REAL TIME ANALYSIS OF VIDEO SIGNALS

BY

OMER MOHAMED ABDEL GADIR, B.Sc.,M.Sc.,M.I.E.E.E.

A Doctoral Thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of the Loughborough University of Technology

SUPERVISOR: D.J. Quarmby, B.Sc., Ph.D., M.I.E.E.E.,

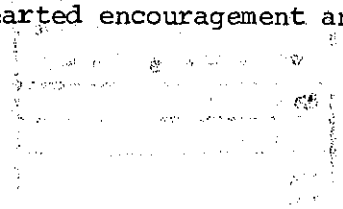
Department of Electronic and Electrical Engineering

ACKNOWLEDGEMENTS

I would like to thank Professor J. W. R. Griffiths, the former Head of the Department of Electronic and Electrical Engineering, and Professor I. R. Smith, the current Head of the Department, for providing research facilities. I am grateful to my supervisor, Dr. D. J. Quarmby for his help, encouragement and constructive criticism throughout the course of this research. In addition, I am indebted to Mr D. W. Hoare for his help and encouragement. Sincere thanks are due to Mr M. A. Sinclair of the Department of Human Sciences for his assistance and for the loan of specialised equipment.

I wish to express my appreciation to the technical and secretarial staff of the Department of Electronic and Electrical Engineering who have been a constant source of help. It is a pleasure to record my gratitude to my friends and colleagues Mr J. W. Goodge and Dr. G. M. Duck with whom I have had many helpful discussions. Sincere thanks are due also to Mrs Ann Hammond for her immaculate typing of the Thesis.

I am greatly indebted to my family, in particular my brother Abdel Rahman, for their whole-hearted encouragement and financial support.



SYNOPSIS

Many practical and experimental systems employing image processing techniques have been built by other workers for various applications. Most of these systems are computer-based and very few operate in a real time environment.

The objective of this work is to build a microprocessor-based system for video image processing. The system is used in conjunction with an on-line TV camera and processing is carried out in real time. The enormous storage requirement of digitized TV signals and the real time constraint suggest that some simplification of the data must take place prior to any viable processing. Data reduction is attained through the representation of objects by their edges, an approach often adopted for feature extraction in pattern recognition systems. A new technique for edge detection by applying comparison criteria to differentials at adjacent pixels of the video image is developed and implemented as a preprocessing hardware unit. A circuit for the generation of the co-ordinates of edge points is constructed to free the processing computer of this task, allowing it more time for on-line analysis of video signals.

Besides the edge detector and co-ordinate generator the hardware built consists of a microprocessor system based on a Texas Instruments TMS 9900 device, a first-in-first-out buffer store and interface circuitry to a TV camera and display devices. All hardware modules and their power supplies are assembled in one unit to provide a stand-alone instrument.

The problem chosen for investigation is analysis of motion in a visual scene. Aspects of motion studied concern the tracking of moving objects with simple geometric shapes and description of their motion. More emphasis is paid to the analysis of human eye movements and measure-

ment of its point-of-regard which has many practical applications in the fields of physiology and psychology. This study provides a basis for the design of a processing unit attached to an oculometer to replace bulky minicomputer-based eye motion analysis systems. Programs are written for storage, analysis and display of results in real time.

CONTENTS

Glossary, Principal Symbols and Abbreviations

ADC	analogue-to-digital converter
B/W	change in grey level from black to white (positive going edge)
CCIR	International Consultative Committee of Radio
$CD^n F(\vec{w})$	nth order central difference
CPU	central processing unit
CRT	cathode ray tube
CS	chip select
CRU	communication register unit
D_H, D_V	horizontal and vertical components of spatial differentials
DAC	digital-to-analogue converter
DMA	direct memory access
$d(\bar{x})$, g (x)	decision or discriminant function
$d_i(\bar{x})$	decision function of class w_i
EF	even field
$E^k_{i,j}$	difference of the average grey levels of a pair of $2^k \times 2^k$ neighbourhoods at (i,j)
F^{-1}	inverse Fourier transform
f_x, f_y	spatial frequencies
FD	field drive
FF	flip-flop
FFT	Fast Fourier Transform
FIFO	first-in - first-out
F(j,k) , g(x,y)	a picture function which is proportional to the grey level at the point (j,k) , (x,y)

