SCAN".

Now, since the main reason for having the "DUMMY SCAN" is to clear the photodiode array of the unreliable video data acquired while the table is stepping to a new position, the "DUMMY SCAN", in particular the transfer pulses at the start of the "DUMMY SCAN" (ΦXA and ΦXB), cannot take place until stepping is complete.

Because of this, the clock to the SL62912,"CCDCK", is stopped at the end of the "REAL SCAN", as previously described, until stepping is finished. This halts the operation of the SL62912 so that it cannot transfer the charge in the photodiodes into the CCD shift register prematurely.

This delay is counted by the x co-ordinate counter and the end of the delay is decoded by the 4 input NAND gate, an element of IC19, to produce signal "$\overline{\text{ESTEP}}$", indicating that the processor and SL62912 can resume operating.

A.4.4.5 Control logic (Sheet 3)

This circuit controls the overall operation of the processor by producing a set of control signals in response to external synchronising signals and internal status signals.

A.4.4.5.1 Generation of CK/2 clocks (timing diagram - start of "REAL SCAN")

Clock signal "$\overline{\text{CK}}$" is received from the clock generator and is input to pin 3 of IC17, a D-type. The D-type is arranged to divide clock "CK" by two. Synchronising signal ΦXB, from the SL62192, is inverted at pin 5 1 and 3 of IC37, and the inverted signal is input to the preset input of the D-type, at pin 4 of IC17. Thus, when ΦXB goes high at the start of each scan, pin 5 of IC17 is forced high, i.e. the clock CK/2 is stopped for the duration of ΦXB. When ΦXB goes low again, pin 5 is clocked low by the next +ve edge of "$\overline{\text{CK}}$" and subsequently produces CK/2 clocks until the next ΦXB occurs.

This ensures that CK/2 always starts the same way round at the beginning of a "REAL SCAN".

Pin 5 of IC17 is output to an array of gates, 2 elements of IC11 74LS37 and 1 element of IC10 74LS02, to produce a set of CK/2 and $\overline{\text{CK/2}}$ clocks for use around the processor.

Generation of CK/2 clocks is enabled by signal "SSYNC" at pin 13 of IC11.

A.4.4.5.2 Processing initiation (timing diagram - frame timing)

Processing is initiated by a signal from the computer which appears at this circuit as signal "SOFTWARE SYNC" (FNCT2 from DR11-B).

When the computer is not ready to receive data from the processor "SOFTWARE SYNC" is low. This, appearing at IC17 pin 13, clears the D-type, so that pin 9 is low, i.e. "SSYNC" is low. This signal then holds the processor initialised.

When the computer sets "SOFTWARE SYNC" high the clear is removed from the D-type. The processor is not, however, immediately enabled since processing can only start at the beginning of a scan. On the next

+ve edge of ΦXB, the D-type is clocked at pin 11 which makes "SSYNC" go high. Processing then starts after the following -ve edge of ΦXB, when CK/2 starts.

A.4.4.5.3 REAL SCAN and DUMMY SCAN generation (- frame timing)

The D-type flip-flop, part of IC18, is used to define when a "DUMMY SCAN" should take place.

The inverted ΦXB signal is used to clock the D-type at IC18 pin 3. The D-type is arranged as a divide by 2 circuit so that every other ΦXB pulse causes signal "DSCAN" to go high, thereby specifying every other scan as a "DUMMY SCAN".

Signal "SSYNC" from IC17 pin 9 holds the D-type preset when the processor is not enabled. This means that when the computer enables the processor, the first ΦXB pulse following, clocks over the D-type making "DSCAN" go high. The first scan is thus a "DUMMY SCAN".

"DSCAN" and "$\overline{\text{DSCAN}}$" are distributed as required.

A.4.4.5.4 Start of scan timing

Two signals are generated at the start of each scan, used during the "REAL SCAN", to enable processing control functions.

The inverted ΦXB signal is input to IC27 pin 3 and pin 1. When ΦXB is high at the beginning of a scan IC27 pin 3 and pin 1 are low. The register is thus cleared. When ΦXB goes low pins 1 and 3 go high removing the clear and setting D1 input high.

The CK/2 pulses, which occur subsequently, clock the high on pin 3 through the 74LS174 register in shift register fashion, so that 6 signals are produced, each one going high 1 CK/2 clock cycle after the previous one.

The signal at IC27 pin 2, which occurs on the first +ve edge of CK/2 after ΦXB, is used as an enable for the processing enable latch and the "WRITE ENABLE" enable latch, elements of IC28, and is also used to clock the "DELAY TWO SCANS" counter, IC22.

The signal at IC27 pin 10, which occurs on the 4th +ve edge of CK/2 after ΦXB, is used to enable the x co-ordinate counter and also to clock the "WRITE ENABLE" enable latch, so that WRITE ENABLES can be generated for the processing stores and edge position store. The delay of three clock cycles is to allow the first three video cells, output from the

SL62912, which are always blank, to be disregarded.

A.4.4.5.5 Controlling the x co-ordinate counter

Signal "CLEAR COUNT" at IC29 pin 8 controls the operation of the x co-ordinate counter.

During the occurrence of ΦXB at the beginning of a "REAL SCAN" pin 10 of IC27 is low which makes pin 13 of IC29 low. Thus, "CLEAR COUNT" is low and the x co-ordinate counter is cleared. This clear is removed when IC27 pin 10 goes high, as described in section A.4.4.5.4, and the counter counts x co-ordinates.

When signal "$\overline{\text{ESCAN}}$" goes low at the end of a "REAL SCAN", pin 12 of IC29 goes low making "CLEAR COUNT" low also. This clears the counter and removes "$\overline{\text{ESCAN}}$", so "CLEAR COUNT" goes high again. The counter then proceeds to count the delay for stepping.

When stepping is complete, the counter is again cleared by ΦXB at the start of the "DUMMY SCAN". The counter counts during the "DUMMY SCAN", but it has no significance at this time.

A.4.4.5.6 Processing enable latch

Signal "EED2" ($\overline{\text{EEDW}}$, AEEDW) is the scan processing enable, defining which part of the REAL SCAN is to be processed (i.e. cell 256 to cell 1598).

When ΦXB is high, at the start of a "REAL SCAN", pin 2 of IC27 is low, which makes pin 10 of IC10 low also. This clears the D-type, IC28, at pin 1 making IC28 pin 5 and signal "EED2" low and disabling scan processing. At the end of ΦXB the clear is removed.

If the succeeding scan is a "DUMMY SCAN", signal "DSCAN" is set high by the trailing edge of ΦXB (IC18 pin 6) which makes IC10 pin 8 high. IC10 pin 10 then stays low, maintaining the clear through the "DUMMY SCAN" and disabling processing (EED2 is low).

In a "REAL SCAN" signal "SSCAN" pulses high when the x co-ordinate counter reaches count 256. This signal clocks the processing enable latch at IC28 pin 3, making "EED2" go high and partly enabling scan processing. (N.B. EED1 is also required, see below.) When the x co-ordinate counter reaches count 1599, in a "REAL SCAN", signal "$\overline{\text{ESCAN}}$" is generated, which makes pin 1 of IC29 low and which re-asserts the clear at IC28 pin 1. This makes "EED2" low at the end of the "REAL SCAN",

disabling processing during the "DELAY FOR STEPPING".

A.4.4.5.7 Write enable generation

Two write enable signals are produced: "$\overline{\text{PRWE}}$" at IC30 pin 12 for the processing stores and "$\overline{\text{STOREWE}}$" at IC30 pin 6 for the edge position store.

"PRWE" is generated by gating together the clocks CK/2 and CK at pins 2 and 13 of IC30, together with an enable signal at IC30 pin 1 from IC28 pin 9.

"$\overline{\text{STOREWE}}$" is generated by gating together the clocks $\overline{\text{CK/2}}$ and CK at pins 4 and 5 of IC30, together with the enable at pin 3.

These write enables are only generated during the time that video data are being processed in a "REAL SCAN". Generation is enabled when pin 9 of IC28 is set high.

At the start of a scan, when ΦXB is high, IC10 pin 10 is low, as previously described, and the "WRITE ENABLE" enable latch is cleared at IC28 pin 13, making IC28 pin 9 low and disabling generation of write enables. If the succeeding scan is a "DUMMY SCAN", signal "DSCAN" is set high by the trailing edge of ΦXB which then maintains a low at pin 10 of IC10 during the scan. If the succeeding scan is a "REAL SCAN" the clear on IC28 pin 13 is removed when ΦXB goes low and IC27 pin 2 is clocked high. At the 4th +ve edge of CK/2 following the end of ΦXB, IC27 pin 10 is clocked high. This signal is used to set the "WRITE ENABLE" enable latch by clocking IC28 at pin 11. IC28 pin 9 then goes high and write enables are produced.

The timing of the start of write enable generation coincides with the time the x co-ordinate counter is enabled.

When the x co-ordinate counter has reached count 1599, signal "$\overline{\text{ESCAN}}$" occurs at IC29 pin 1, which causes the clear signal at IC10 pin 10 to occur again, clearing the "WRITE ENABLE" enable latch and disabling write enable generation.

A.4.4.5.8 Delay two scans counter

Since the "THREE SCAN METHOD" is employed in this circuit, the data from two complete scans must be stored in the processing stores, before processing can take place, when the third scan is being received.

In order to allow two scans of data to be stored when processing is

initiated from the computer the "DELAY TWO SCANS" counter delays the production of the edge detector enable signal "EED1" until the data for the third scan of a frame are entering the processor (see timing diagram - frame timing).

IC22 is the delay counter and is arranged to count 6 ΦXB pulses from the time "SOFTWARE SYNC" is received until it generates "EED1", i.e. the first scan consists of a dummy scan and a real scan which is 2 ΦXB pulses. The second scan also contains 2 ΦXB pulses, giving 4 pulses. The third scan consists first of a dummy scan which is preceded by the 5th ΦXB and a real scan which is preceded by the 6th ΦXB, at which point processing is enabled.

The load inputs of IC22, pins 3, 4, 5, 6, are hardwired to value 2 (binary 0010). The load signal IC22 pin 9 is connected to signal "SSYNC" and, as previously described, "SSYNC" is low before the computer enables the processor. The clock input of IC22, pin 2, is controlled by ΦXB through IC27. When the processor is disabled the SL62912 runs freely on clock "CCDCK", so a stream of ΦXB pulses is continuously generated which makes IC27 pin 2 pulse correspondingly clocking IC22 pin 2. So with the counter input set to 2, the load pin set low and clocks at pin 2, the counter is loaded with value 2 continuously, before the processor is enabled.

After a computer enable "SSYNC" goes high at the first +ve edge of ΦXB following. This makes pin 9 of IC22 high and the counter stops loading. Since the counter output is at value 2 at this time, the D-output, pin 11 is zero, which, inverted, makes the P and T inputs of the counter (pins 7, 10) high, which enables the counter to count. Subsequent -ve edges of ΦXB, clocked through IC27, are then counted. When 6 ΦXB's have occurred, the counter output is at value 8. Thus, the D-output is high, making "EED1" high. Also, the same signal inverted makes P and T low, disabling further counting. "EED1" thus remains high thereafter until the processor is once more disabled.

A.4.5 Computer Interface Control

Circuit diagram: Sheet 9

A.4.5.1 Control signals from computer

Two control signals are received from the computer, "FNCT2", which is the processor enable signal, and "FNCT1" which is used to enable the generation of control signals from the processor to the computer.

"FNCT1" is buffered at IC36 pin 4 and then goes to IC12 pin 13 to enable generation of interrupts and also to IC9 pin 2 to ultimately enable generation of D.M.A. control signals.

"FNCT2" is buffered at IC36 pin 10 to produce signal "SOFTWARE SYNC" which is used to initiate processing.

A.4.5.2 Interrupt request generation

The interrupt request signal "ATTN" is generated at IC12 pin 9 and is buffered out to the DR11-B interface at IC34 pins 15 and 5.

Signal "FNCT1" goes high when processing is enabled which makes IC12 pin 13 high, removing the clear from the interrupt latch, and enabling interrupts to be subsequently generated by the processor.

An interrupt request can be generated from two different sources depending on the stage which the processor has reached in a scan.

At the beginning of a "REAL SCAN" period signal "EED2" is generated to enable the edge detector. This signal is also input to IC12 pin 11, the clock input of the interrupt latch. The rising edge of "EED2" clocks pin 9 of IC12 high, which makes "ATTN" go high generating an interrupt request. When the computer recognises the interrupt it pulses "FNCT1" low, which clears the interrupt at IC12 pin 13.

This interrupt is intended as a request to the computer to step the table to the next position.

During the "DUMMY SCAN", when all the processed data which were produced in the "REAL SCAN" have been transmitted to the computer, signal "ENDOP" occurs indicating the end of transmission. This signal is inverted at IC24 pin 10 and appears at IC12 pin 10, the preset input of the interrupt latch. When "ENDOP" goes high, pin 9 of IC12 is set high which generates an interrupt request. "FNCT1" is again pulsed low when the interrupt is acknowledged to clear the request.

This interrupt is intended to inform the computer that new data have been sent.

A.4.5.3 D.M.A. control signal generation

Two control signals are generated to control D.M.A. data transfers: "SINGLE CYCLE" which holds the computer bus for use by the processor during a data transfer, and "CYCLE REQUEST" which asks the DR11-B to transfer a data word.

When signal "FNCT1" is high it enables D.M.A. requests to be made, by making IC9 pin 2 high and removing the clear from IC12 pin 1, the clear input of the D.M.A. latch.

At the start of the dummy scan, signal "DSCAN" goes high. This is input to IC12 pin 3 and clocks the D.M.A. latch so that IC12 pin 5 goes high and IC12 pin 6 goes low. Pin 5 partially enables the generation of "CYCLE REQUEST" signals by setting IC9 pin 13 high, and is also output as signal "COMPEN", which goes to enable the "EDGE POSITION" store transmit logic. Pin 6 is output to the DR11-B interface through buffer IC34 as "SINGLE CYCLE".

When data transmission is complete, signal "ENDOP" goes high, which makes IC12 pin 1 low, clearing the D.M.A. latch. This terminates signal "SINGLE CYCLE", sets "COMPEN" low, disabling the store transmit logic, and disables generation of further "CYCLE REQUESTS", by setting IC9 pin 13 low. "ENDOP" is also removed when "COMPEN" goes low, removing the clear from the D.M.A. latch.

During the period when D.M.A. latch is set, data can be transmitted to the computer.

In the case when no edges are detected in a scan and there are no data to transmit, signal "ENDOP" goes high immediately the D.M.A. latch is set, so that no data can be transmitted.

In the case where there are data to transmit, "CYCLE REQUEST" pulses are generated while the D.M.A. latch is set.

At the end of a "REAL SCAN" signal "AEEDW" goes high making IC9 pin 12 high. When IC9 pin 13 goes high at the start of a "DUMMY SCAN" the clear is removed from IC23 pin 1 (end cycle is low at this time). The 74LS161 counter now counts "CK" clock pulses. As the count passes through values 4, 5, 6, 7, pin 12 is high. When the count reaches value 8 pin 12 goes low again and pin 11 goes high. Pin 11 is inverted at IC9 pin 8 and disables any further counting by setting IC23 pins 7 and 10 low. The counter sticks in this state. In this way a pulse is generated at IC23 pin 12, which is output through IC34 as signal "SINGLE CYCLE" to the DR11-B.

The DR11-B now transfers the data word which is currently output from the edge position store to the computer memory. On completion of transfer the DR11-B pulses signal "END CYCLE" high. This pulses IC36 pin 1 producing a clear at IC23 pin 1.

The counter now counts up to 8 again, producing another "CYCLE REQUEST" pulse. This again is followed by an "END CYCLE", the process repeating until "ENDOP" goes high, clearing the D.M.A. latch and disabling further "CYCLE REQUESTS".