$G(j,k)$	processed picture function
$G_i(f_x,f_y)$	Fourier transform of the input picture function $g_i(x,y)$
$G_o(f_x,f_y)$	Fourier transform of the output picture function $g_o(x,y)$
$\vec{h} = (h_j,h_k)$	interval at which spatial differences are taken
$H(f_x,f_y)$	filter transfer function
$h(x,y)$	filter impulse response
I/O	input/output
J	sum of squared distances between cluster mean and points within the cluster
LD	line drive
LSB	least significant bit
M_j	sample mean vector of set S_j (cluster mean)
MOS	metal oxide silicon
MS	mixed sync
MSB	most significant bit
MVB	mixed video blanking
N_c	number of cluster domains
$n(x,y)$	noise picture function
P.C.board	printed circuit board
PC	program counter
pefs	picture elements per frame
$P_{gg}(f_x,f_y)$	power spectral density of $g(x,y)$
$P(J\|I)$	probability of getting the noise distorted picture function $J(x,y)$ given the original picture function $I(x,y)$

$P_{sg}(f_x,f_y)$	cross-power spectral density of s(x,y) and g(x,y)
$P(\bar{x}\|w_j)$	probability density function for $\bar{x}$ conditional on w_j being the state of nature
$P(w_j)$	a priori probability that nature is in state w_j
$P(w_j\|\bar{x})$	conditional density function of class w_j (a posteriori probability)
R	maximum size of a cluster
RAM	randomly accessible memory
ROM	read-only memory
RS-232	standard for serial data communication
S	number of patterns
S_j	set of samples belonging to the jth domain
ST	status register
s(x,y)	ideal picture function
$\hat{s}(x,y)$	optimal approximation to s(x,y)
T	minimum dwell time
$T_L(j,k)$, $T_U(j,k)$	lower and upper threshold values
TTL	transistor-transistor logic
TV	television
TTY	teletypewriter
$\vec{w} = (j,k)$	pixel location
VAS	video analysis system
V/Div	volts per division
w_i	ith pattern class
W/B	change in grey level from white to black (negative going edge)

WP	workspace pointer
$\overline{x}$	pattern vector
x_c, y_c	centre of corneal reflection
x_p, y_p	centre of pupillary disc
$\lceil\alpha\rceil$, $\lfloor\alpha\rfloor$	ceiling and floor operators
∇	gradient operator
∇^2	Laplacian operator
$\nabla^n F(\vec{w})$	nth order backward difference
$\Delta^n F(\vec{w})$	nth order forward difference
$\lambda(\alpha_i \mid w_j)$	loss incurred for taking action α_i when the state of nature is w_j
μ_i	ith component of the mean
ξ_i	edge fitting error
σ	linear operator
$\sum$	covariance matrix
$\rightarrow$	becomes
*	convolution
$\approx$	approximately equal to
(subscript)16	hexadecimal character, base 16
$\in$	belongs to

CHAPTER 1

INTRODUCTION

1.1 Background

Pattern recognition has aroused profound interest because of its potentially effective practical use in various fields and its sensational applications. To formulate an artificial recognition mechanism, some researchers have endeavoured to understand the learning technology of living beings and then computerise it. This approach has proved to be ineffectual due to the complexities of the process of perception of living organisms, a great deal of which is not yet comprehended; and the extensive parallelism inherent in the biological processes. In spite of efforts in the fields of philosophy, psychology and physiology, the use of computers to emulate human perception or that of other living creatures remains largely speculative in nature.

A more pragmatic approach in solving the problem of pattern recognition has been to construct a machine for classifying given images into certain preassigned categories. This has been achieved through the formulation of many analytical and syntactic techniques and their implementation in digital computers. In this respect research deals usually either with general classification procedures, independent of applications, or with techniques strongly dependent on specific applications such as typewritten character recognition, fingerprint and chromosome classification, etc., (both approaches are inevitably and inextricably inter-related). Unfortunately, while enlightenment on the many complex and subtle requirements of

pattern recognition systems has been provided by the first approach, it has failed to yield a general solution to the problem of recognition of objects and situations in spite of many simplifying assumptions. It is noteworthy that a major contribution of the general approach is the richness of the developed concepts and procedures which are applicable and effective in solving some facets of the whole problem.