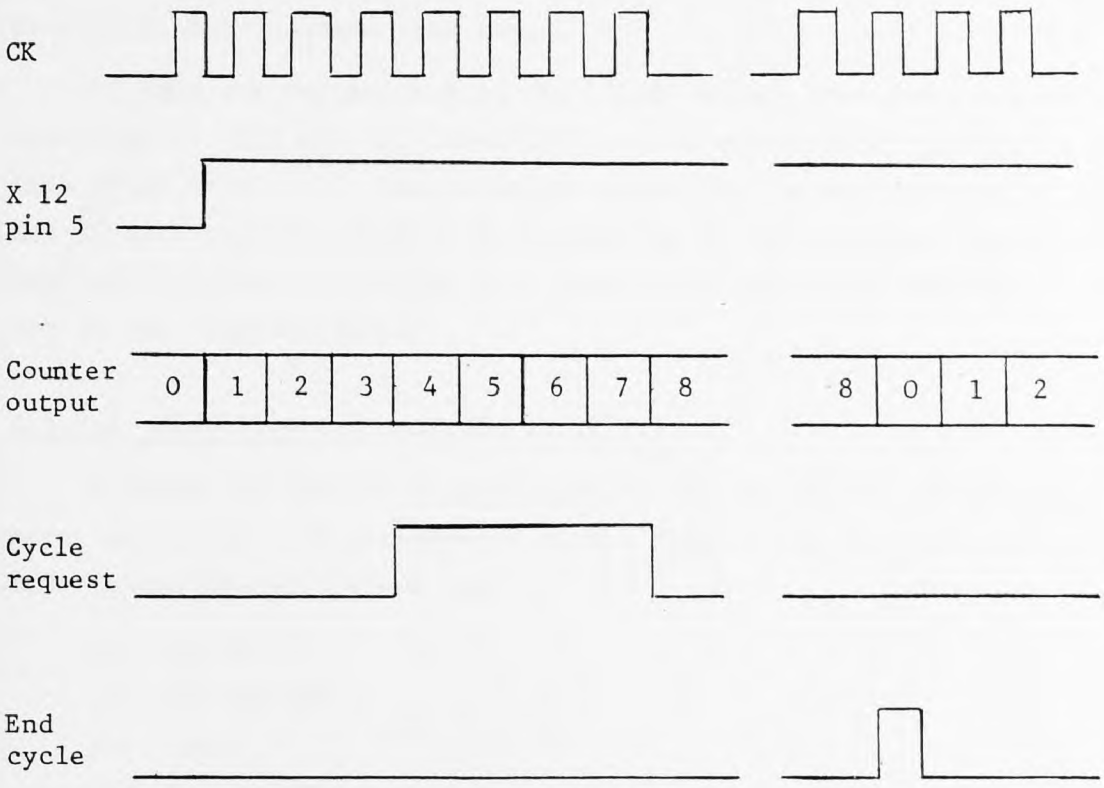


Figure A.10: Generating cycle request

A.4.5.4 Generation of "DSTAT" control word

The "DSTAT" word is generated in company with the interrupt request signal "ATTN" to indicate to the computer the nature of the request.

At the start of the "REAL SCAN" signal "RSCAN" goes low during the ΦXB pulse. This sets the DSTAT word to zero, by clearing one element of IC35, making IC35 pin 5 low and presetting the other element of IC35, making IC35 pin 8 low. DSTATC is tied low permanently.

When the interrupt request at the beginning of the REAL SCAN occurs,

DSTAT = 0, which means that the request is for the computer to step the table to the next position.

At the end of a "REAL SCAN" when signal "ESCAN" goes low it sets IC35 pin 8 high, by applying a clear at IC35 pin 13. This makes DSTAT = 1.

If, at the beginning of the "DUMMY SCAN", no data are transferred to the computer (because no edges were found) the DSTAT word is still value 1 when the second interrupt request occurs (that occurring at the end of data transfer). This informs the computer that a scan has been completed, but no edges were found.

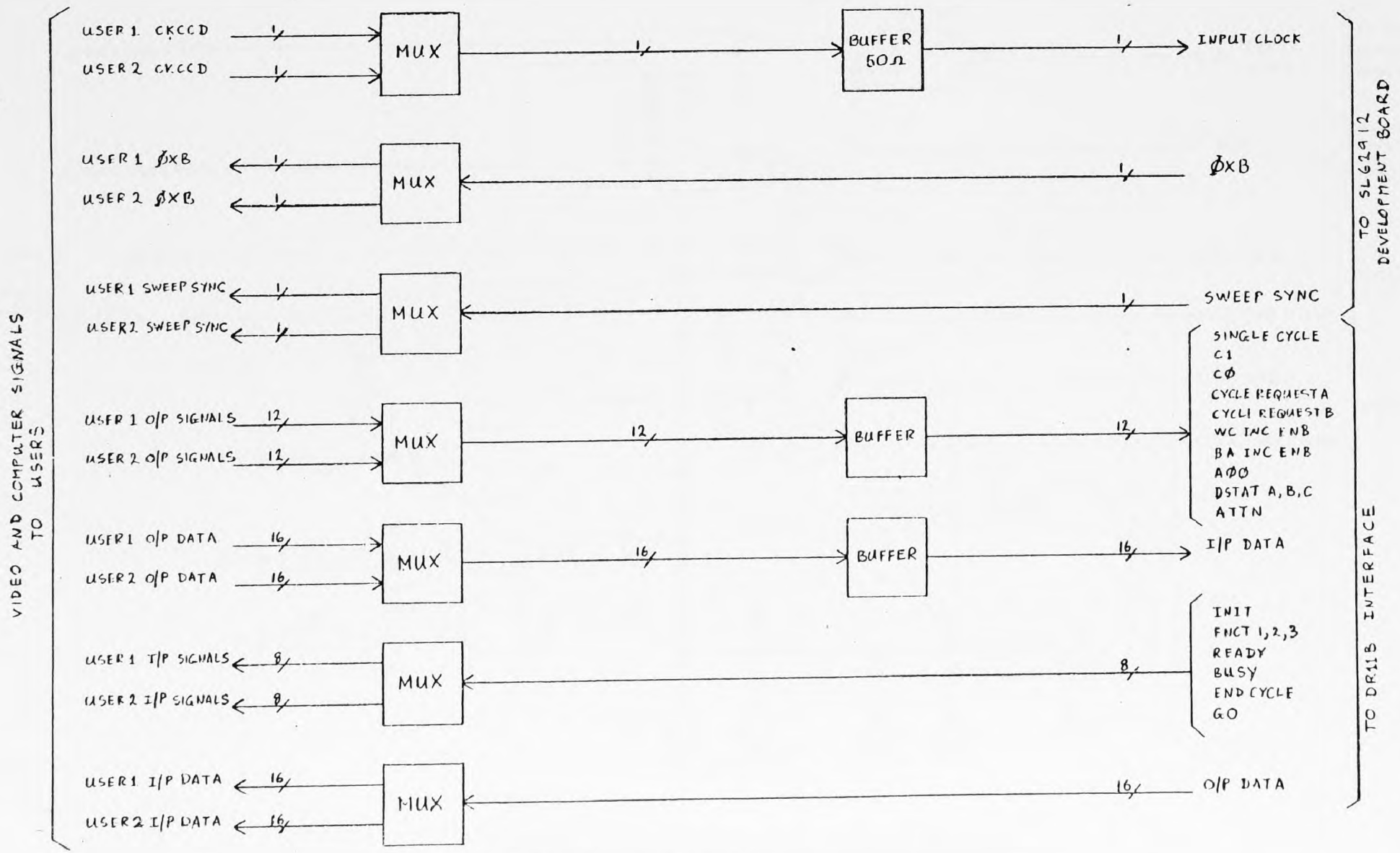
If data are transmitted in the "DUMMY SCAN", the cycle requests generated at IC23 pin 12, clock IC35 pin 3, making pin 5 go high. At this point DSTAT = 3. Now, when the interrupt request is made at the end of data transfer DSTAT = 3, indicating to the computer that a scan has been completed and edges were found, the data corresponding to which are in the computer memory.

A.4.5.5 Miscellaneous controls to DR11-B

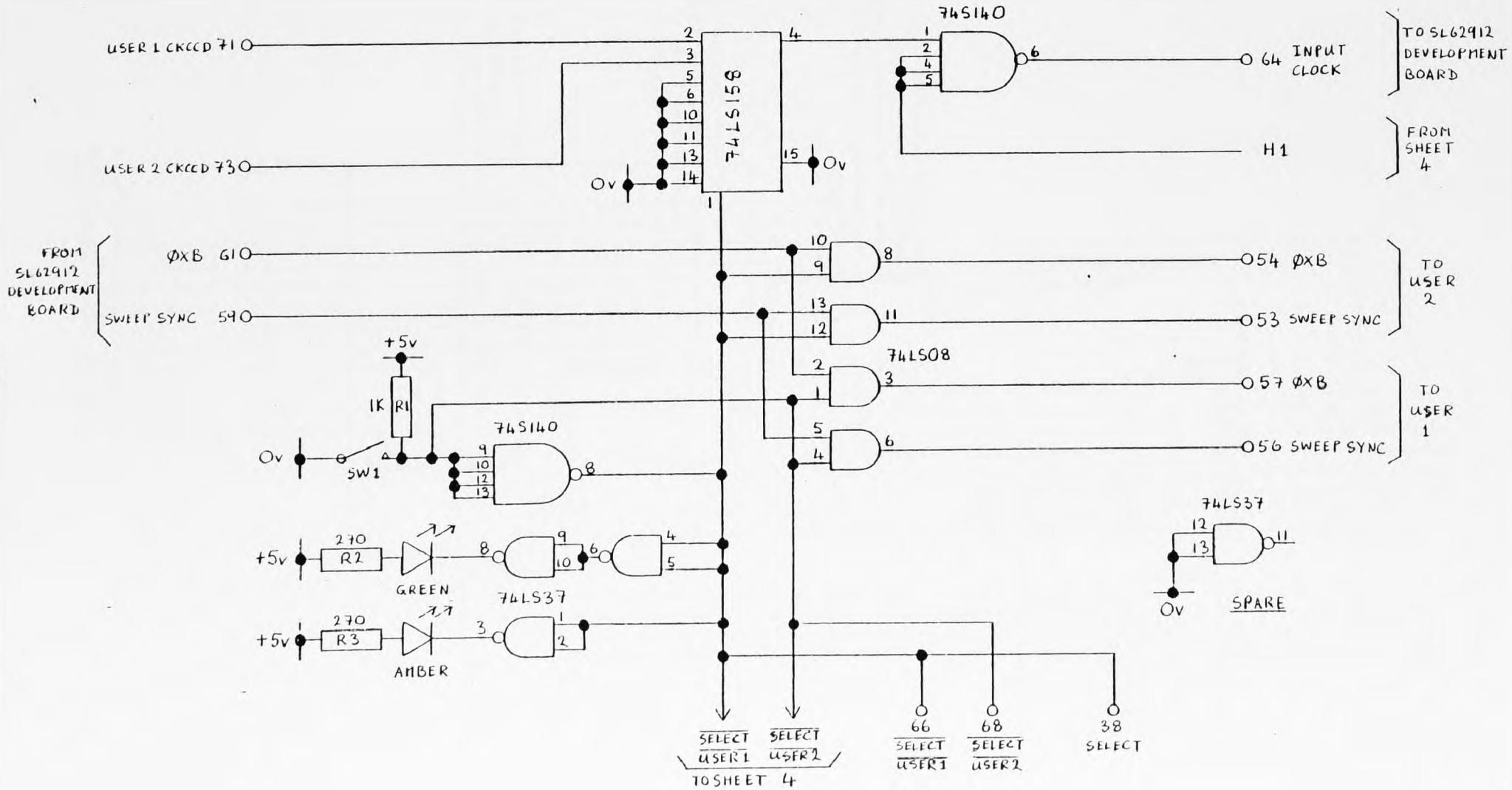
A number of control signals required by the DR11-B are produced here, which are tied permanently either high or low and buffered through IC34 to the DR11-B. These are

(a) CO and C1
(b) WC INC ENB
(c) AOO
(d) BA INC ENB.

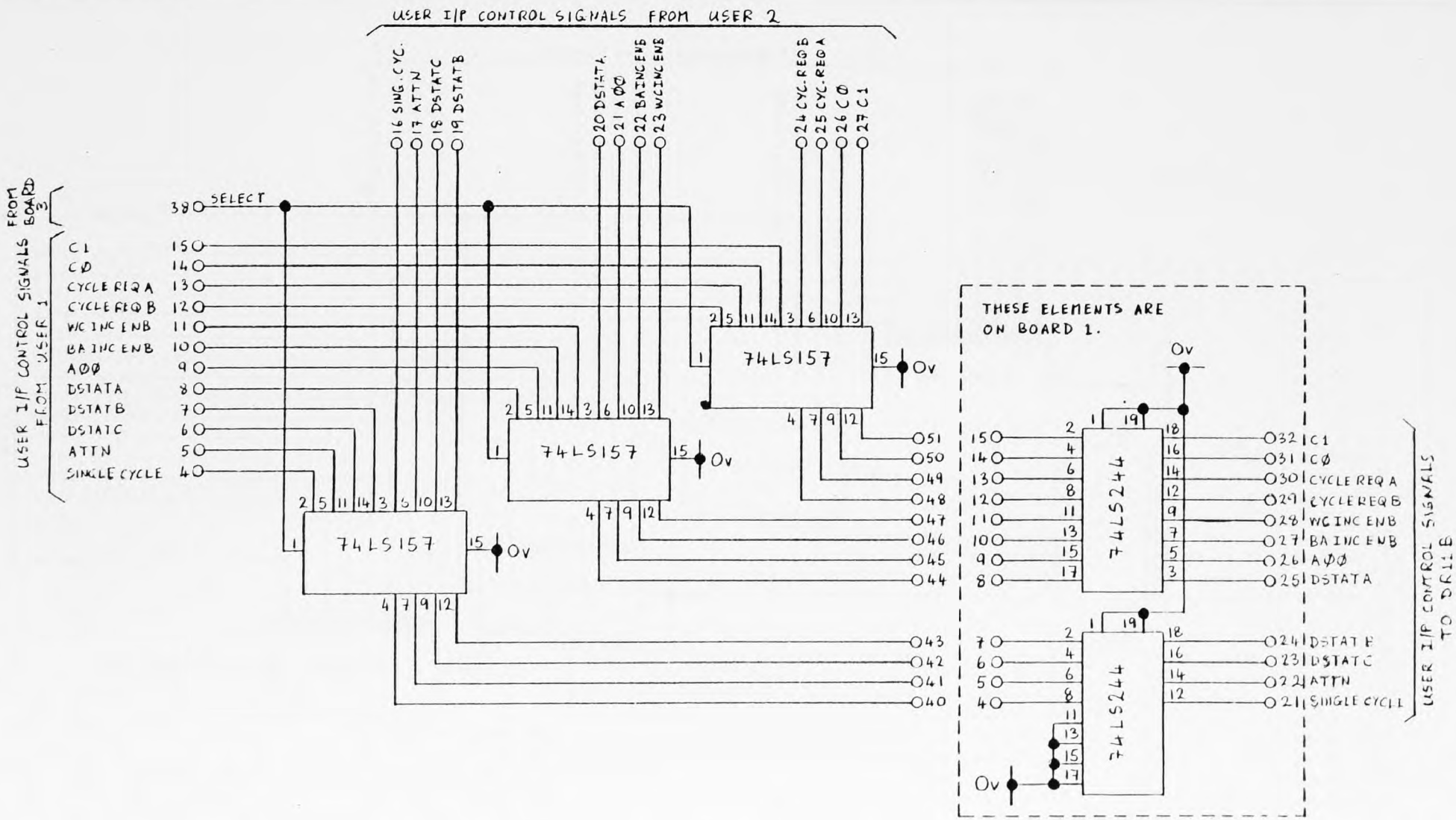
The function of these signals is detailed in section A.1.