The second approach has resulted in realisation of practical working systems. The processing performed by such systems can be very demanding as a result of the complexity of the algorithms employed and the computational time required. Because of the unavailability of learning mechanisms, whereby a machine teaches itself how to classify patterns there is a great degree of man-machine co-operative interaction. An operator has to teach an uneducated machine how to distinguish between different class patterns and describe situations employing, in most cases, heuristic algorithms and relying to a great extent on his intuition. In practice, the solution to each problem is formulated from a set of scarce empirical data representing class patterns.

To summarise the previous discussion: all practical and experimental pattern recognition systems are goal-guided, or in other words each system is devoted to the solution of a particular problem or set of problems and the majority of these systems are computer-based. The advent of single-chip microprocessors with their low cost and computational capabilities offers an attractive alternative to the bulky and expensive computer-based systems. Implementation of pattern recognition techniques can be realised through

the design of cheap stand-alone instruments. In such instruments the overhead cost and the complex hardware essential to control the multi-user environment are eliminated.

The accelerated development of microprocessors and the concurrent advances in memory technology have already ushered a revolutionary era in the design of "intelligent" instruments. Like other fields of human knowledge, pattern recognition will benefit to a great extent from this development. In the author's opinion self-contained microprocessor-based pattern recognition systems will become a fast moving and proliferating phenomenon.

1.2 Objectives

The objectives of this research were twofold:-

1. To design and build a microprocessor-based real time video analysis system (VAS) of reasonable power and versatility.
2. To use the system in conjunction with an on-line TV camera to provide solutions to some aspects of the pattern recognition problem.

These objectives crystallised as a result of the availability of single-chip 16-bit microprocessors and low cost digital storage devices. The 8-bit microprocessors introduced in the early seventies are inadequate for manipulating standard CCIR 625 line TV images when pixel location on a raster is identified using a Cartesian coordinate system overlaid on the video image.

In addition to the microprocessor system, VAS incorporates a preprocessing unit and interface circuitry to a TV camera, data terminal and display devices. Transportability was of particular

concern and for this reason all hardware modules and the system's power supply were assembled in an 18u high cabinet to provide a self-contained device.

Two main constraints have greatly influenced the design and development of VAS and these are:-

1 The available microprocessors have neither the computing power nor the sophistication of mainframe computers, minicomputers or array processors.*

2 Microprocessors, being sequential machines and much slower than other digital computers, are not endowed with the capability of performing complicated processing within the context of real time video.

Due to these constraints the ability of a microprocessor to accomplish complex pattern recognition techniques is very limited (if not impossible). On the other hand it is quite feasible to implement some aspects of both analytical and syntactic approaches provided that the patterns under investigation are simple and the algorithms employed do not require long execution time.

The real time constraint makes it extremely difficult for a microprocessor to execute any technique in which the whole video frame or even a small part of it has to be stored and then processed pixel by pixel. To circumvent this constraint requires a specialised

---

* *When the design of VAS was contemplated in early 1977, the contemporary 16-bit microprocessors were much less developed than today's. Since then there have been great strides in microprocessor architectural evolution and recently much more powerful microprocessors have been introduced.*

preprocessing hardware to reduce the excessive volume of pictorial data into a format amenable for quick and efficient processing.

In view of the above discussion,the following guidelines were adopted during the formalisation of the design procedure:-

1. As a suitable representation of a picture is by line drawing the preprocessor, or feature extractor, was envisaged to be an edge detector in which the observed pattern is represented by a list of coordinates of its edge points. Storing edge patterns rather than a complete video image greatly reduces the storage requirement.

2. Rather than striving for the unattainable general approach towards solving scene analysis problems, it was decided to follow the plausible and pragmatic notion of building a goal-guided system.

3. For easy programming the system should be provided with a monitor and an assembler.

4. It is very difficult and sometimes very expensive to program the system for all eventualities when real data is processed, so it is wise to leave the door open for human interaction.

The problem domain under investigation was defined as that of tracking of moving objects in a visual scene. VAS was used to observe moving targets and from the gathered information an appropriate representation of the situation is displayed to the operator on a storage oscilloscope. More attention was paid to the analysis of human eye movements and measurement of its point-of-regard which has wide applications in the clinical, physiological and psychological

fields, and in human factors studies. The eye sensing device used to monitor the eye movements is an oculometer, which tracks two elements of the eye detail; the pupillary disc and the corneal reflection. The output of the oculometer is a video signal containing pertinent details of the eye and can be connected directly to VAS.

1.3 Summary

In this section a brief overview of the thesis is presented.

In Chapter 2 a summary view of pattern recognition techniques is presented to enlighten the complex, subtle and sometimes paradoxical requirements of pattern recognition systems. Both decision-theoretic classification procedures and the fundamentals of automatic scene analysis are covered. Block diagrams summarising the techniques and illustrating the relationship between them are provided. The last part of the chapter analyses the requirements of real time systems.

Parallel edge detection techniques, in contrast to sequential edge detectors, can be applied simultaneously everywhere in an image and as a consequence are more advantageous concerning speed and flexibility. Chapter 3 is devoted to such techniques and explains both linear and non-linear operations. Evaluation of the computational requirement for different approaches is included to gain an insight into the complexities of the operations.

Chapter 4 opens with a brief comment on the parallel edge detection schemes in view of the adequacy of their implementation as hardware preprocessing units. A novel edge detection procedure satisfying the constraint of real time video analysis is introduced. This is followed by presentation of its hardware realisation and results obtained using real world scenes.

To create a stand-alone device the hard-wired edge detector was interfaced to a processing unit. The processing facility is provided by a TMS9900 microprocessor system. A brief description of the microprocessor and factors influencing the design procedure are given in Chapter 5. The hardware design of the microprocessor system, the interface circuitry and control is described at system level with the aid of block diagrams representing the rudimentary building blocks. Detailed circuit diagrams are illustrated in an appendix. The software support is briefly explained.

General application programs for storage and display of edge patterns extracted from the video image are discussed in Chapter 6. In this chapter and elsewhere in the thesis flowcharts are emphasised as opposed to the coding of individual programs for reasons of generality and economy of space. A final section deals with the requirements of real time programming and the constraints imposed by the processing of video signals.

Chapter 7 describes algorithms designed to provide real time information on the position of moving objects. Factors that render the formulation of a reasonably general computational solution to the tracking problem a difficult task are covered, together with simplifying assumptions deemed essential to obtain practical solutions. Problems chosen for investigation are tracking of a single object, tracking of two objects and identification of a moving object in a scene which, in addition, includes a stationary object. Objects tracked are circular discs exercising planar motion. Vector fields representing velocities of tracked objects are plotted on the storage oscilloscope using stored motion history data.

Chapter 8 gives an account of some features of the anatomy of the human eye and characteristics useful for eye movement measurements, followed by a review of techniques used to monitor the movements of the eye and their implementation. The majority of the techniques for the measurement of the point-of-regard require either attachments to the subject or careful stabilisation of his head. Alternatively,the head position can be monitored and head position signals are combined with eye motion signals to obtain a measure of the point-of-regard. Unlike other techniques, the double Purkinje image tracking system and the corneal reflection-pupil centre measurement system allow relatively free natural head motion. More emphasis is paid to the second technique as it is the basis of an oculometer used in conjunction with VAS to perform eye motion analysis.