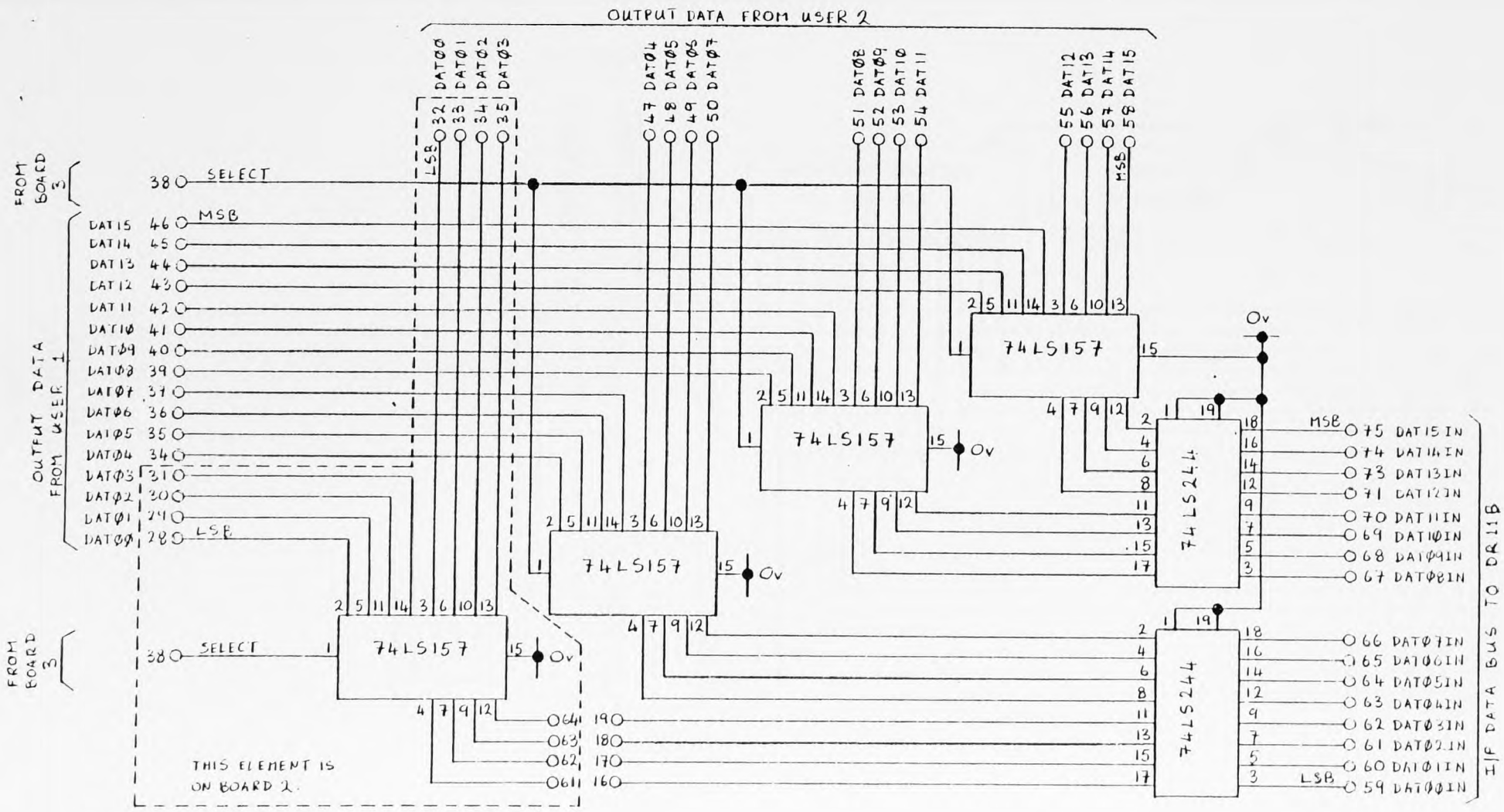
BLOCK DIAGRAM: MULTIPLEXING FUNCTION



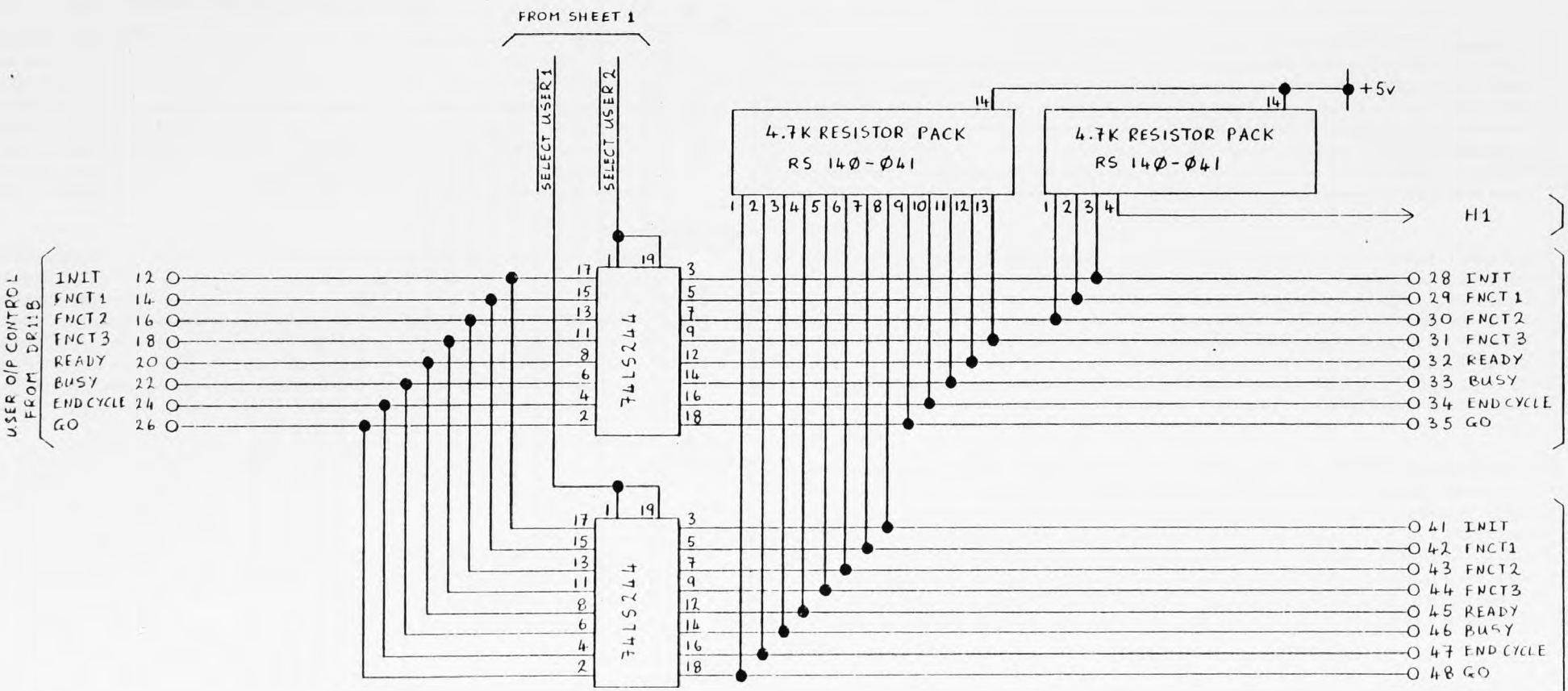
MULTIPLEX FUNCTION: SHEET 1, (PART OF BOARD 3)



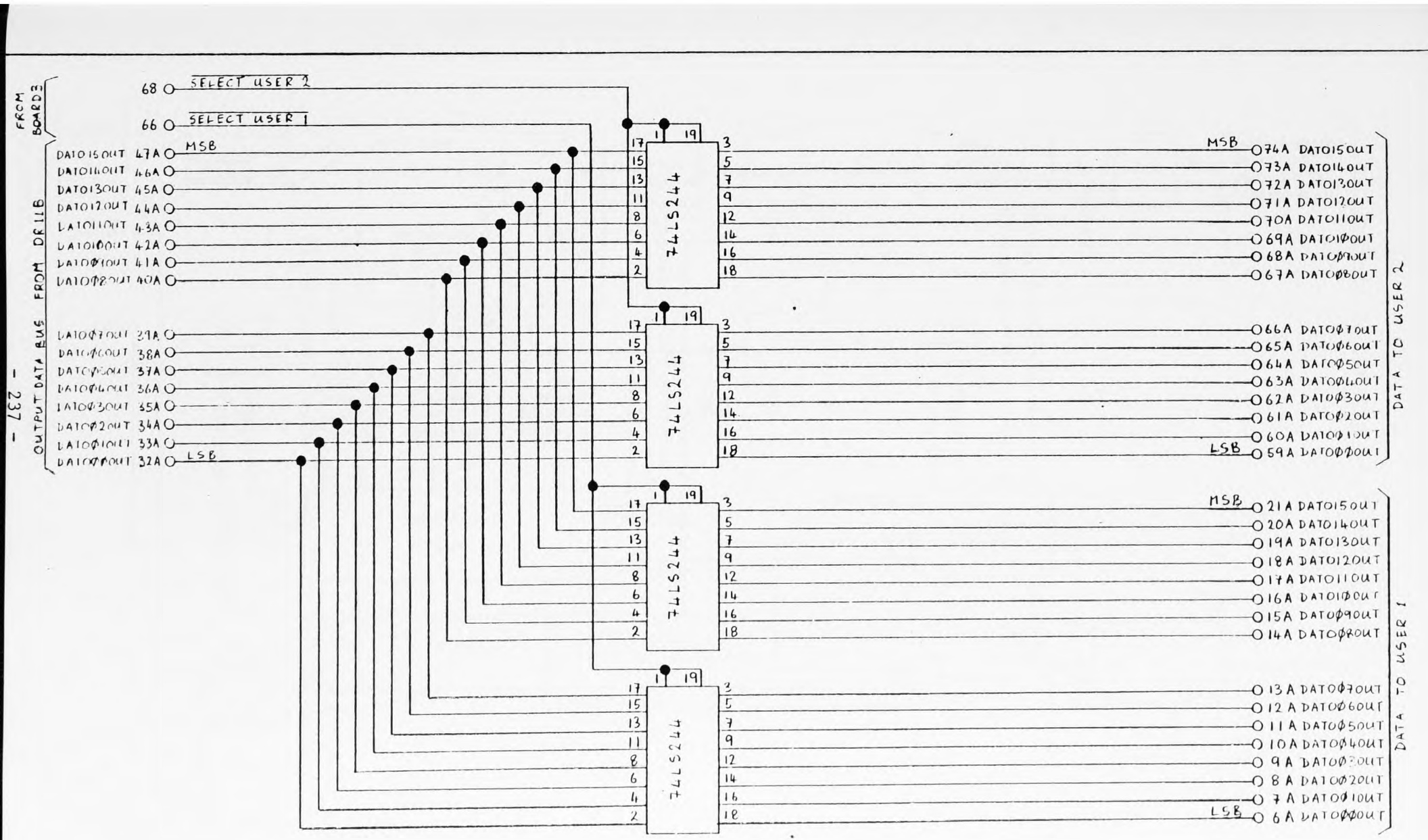
MULTIPLEX FUNCTION: SHEET 2, (PARTS OF BOARDS 1 AND 2)



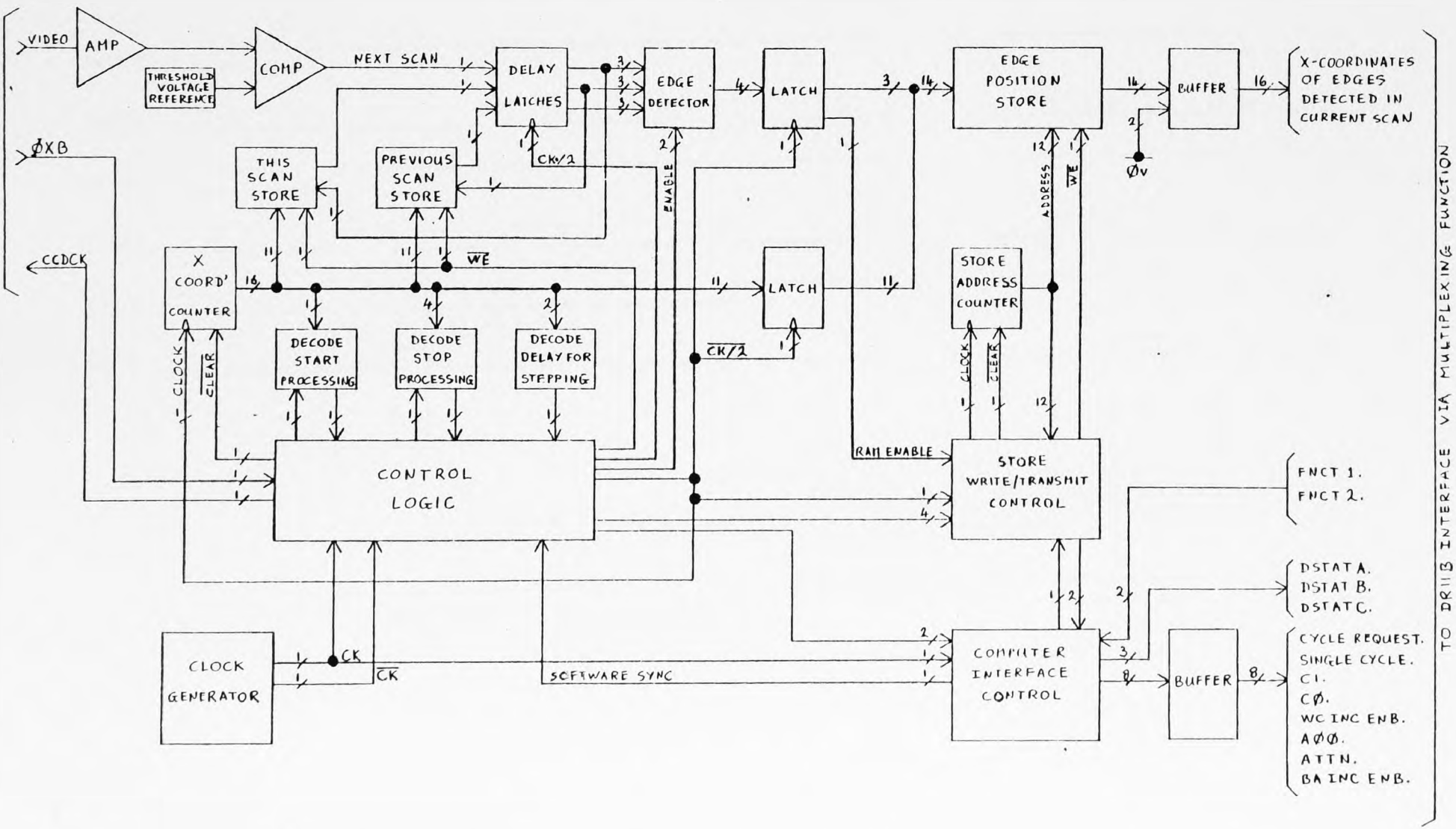
MULTIPLEX FUNCTION: SHEET 3 (PARTS OF BOARDS 1 AND 2)



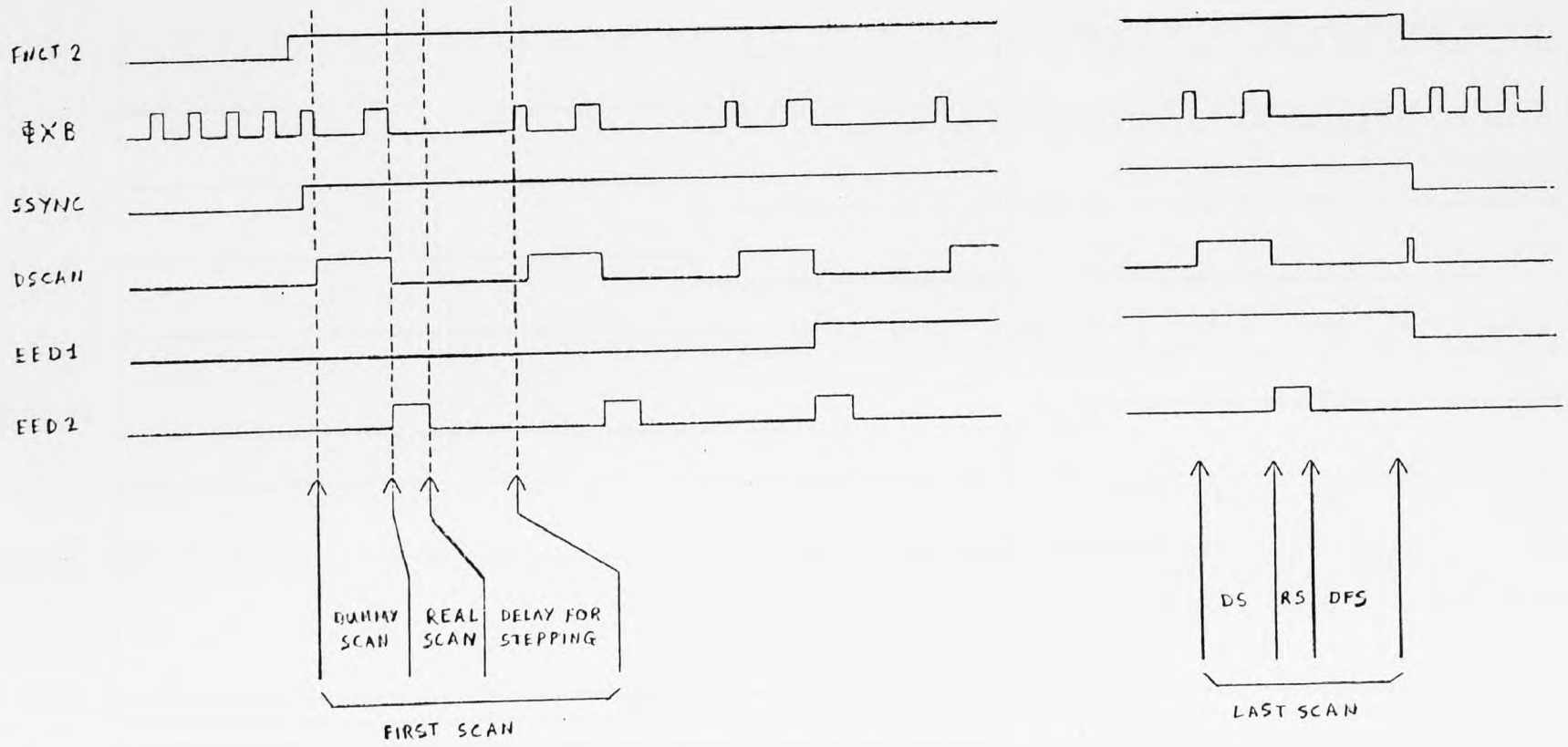
MULTIPLEX FUNCTION: SHEET 4, (PART OF BOARD 3)