The description of a real time system for eye motion analysis is presented in Chapter 9. An oculometer built in the Department of Human Sciences, University of Technology, Loughborough, was used in proving trials of the system. A software package has been designed for on-line acquisition of optical features of the eye, calculation of its point-of-regard and its display in real time on the storage oscilloscope. Concurrently,displayed data is stored to provide the time history of the point-of-regard for further processing. A set of interactive programs allows the reduction of the stored data into a form where its salient features can be evaluated, and yield a graphic display of fixation point scan patterns. The experimenter can proceed directly from monitoring a subject's eye to statistical analysis and display of scan patterns without any manual manipulation of data. Results obtained using simulation data

and data collected while subjects were viewing different stimulus patterns are presented. The graphic display of fixation point scan paths is indicative of the adaptive scan procedure adopted by subjects trying to assimilate information contained in stimulus patterns.

Important conclusions resulting from this research and closing remarks are the subject of Chapter 10. An overall evaluation of VAS, its performance and applicability in other fields of research are presented.

The appendix provides the reader with detailed circuit diagrams of the whole system.

CHAPTER 2

PATTERN RECOGNITION TECHNIQUES: AN OVERVIEW

2.1 Introduction

The study of recognition machines probably began with the works of McCullock and Pitts (1943), and Wiener (1948)[1]. Phenomenal advancement of the digital computer technology has triggered such an enormous expansion in the field of pattern recognition, that now it has become a subject of interdisciplinary study comprising engineering, computer science, statistics, linguistics, psychology, physiology and medicine. The bulk of material dealing with it is widely scattered and there is no general methodology towards solving the problems of machine perception. In this chapter different pattern recognition techniques are surveyed to investigate the possibility of implementing some of them in a microprocessor-based real time system.

2.2 Pattern Recognition Techniques

Pattern recognition is the study of artificial, as well as natural mechanisms that analyse, detect, recognise, and describe patterns in sensory and/or numerical data[1]. Multitudes of techniques have been directed towards the solution of the problem of pattern recognition[1-7], and these could be grouped, as shown in Fig.2.1, into analytical techniques, which deal with patterns on a quantitative basis, and syntactic techniques which explicitly utilise the structure of patterns in the recognition process. Furthermore, the analytical techniques are performed either by applying statistical classification

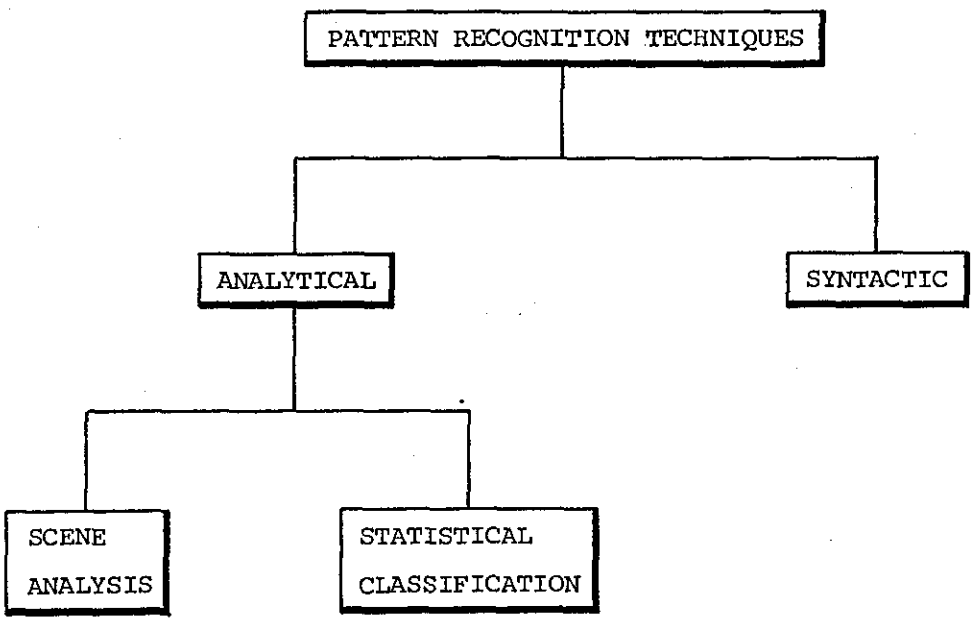


Fig. 2.1: Classification of Pattern Recognition Techniques

theory or by employing computer scene analysis methods.