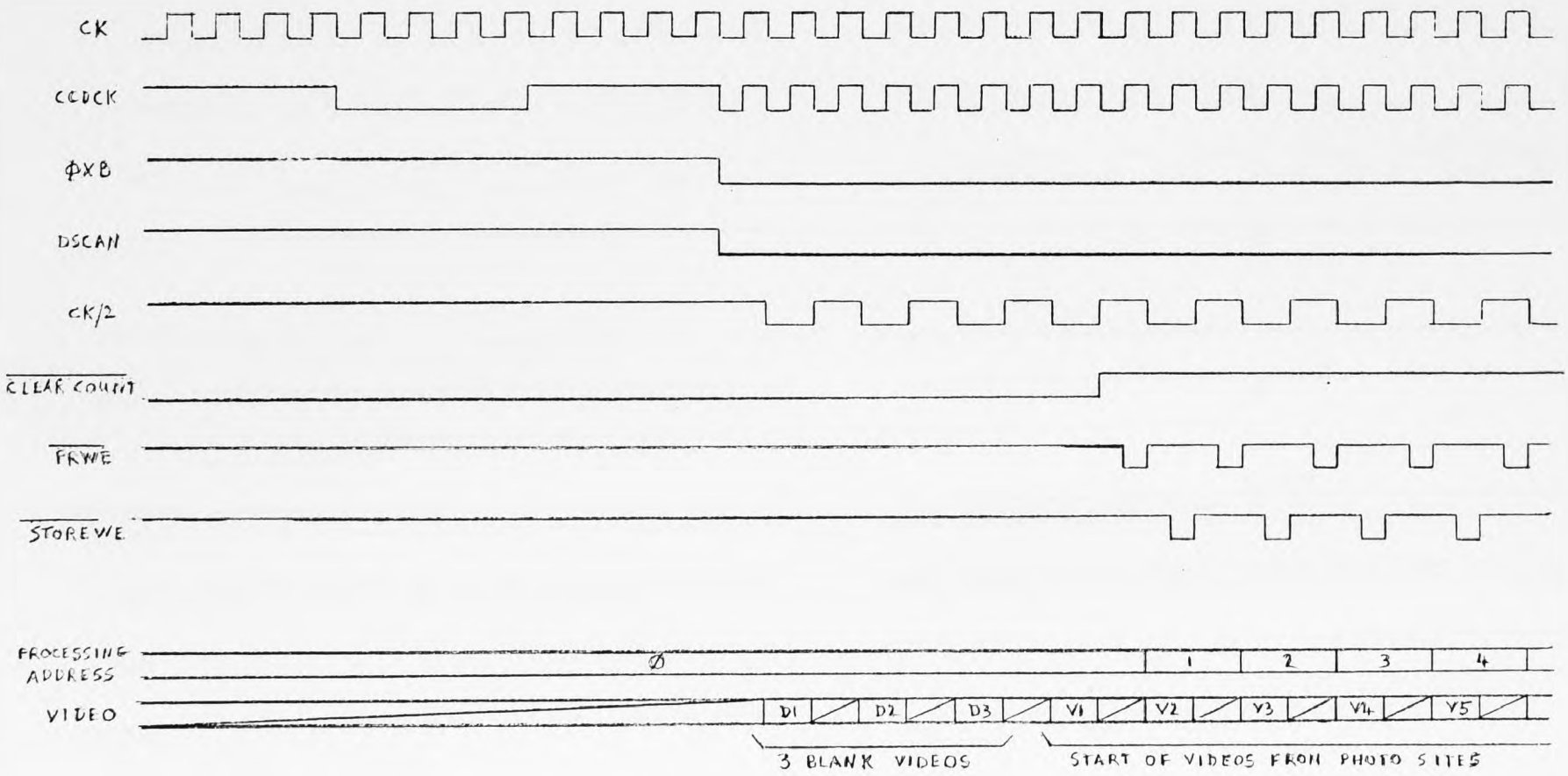
MULTIPLEX FUNCTION: SHEET 5 (PART OF BOARD 2)



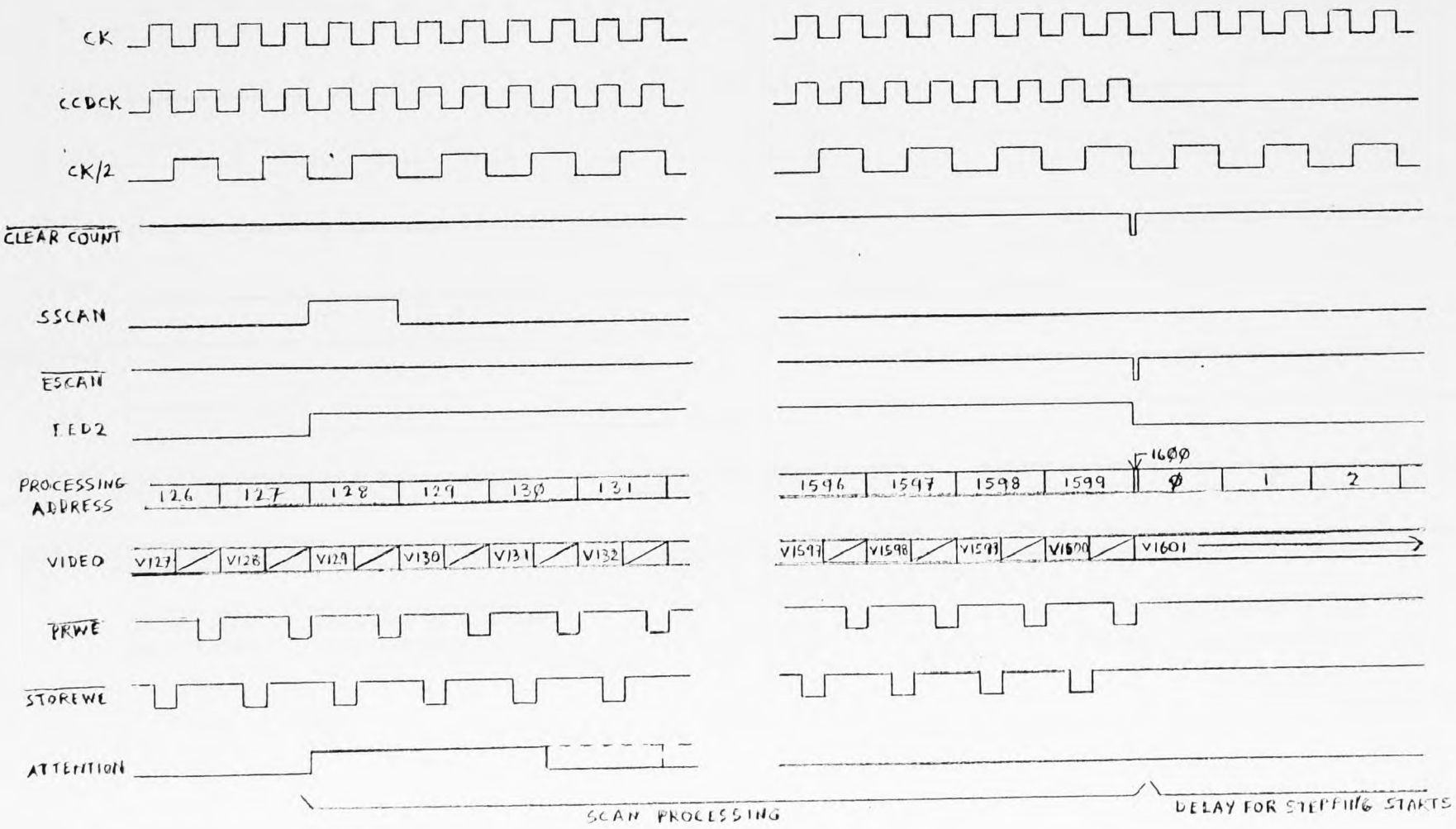
BLOCK DIAGRAM: SILHOUETTE IMAGE PREPROCESSOR



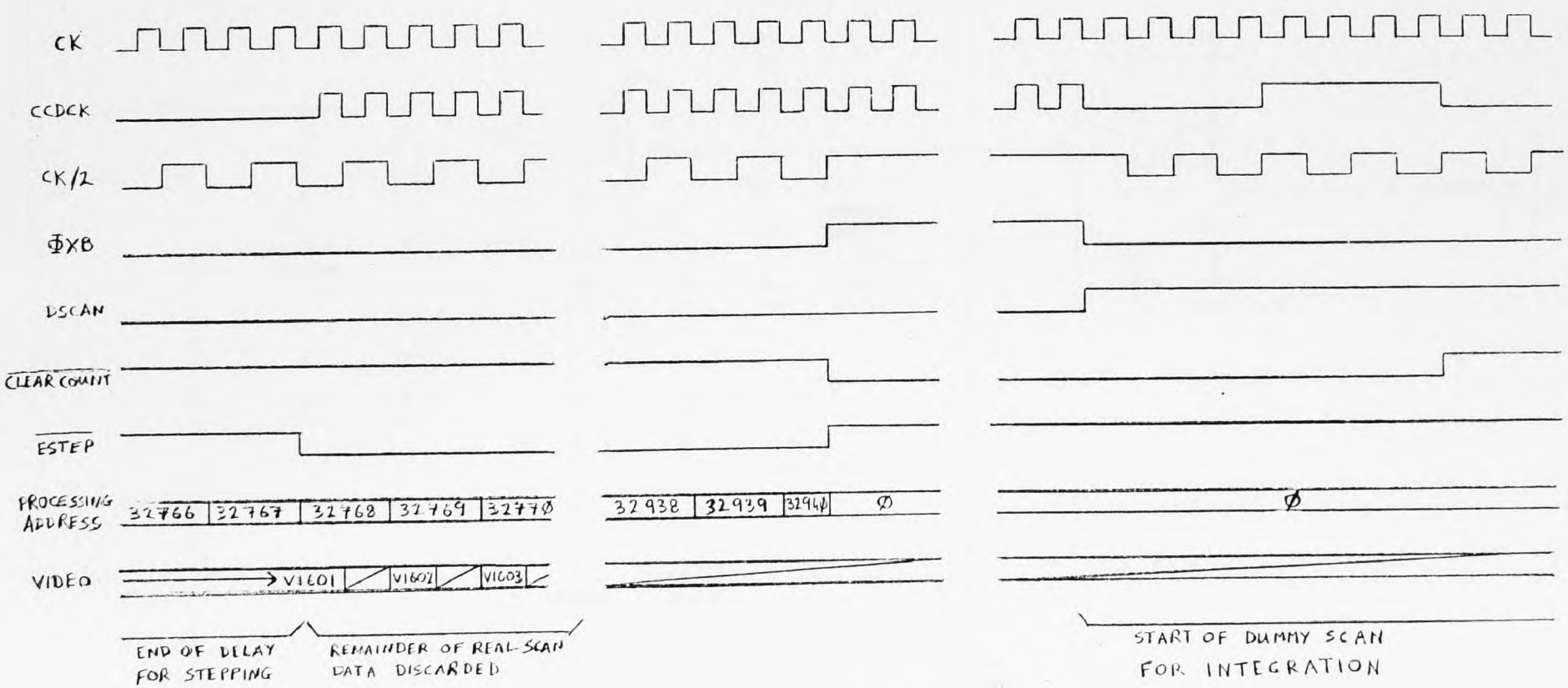
SILHOUETTE IMAGE PREPROCESSOR: FRAME TIMING (NOT TO SCALE)



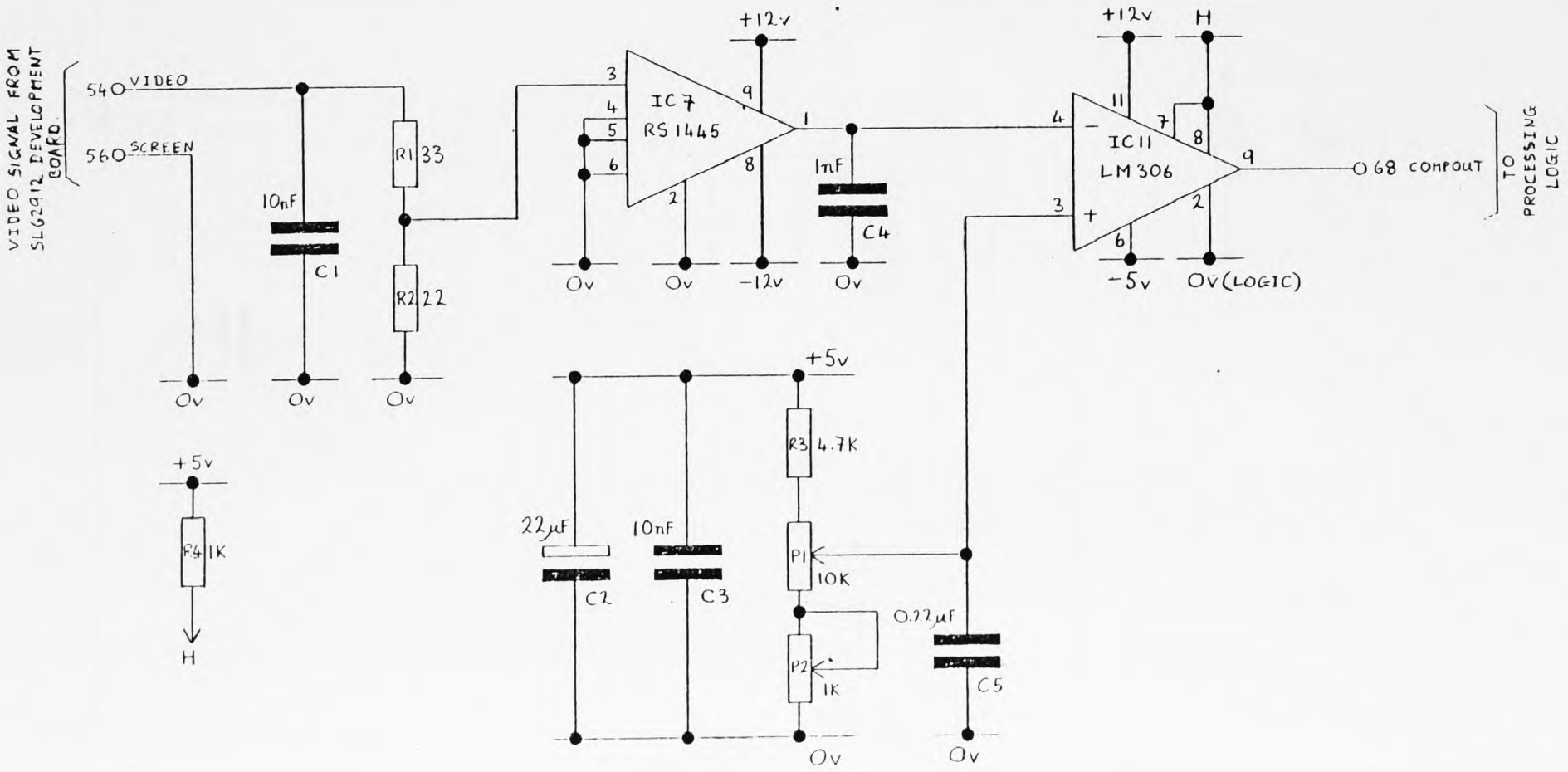
SILHOUETTE IMAGE PREPROCESSOR: START OF REAL SCAN TIMING



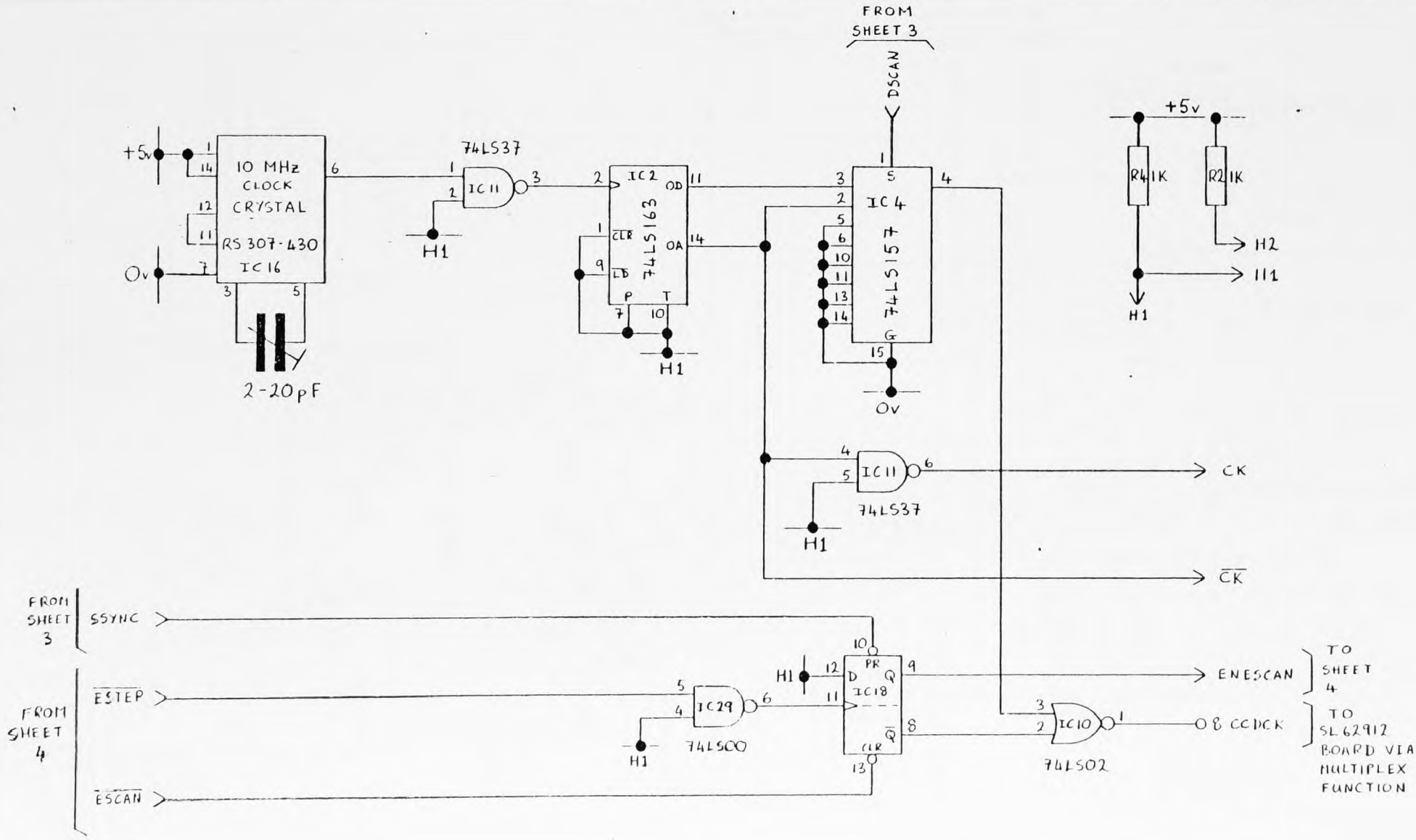
SILHOUETTE IMAGE PREPROCESSOR: START AND END OF PROCESSING TIMING



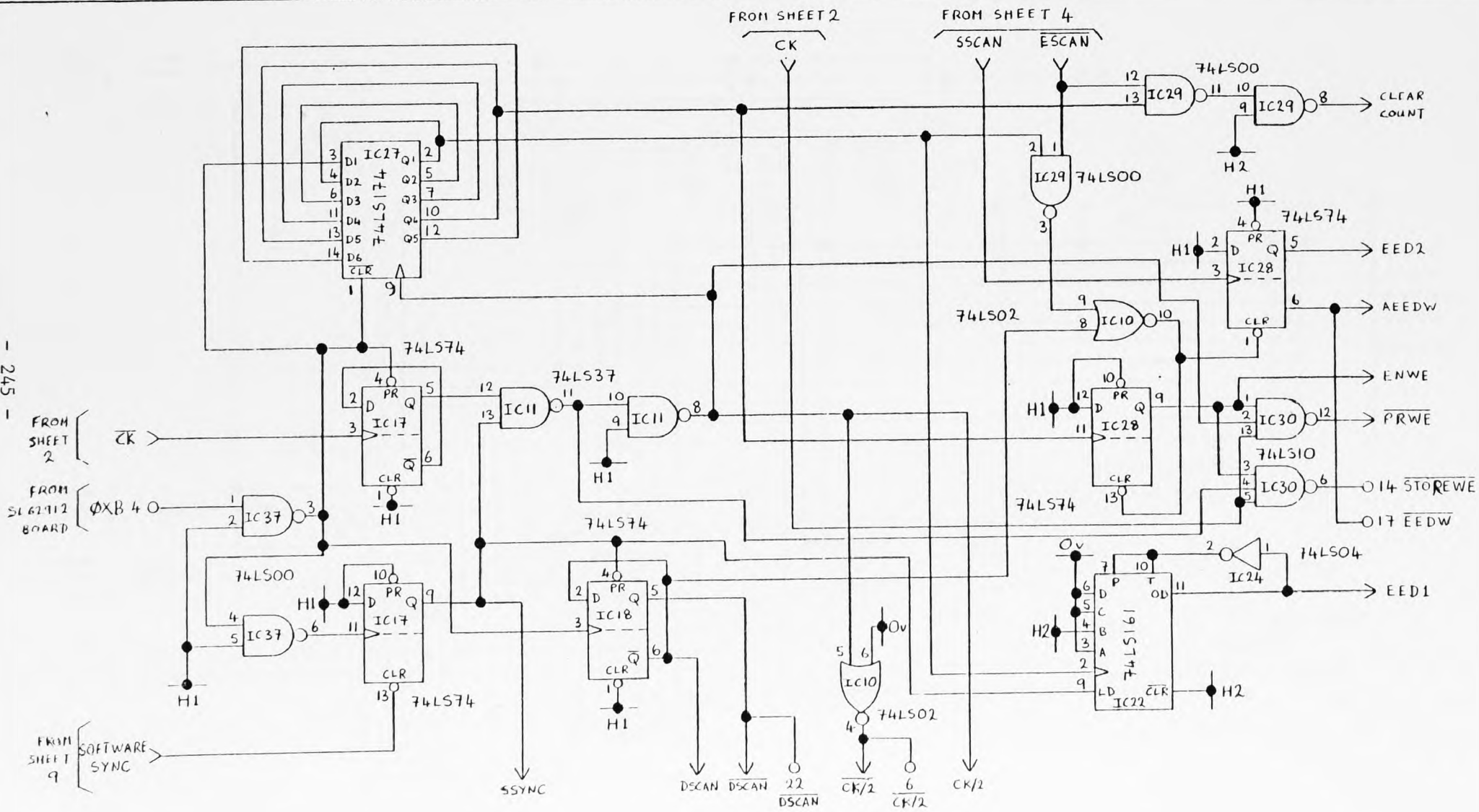
SILHOUETTE IMAGE PREPROCESSOR: END OF STEPPING - START OF DUMMY SCAN



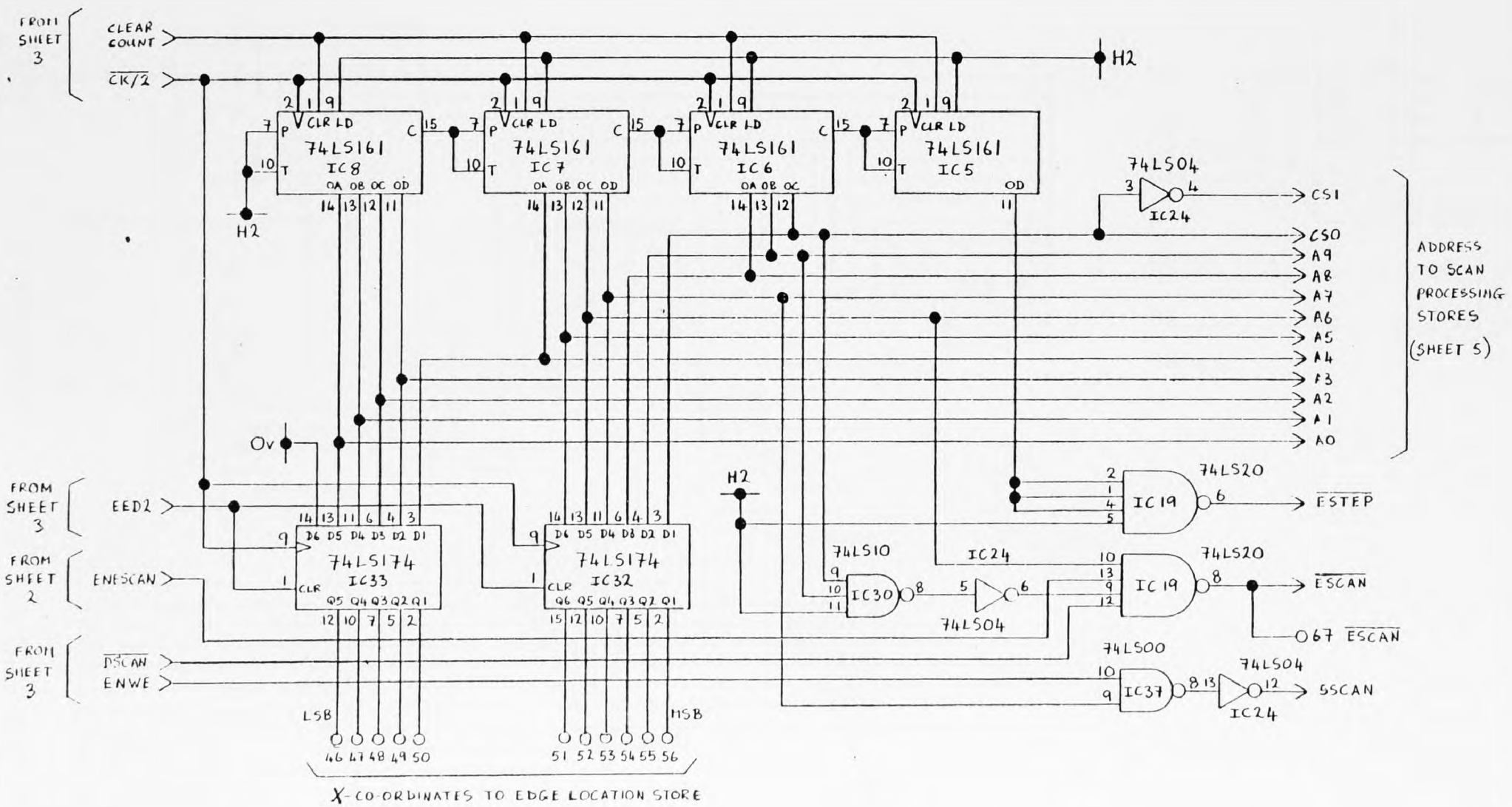
SILHOUETTE IMAGE PREPROCESSOR: SHEET 1 (PART OF BOARD 6)



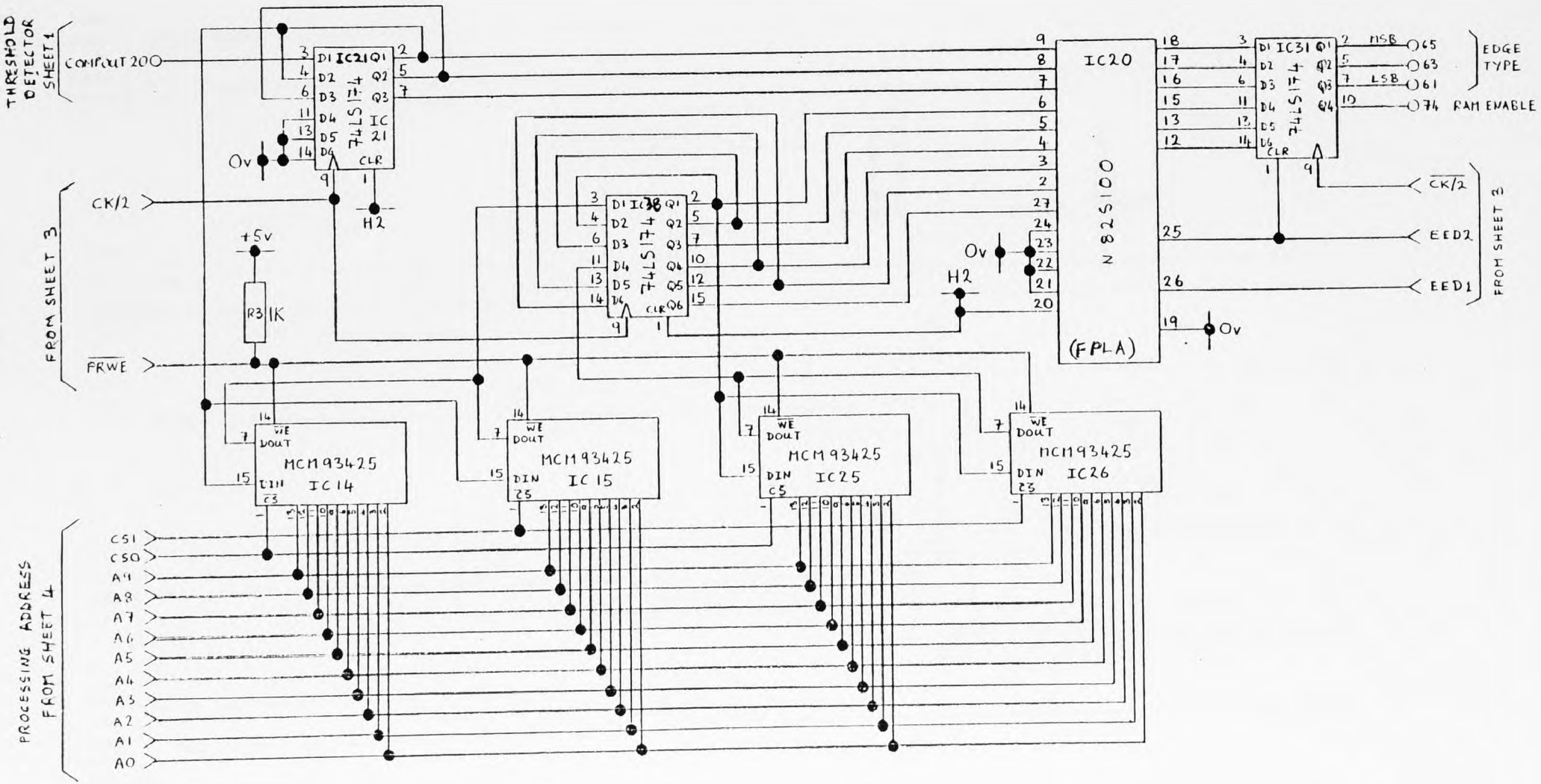
SILHOUETTE IMAGE PREPROCESSOR: SHEET 2 (PART OF BOARD 4)



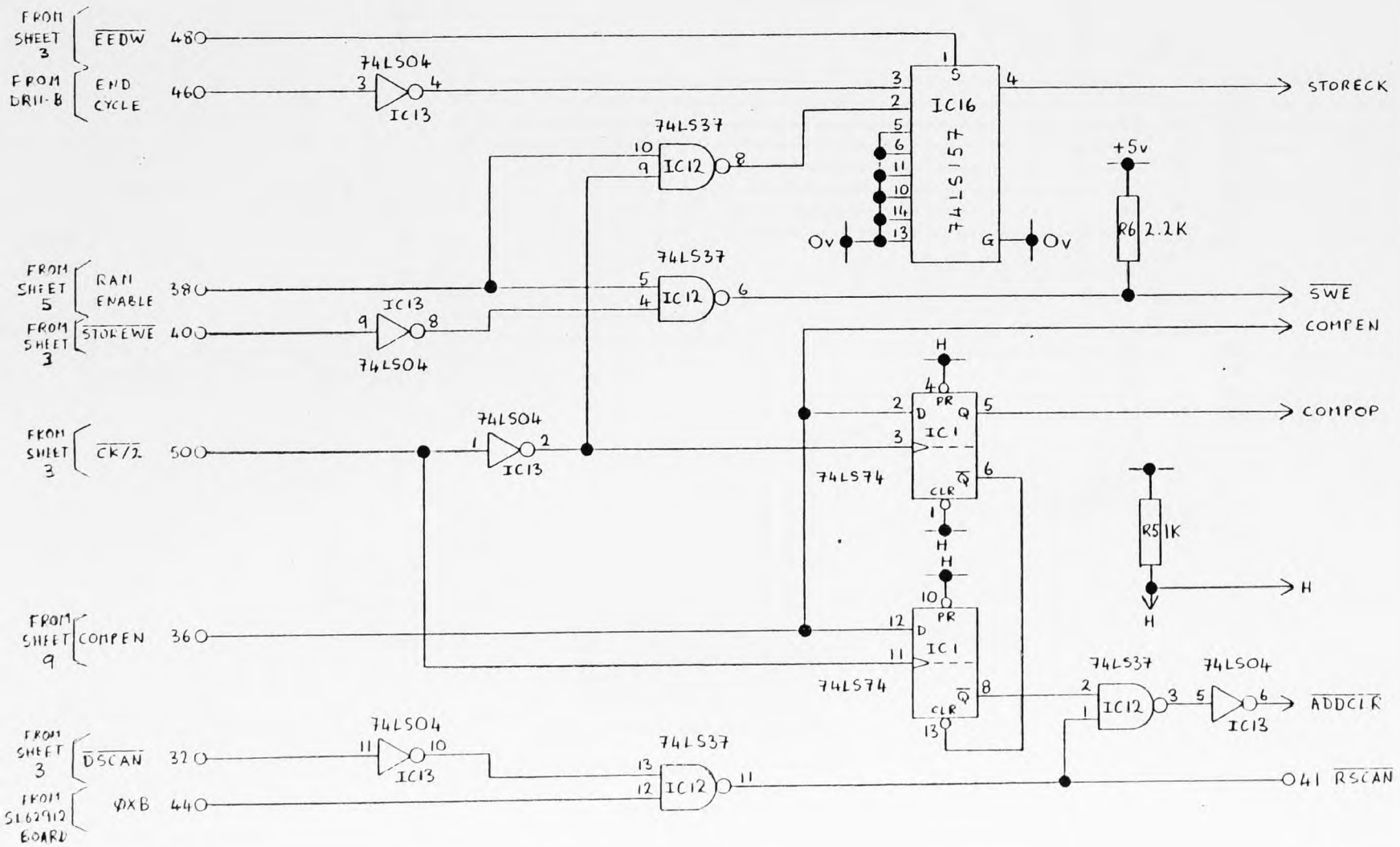
SILHOUETTE IMAGE PREPROCESSOR: SHEET 3 (PART OF BOARD 4)



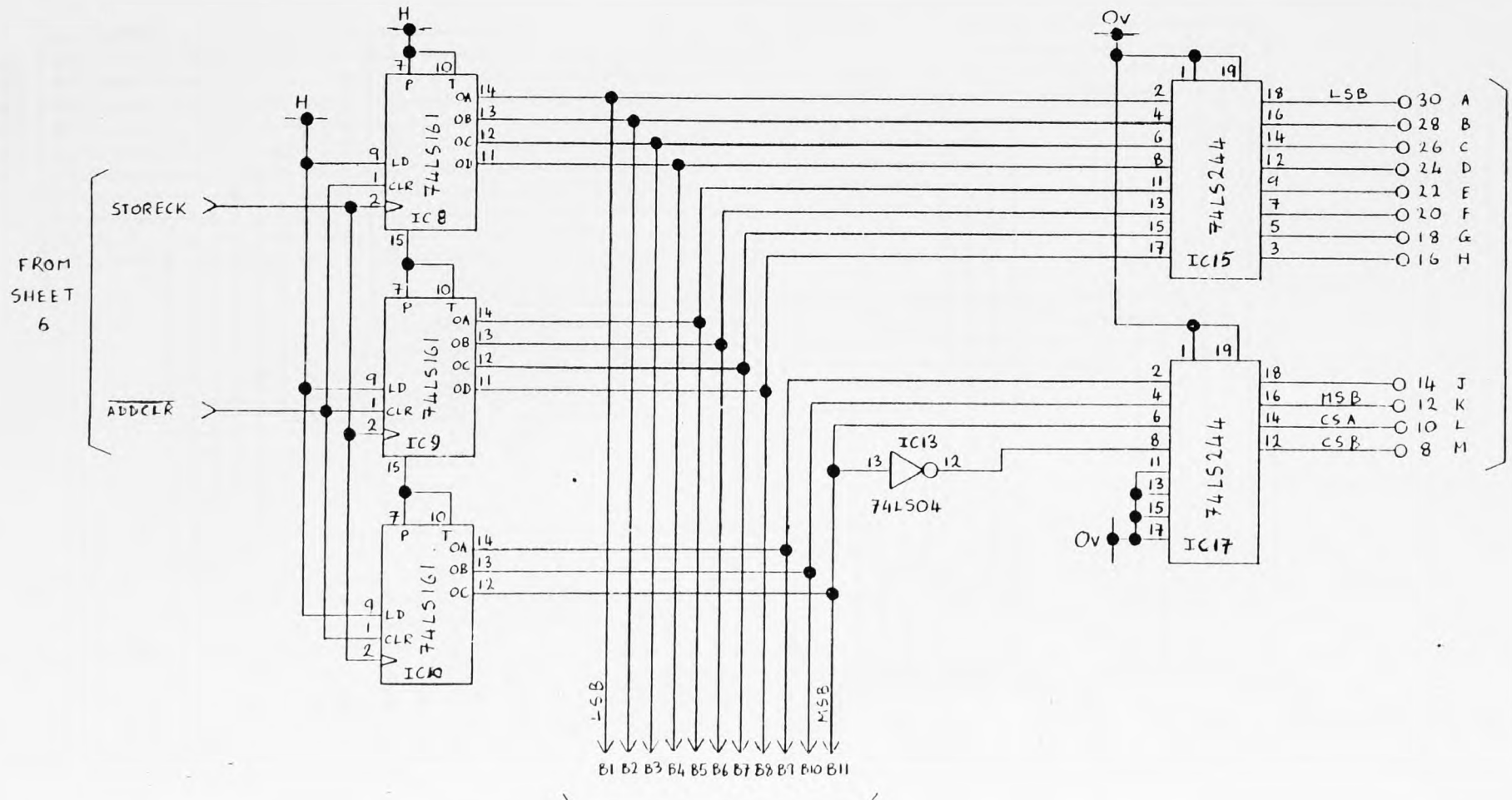
SILHOUETTE IMAGE PREPROCESSOR: SHEET 4 (PART OF BOARD 4)



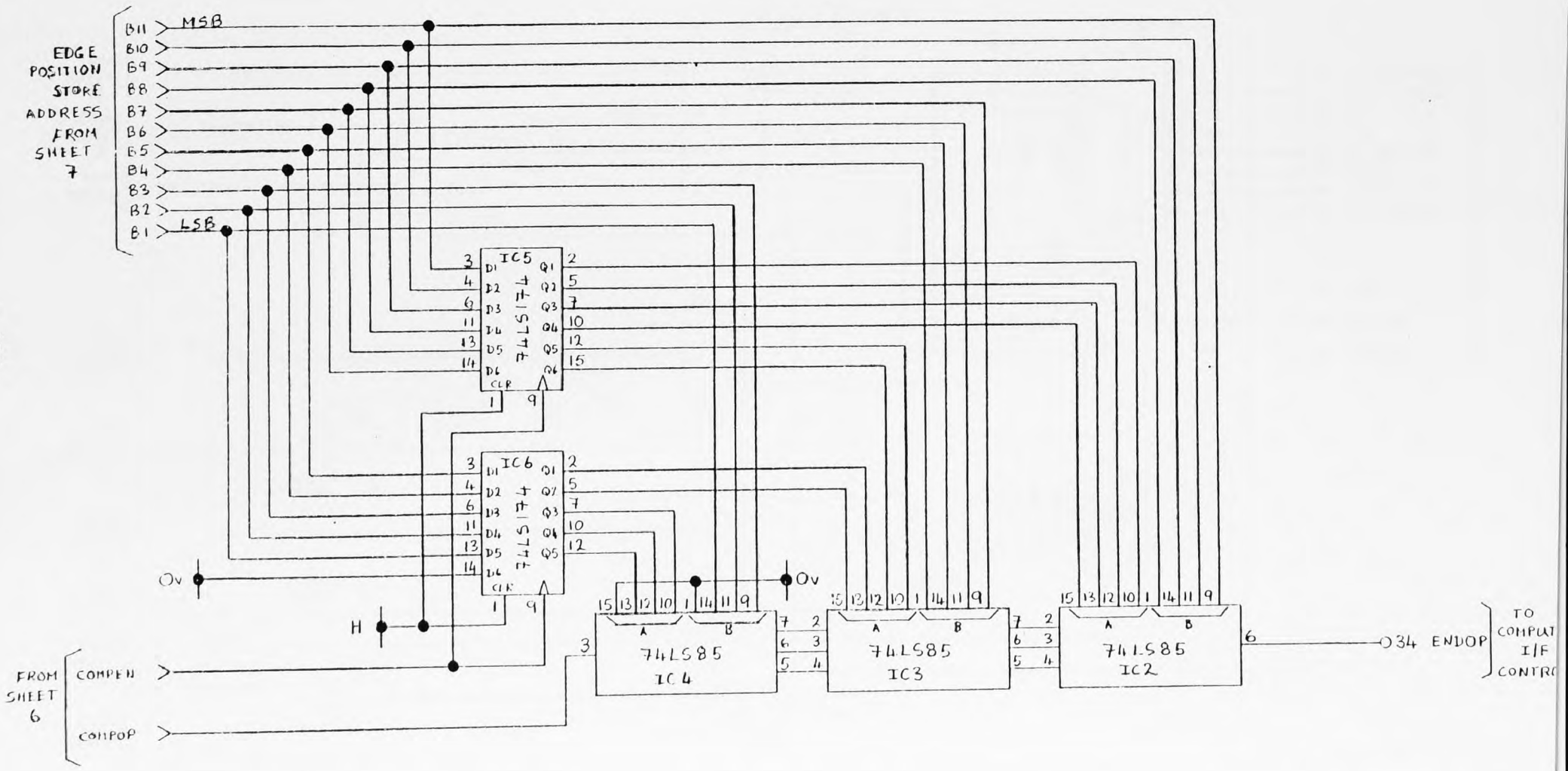
SILHOUETTE IMAGE PREPROCESSOR: SHEET 5 (PART OF BOARD 4)



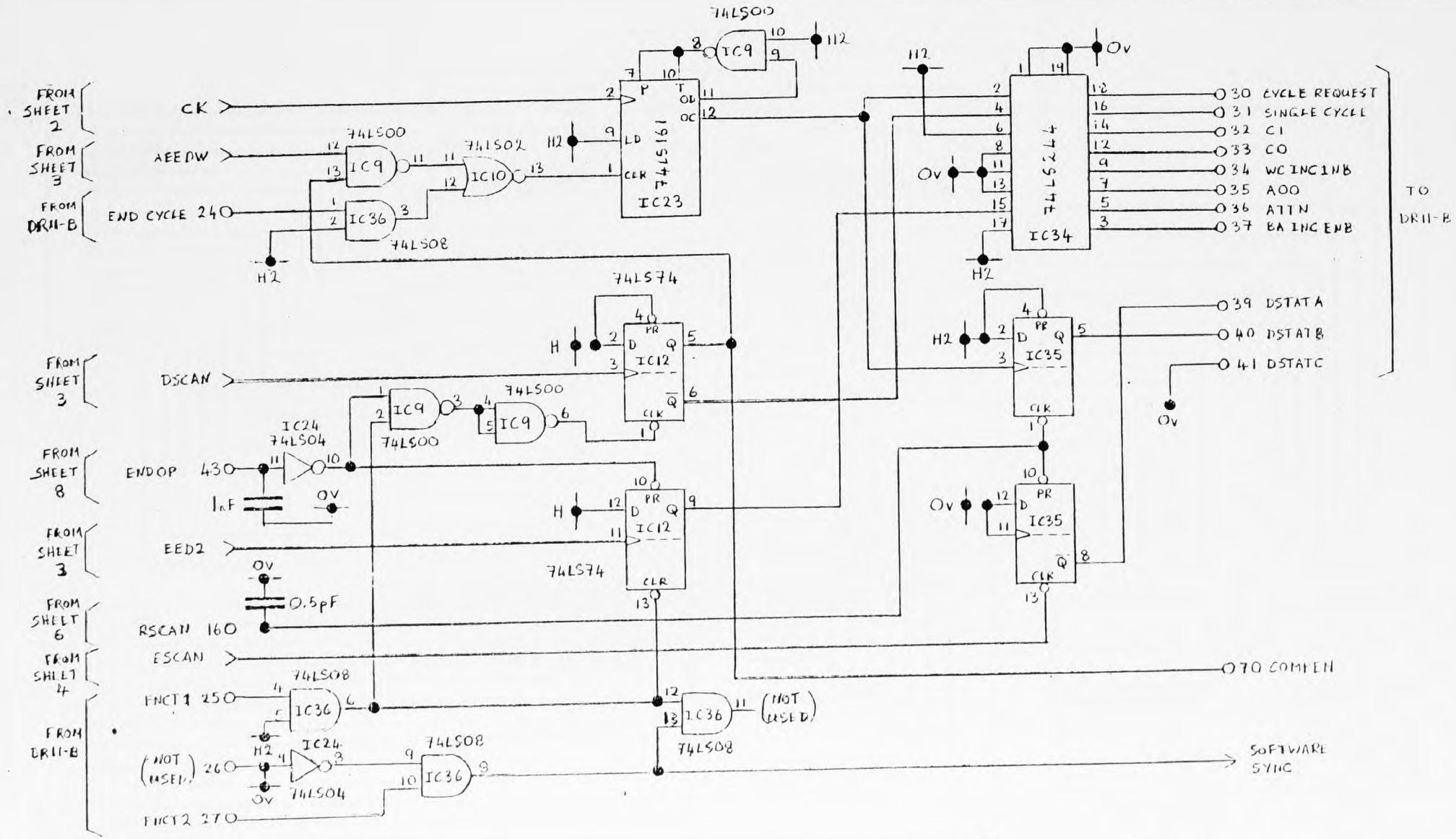
SILHOUETTE IMAGE PREPROCESSOR: SHEET 6 (PART OF BOARD 6)



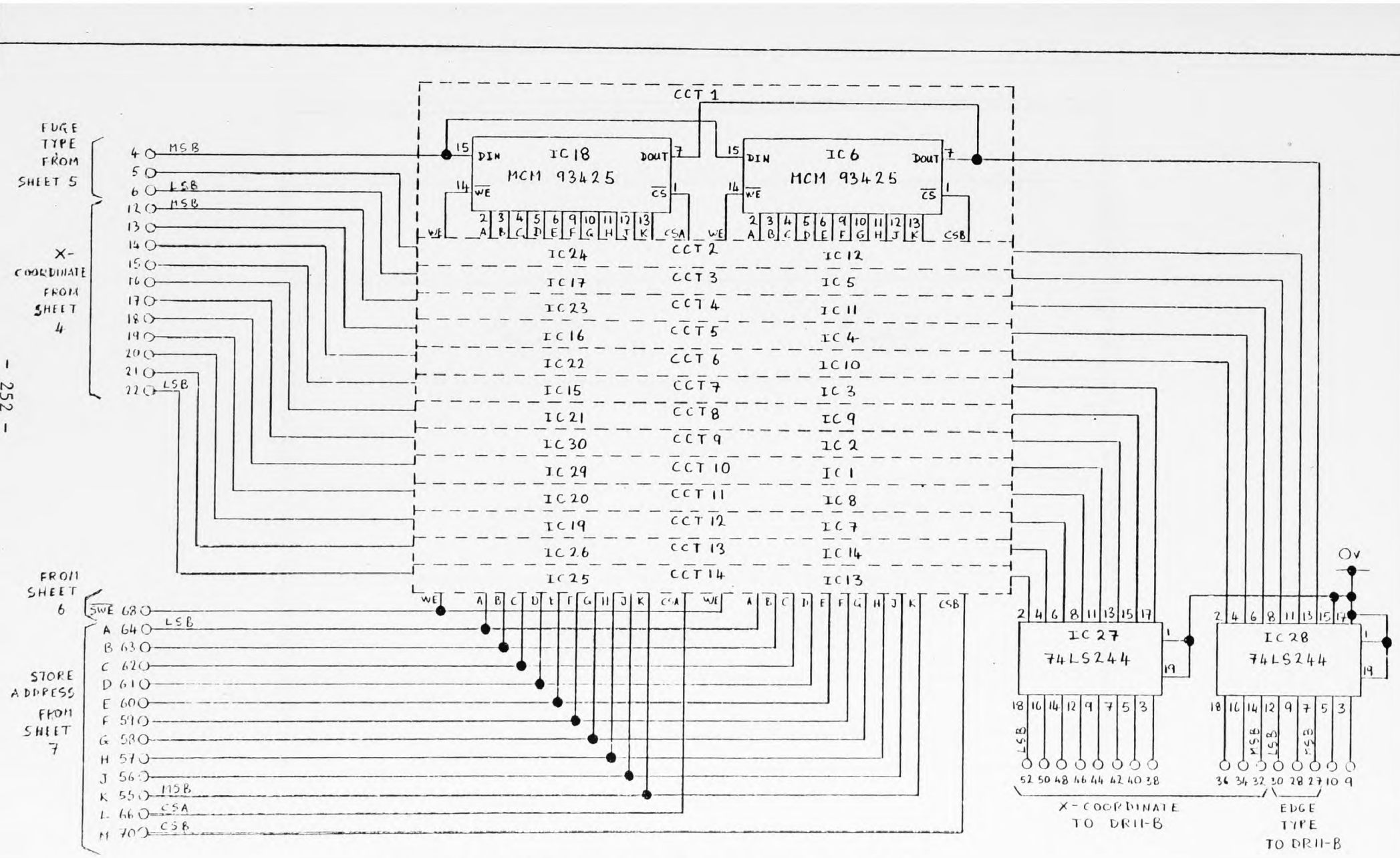
SILHOUETTE IMAGE PREPROCESSOR: SHEET 7 (PART OF BOARD 6)



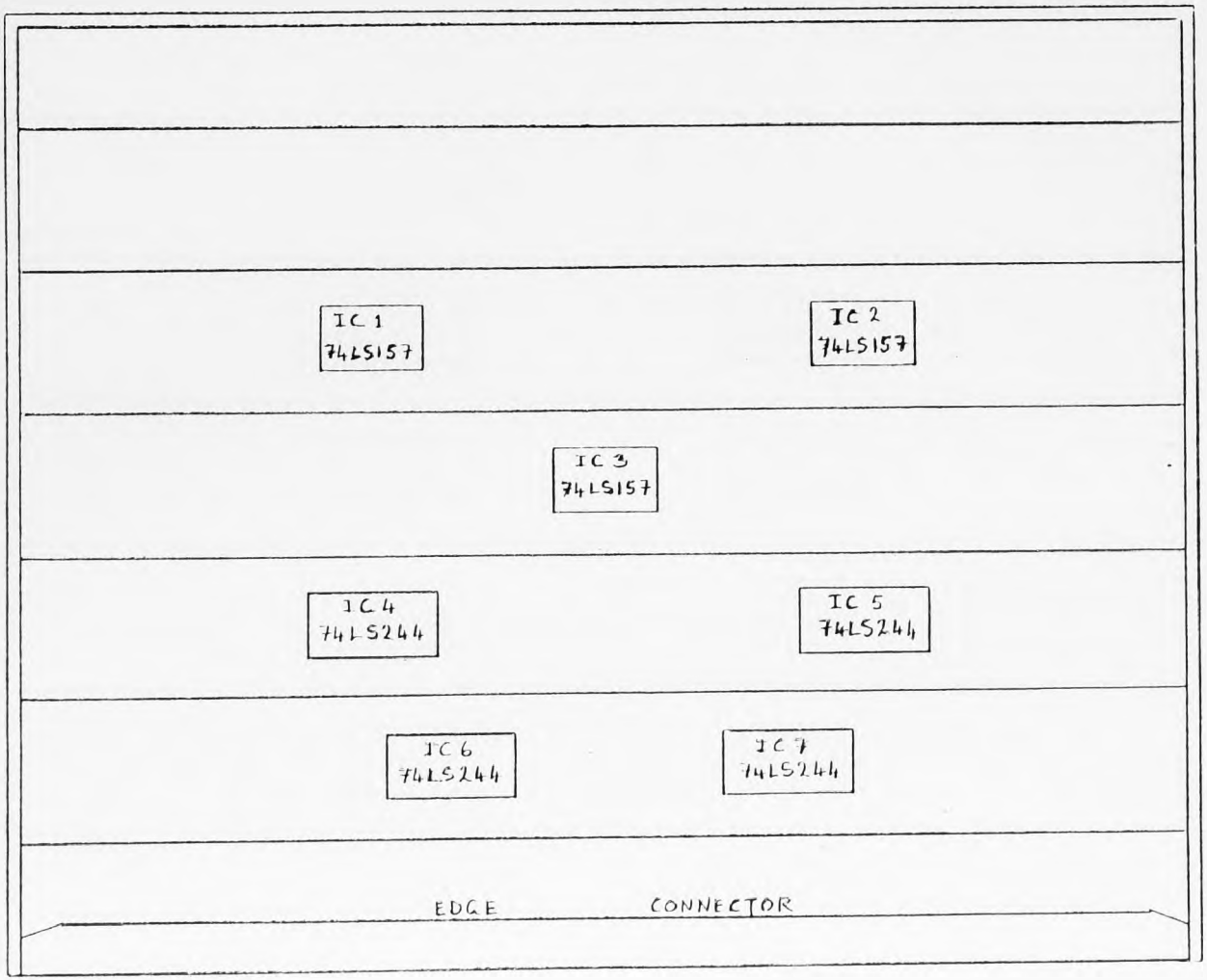
SILHOUETTE IMAGE PREPROCESSOR: SHEET 8 (PART OF BOARD 6)



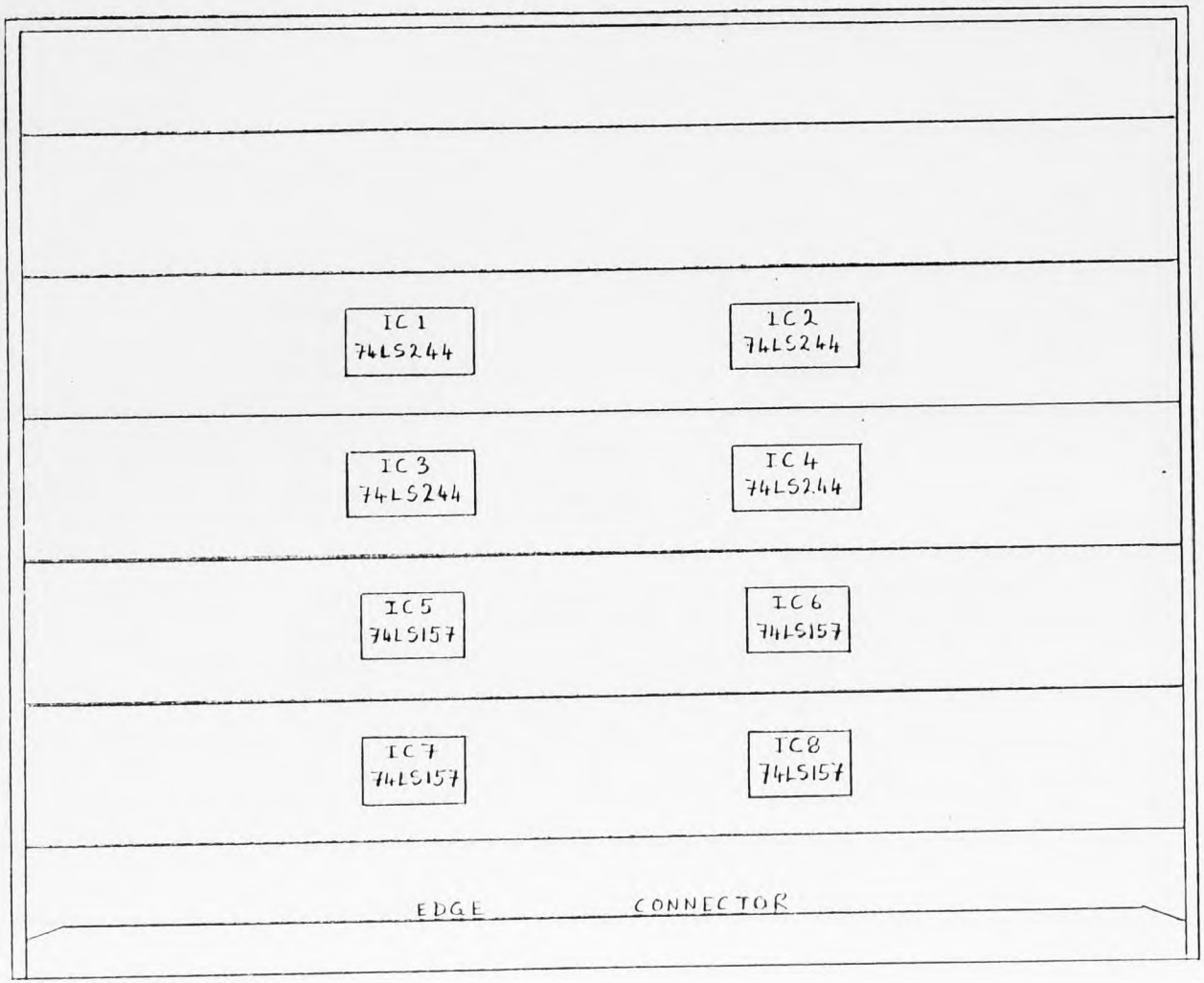
SILHOUETTE IMAGE PREPROCESSOR: SHEET 9 (PART OF BOARD 4)



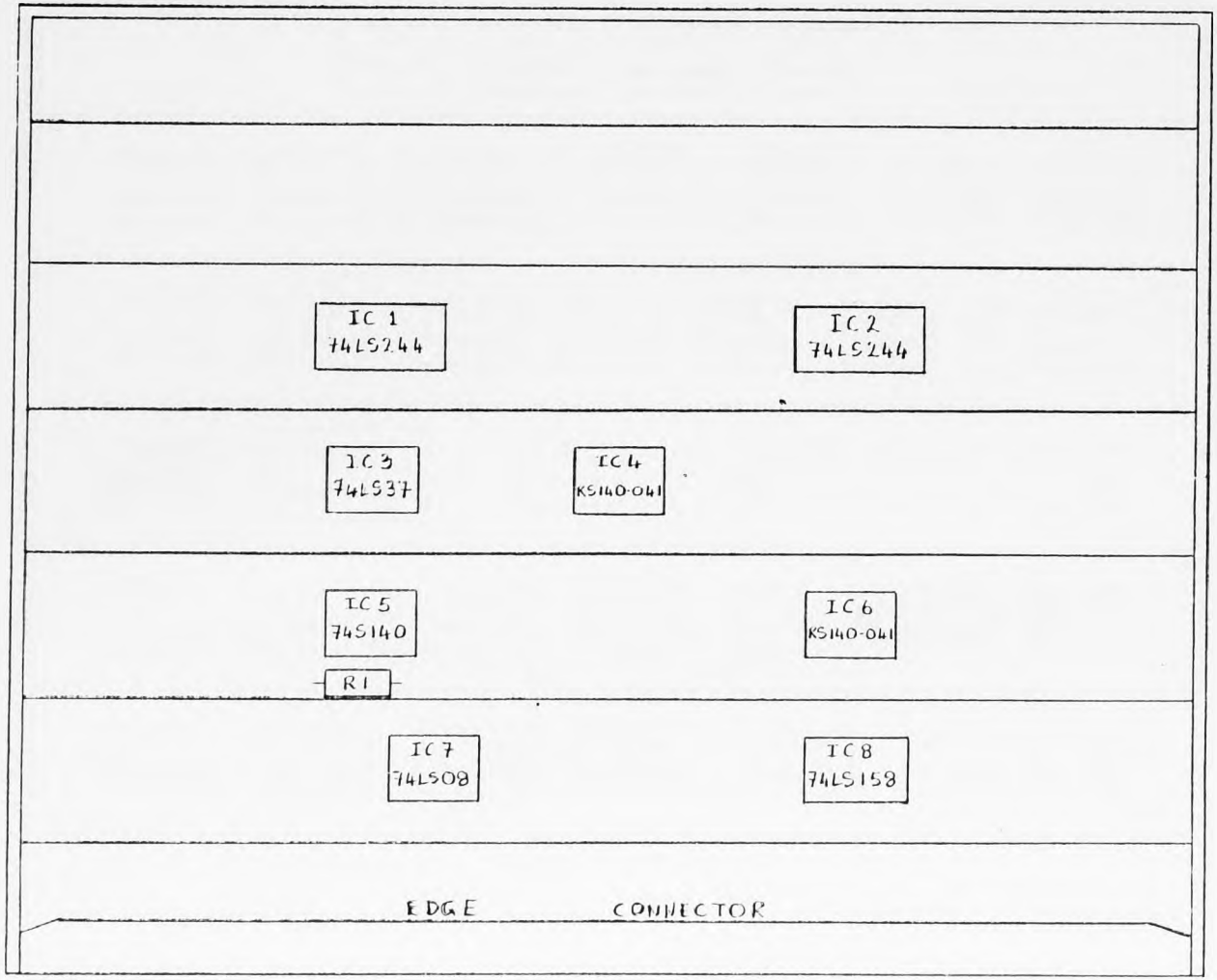
SILHOUETTE IMAGE PREPROCESSOR: SHEET 10 (BOARD 5)



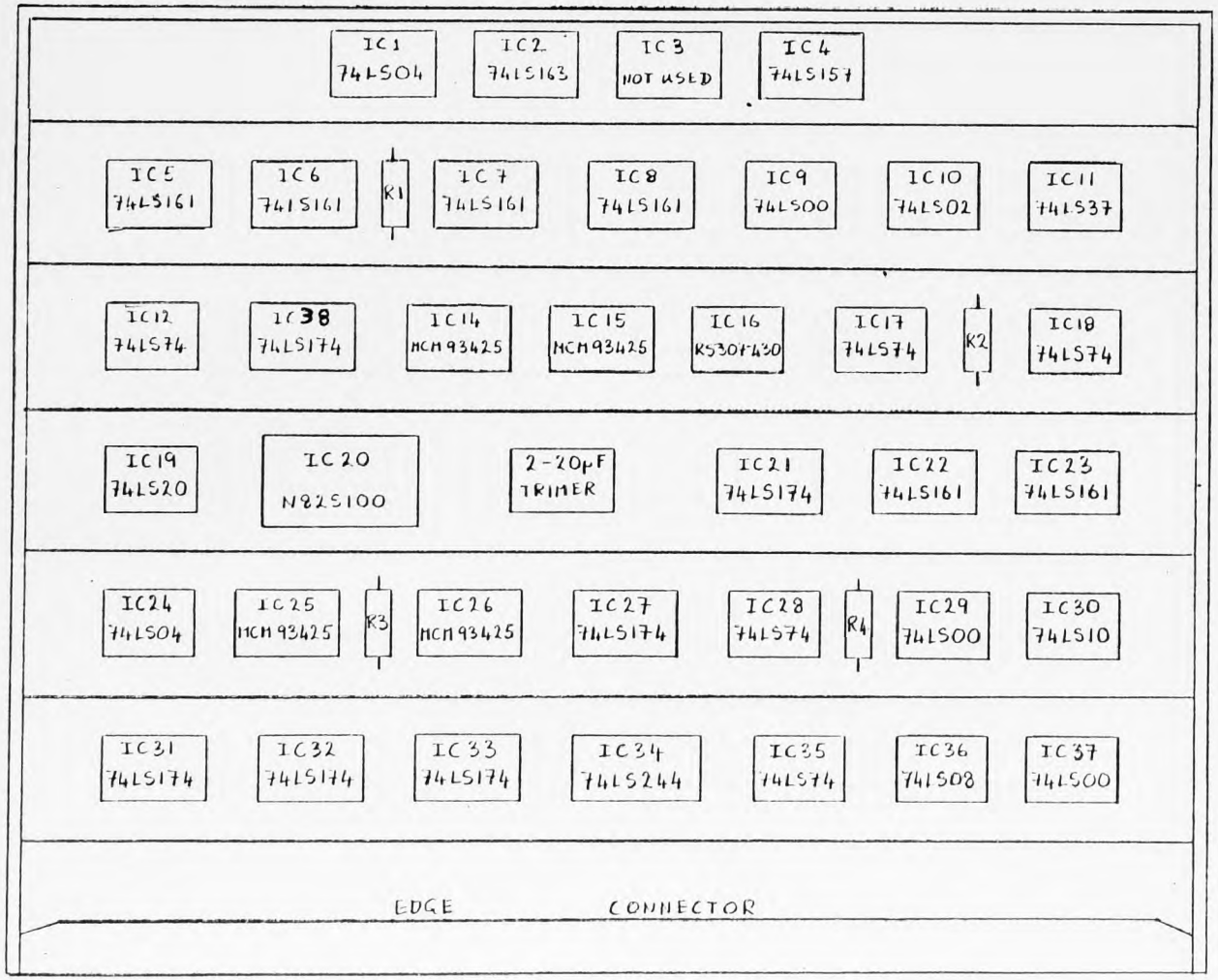
BOARD 1 LAYOUT - PART OF MULTIPLEX FUNCTION



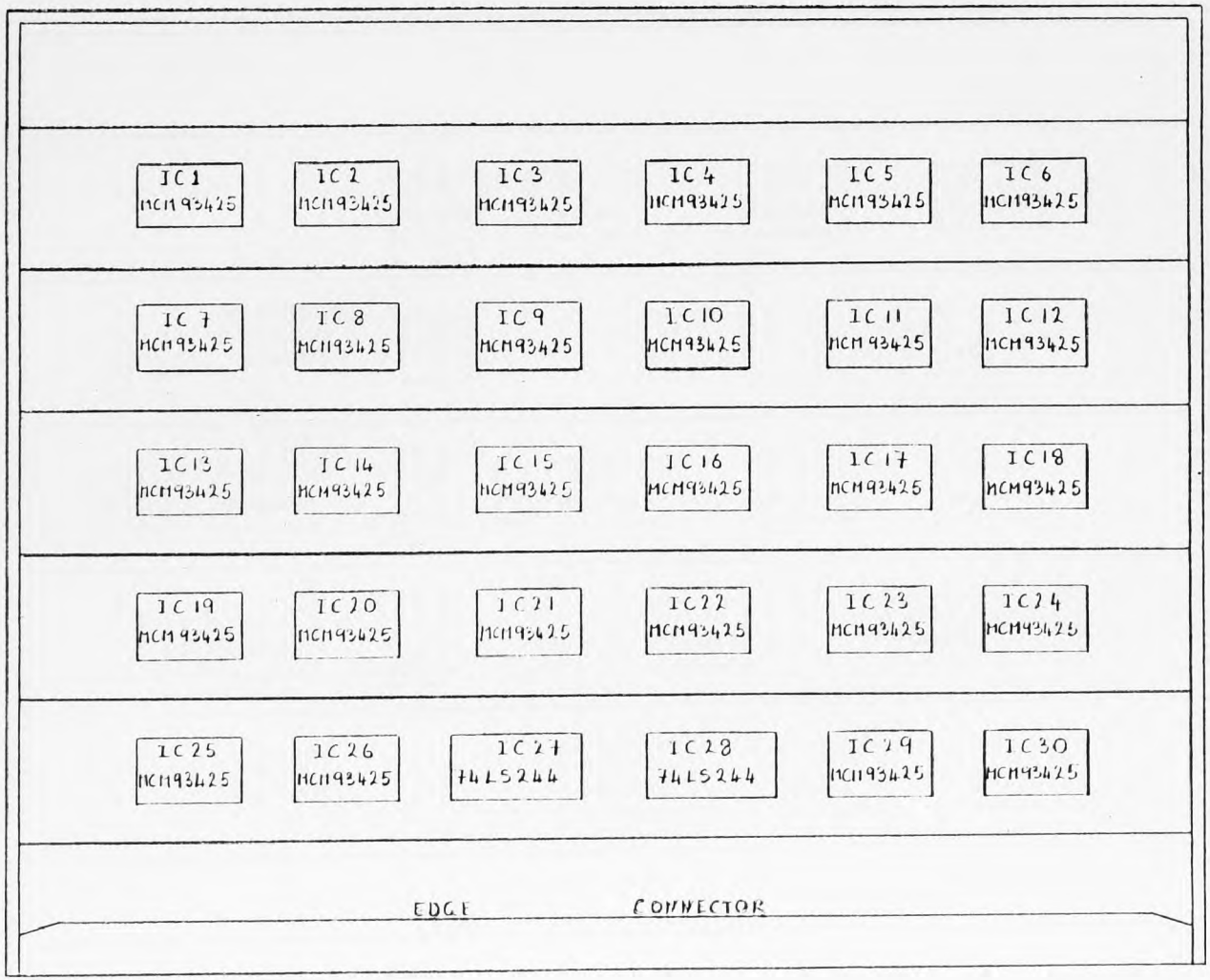
BOARD 2 LAYOUT - PART OF MULTIPLEX FUNCTION



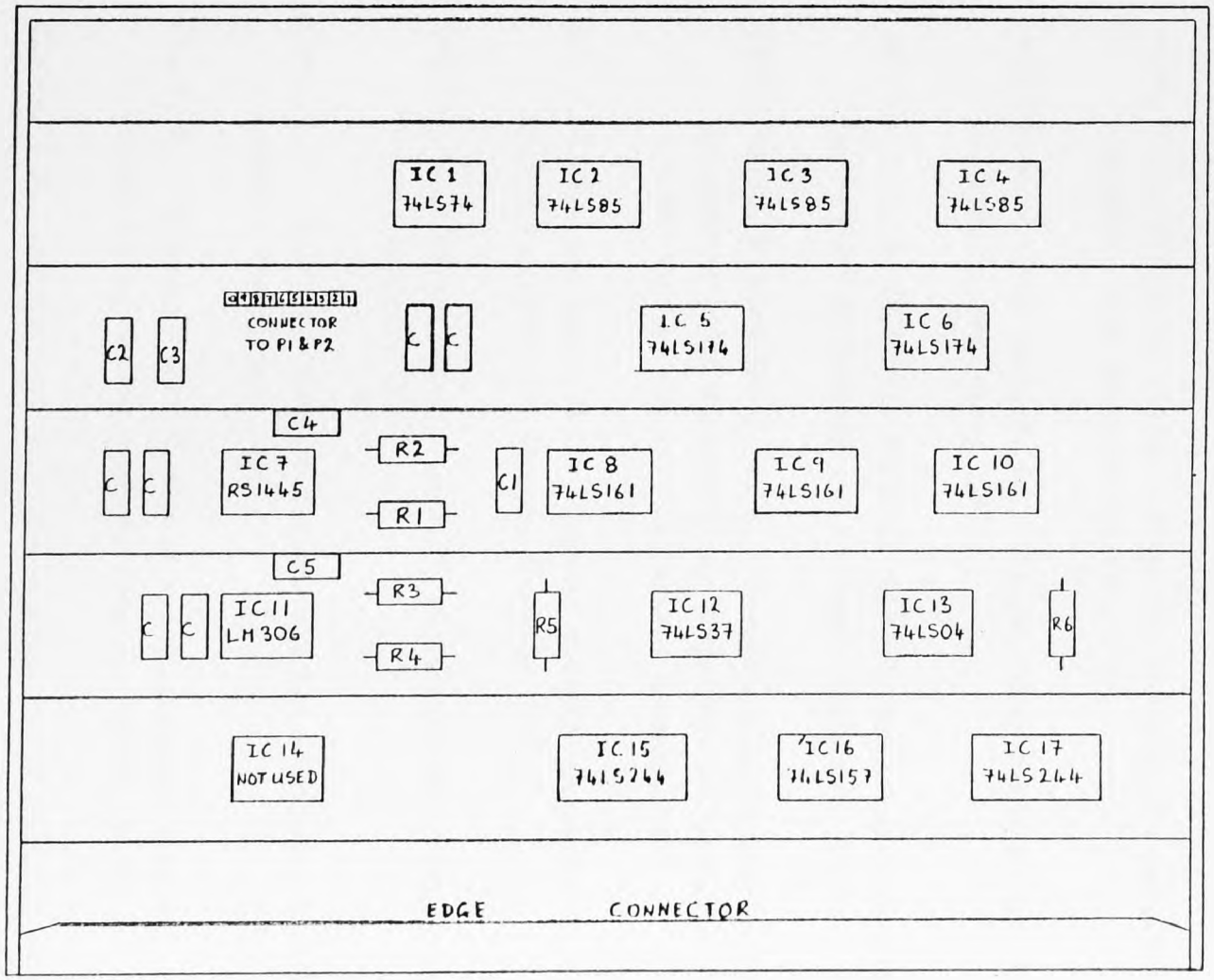
BOARD 3 LAYOUT - PART OF MULTIPLEX FUNCTION



BOARD 4 LAYOUT - PART OF SILHOUETTE IMAGE PREPROCESSOR



BOARD 5 LAYOUT - PART OF SILHOUETTE IMAGE PREPROCESSOR



BOARD 6 LAYOUT - PART OF SILHOUETTE IMAGE PREPROCESSOR

REFERENCES

1 Abdani, S. K., "Feature extraction algorithms", Department of Computer Science, University of Wisconsin, U.S.A., 1970.

2 Abramson, N., "Information theory and coding", McGraw-Hill, 1963.

3 Agin, G. J., "Computer vision systems for industrial inspection and assembly", IEEE Computer Society, 1980.

4 Andrews, H. C., and Hunt, B. R., "Digital image restoration", Prentice-Hall, Inc., New Jersey, 1977.

5 Beynon, J. D. E. (Editor), "Optical self-scanned arrays", A Symposium, Institution of Electronics and Radio Engineers, 1979.

6 Bhattacharya, S., and Rooks, B. W., "Microprocessor controlled optical measurement", 5th International Conference on Automated Inspection and Product Control, Stuttgart, Germany, 1980.

7 Boyle, W. S., and Smith, G. E., "Charge coupled semiconductor devices", Bell Syst. Tech. J., 49, pp. 587-593, 1970.

8 Brezina, J., "Radius density method", University of London Computer Centre, Seminar, 1977.

9 Chen, C. H. (Editor), "Pattern recognition and artificial intelligence", Academic Press, Inc., 1976.

10 Clark, M., "Numerical description of two dimensional shapes", University of London Computer Centre, Seminar, 1978.

11 Clarke, G.M., and Bedford, J., "The application of the minicomputer to high speed on-line optical inspection systems", Proc. VII IMECO Conference, paper AQC119, London, 1976.

12 Cornsweet, T. N., "Visual perception", Academic Press, Inc., 1970.

13 Dodwell, P. C., "Visual pattern recognition", Holt, Rinehart and Winston, Inc., 1970.

14 Dudani, S. A., "Region extraction using boundary following", 'Pattern recognition and artificial intelligence', Academic Press, pp. 216-231, 1976.

15 Eccles, M. J., McQueen, M. P. C., and Rosen, D., "Analysis of the digitised boundaries of planar objects", Pattern Recognition, Pergamon Press, vol. 9, pp. 31-41, 1977.

16 Elrlich, R., and Weinberg, B., "Method for characterisation of grey shape", Sedimentology, pp. 205-212, 1970.

17 Fairchild Semiconductors, Ltd., "CCD 121HC Development Board Brochure", 1976.

18 Fam, A. T., "Discriminant analysis of digitised curves", 3rd Int. J. Conf. on Pattern Recognition, 1976.

19 Finkelstein, L., Norton-Wayne, L., and Hill, W. J., "Advanced signal processing in automatic inspection", Proc. IICA, Recent Advances in Measurement Technology, Melbourne, Australia, 1975.

20 Freeman, H., "On the digital computer classification of geometric line patterns", Proc. Nat. I. Electronics Conference, vol. 18, pp. 312-324, 1962.

21 Freeman, H., and Glass, J. M., "On the quantisation of line-drawing data", IEEE Trans. Syst. Sci. Cyber., 5(1), pp. 70-78, 1969.

22 Fry, P. W., "Silicon photodiode arrays", Review Article, Journal of Physics E: Scientific Instruments, vol. 8, pp. 337-349, 1975.

23 Fry, P. W., "Enhancement of uniformity of response of self-scanned arrays", Pt. 5 of "Optical Self-Scanned Arrays", A Symposium, The Radio and Electronic Engineer, vol. 49, No. 10, pp. 506-508, 1979.

24 Fry, P. W., "Applications of self-scanned integrated photodiode arrays", The Radio and Electronic Engineer, vol. 46, No. 4, pp. 151-160, 1976.

25 Goodyear, C. C., "Signals and information", Butterworth & Co., Ltd., 1971.

26 Granlund, G. H., "Fourier preprocessing for hand print character recognition", IEEE Trans., vol. 21, pp. 195-201, 1972.

27 Gregory, R. L., "Eye and brain", World University Library, 1974.

28 Grenander, U., "Pattern synthesis", Springer Verlag, New York, 1976.

29 Hansen, G. L., "Introduction to solid-state television systems", Prentice-Hall, Inc., 1969.

30 Hurvich, L. M., and Jameson, D., "Perception of brightness and darkness", Ally & Bacon, Inc., Boston, 1966.

31 Hyvarinen, L. P., "Information theory for systems engineers", vol. 5 of Lecture Notes in Operational Research and Mathematical Economics, 1968.

32 Institute of Physics, "Machine perception of patterns and pictures", The Institute of Physics, Conference Series No. 13, 1972.

33 Jarvis, J. F., "Pattern recognition in line-drawing using regular expressions", 3rd Intl., Jnt. Conf. on Pattern Recognition, pp. 189-192, 1976.

34 Jarvis, J. F., "Visual inspection automation", IEEE Computer Society, 1980.

35 Kammenos, P., "Performance of polar coding for visual localisation of planar objects", Proc. 4th Intl. Conf. Robot Technology, pp. 143-154, 1978.

36 Kasdan, H. L., and Mead, D. C., "Automatic visual inspection by optical computing", 3rd International Conference on Automated Inspection and Product Control, Nottingham, England, 1978.

37 Koulopoulos, C., "Survey of optical data gathering transducers", The City University, London, Department of Systems Science, Research Report DSS/CK/148, 1977.

38 Medalia, A. I., "Dynamic shape factors of particles", Powder Technology, 4(3), pp. 117-138, 1970/71.

39 Nahin, P., "Slope density method", IEEE Comput., vol. 21, p. 1233, 1972.

40 Norton-Wayne, L., Hill, W. J., and Finkelstein, L., "Image enhancement and pre-processing", SPIE Proceedings, vol. 130, 'Automation and Inspection Applications of Image Processing Techniques', pp. 29-35, 1977.

41 Oppenhein, A. V. (Editor), "Applications of digital signal processing", Prentice-Hall, Inc., New Jersey, U.S.A., 1978.

42 Papoulis, A., "Probability, random variables and stochastic processes", McGraw-Hill, New York, 1965.

43 Parks, J. R., "Image processing - a new production tool", IEE Colloquium on 'The Industrial Uses of Image Processing', 1978.

44 Pavlidis, T., "Algorithms for shape analysis of waveforms', Proc. 4th Intl. Conf. on Pattern Recognition, pp. 70-85, Kyoto, Japan, 1978.

45 Pavlidis, T., "Structural pattern recognition", Springer-Verlag, Berlin, New York, 1977.

46 Persoon, E., and Fu, K. S., "Shape discrimination using Fourier descriptors", 2nd Intl. Jnt. Conf. Patt. Recog., pp. 126-130, Copenhagen, Denmark, 1974.

47 Peterson, C., "Automated visual inspection", Second Int. Jnt. Conf. on Pattern Recognition, pp. 41-42, Copenhagen, Denmark, 1972.

48 Piper, D. J. W., "Data processing in biology and geology", edited by Cutbill, J. T., pp. 97-103, 1970.

49 Purll, D. J., "Automatic inspection and gauging using photodiode arrays", I. Mech. E., 1976.

50 Purll, D. J., "Survey of the present state of the art in applications of solid-state image scanners", Proc. of SPIE, pt. 145, pp. 9-17, SIRA, 1978.

51 Purll, D. J., "Automated surface inspection with solid-state image sensors", Proc. Soc. Photo-Opt. Instrum. Engrs., pt. 145, pp. 18-25, 1978.

52 Raphael, B., "The thinking computer", W. H. Freeman & Co., San Francisco, U.S.A., 1976.

53 Rose, A., "Vision - human and electronic", Plenum Press, New York, London, 1974.

54 Rosenfeld, A., "Picture processing by computer", Academic Press, New York, 1969.

55 Rosenfeld, A., and Kak, A. C., "Digital picture processing", Academic Press, New York, 1976.

56 Sangster, F. L. J., and Teer, K., "Bucket-brigade electronics, new possibilities for delay, time-axis conversion and scanning", IEEE J. Solid-State Circuits, SC-4, No. 3, pp. 131-136, 1969.

57 Schwarz, H. P., and Shane, K. C., "Measurements of particle shape by Fourier analysis", Sedimentology, pp. 213-231, 1969.

58 SIRA, "Automatic surface inspection systems", Sheet ref. No. SD82.

59 SIRA, "Automation and inspection application of image processing techniques", International Seminar, The City University, London, 1977.

60 Sklansky, J., "On filling cellular concavities", Computer Graphics and Image Processing, pp. 236-247, Academic Press, Inc., 1975.

61 Tou, J. T., and Gonzalez, R. C., "Pattern recognition principles", Addison-Wesley Publishing Co., 1974.

62 Vann, M. A., "Self-scanned photodiode arrays, characteristics and applications", Optics and Laser Technology, pt. 6, pp. 209-218, 1974.

63 Weimer, P. K., et al, "Photodiode arrays", IEEE Spectrum, pp. 52-65, 1969.

64 Zahn, C., "A formal description for two dimensional patterns", Proc. Int. Jnt. Conf. Artificial Intelligence, pp. 621-628, 1969.

65 Zahn, C., and Ruskies, R. Z., "Fourier descriptors for plane closed curves", IEEE Trans. Computers, pp. 843-849, 1974.