The simplest form of a recognition machine, is a sorter of data into classes, a functional block diagram of which is shown in Fig. 2.2. The first step is concerned with the representation of input data which can be measured from the objects to be recognised, i.e. the sensing problem. This is followed by extraction of characteristic features or attributes from the received input data and reduction of the dimensionality of pattern vectors by preprocessing and feature extraction techniques. The last step involves the determination of optimum

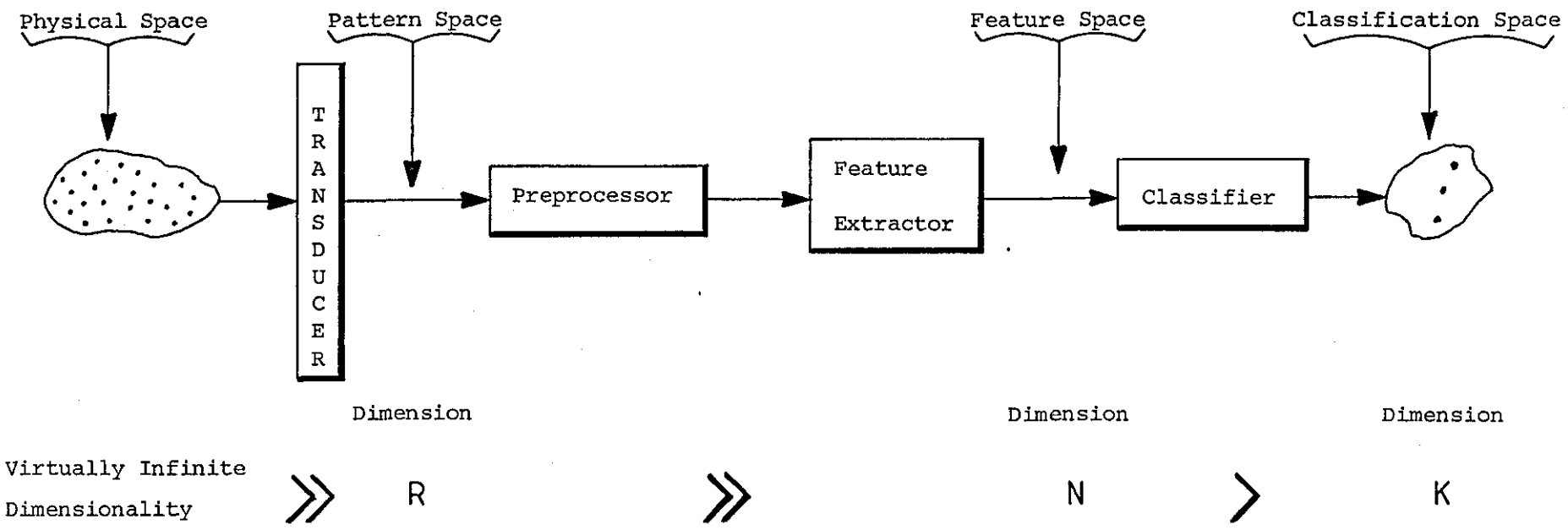


Fig. 2.2: Functional Block Diagram of a Pattern Recognition System[1,4]

decision procedures, which are needed in the identification and classification process. Although the distinction between preprocessing, feature extraction and classification is not essential, the concept of functional breakdown provides a better understanding of the pattern recognition problem.

Other aspects of recognition technology include searching for patterns, detecting them, determining their location in the field of perception, describing and summarising them, tracking of moving objects and description of their motion.

2.3 Statistical Decision Theory

Chow (1957), and Sebestyen (1961,1962), were among the first to apply statistical decision theory in the design of pattern classifiers[1]. Since then statistical decision theory has found wide applications in the field of pattern recognition.

In an automatic classification scheme, using a decision-making process, the feature extractor reduces each class to feature vectors $\bar{x}$ in an n-dimensional Euclidean space[2,3], where

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The set of patterns belonging to the same class correspond to an ensemble of points scattered within some region of the feature space. Assuming that the machine is designed to recognise S different classes denoted by $w_1, w_2, \ldots w_s$, then the feature space can be viewed as consisting of S regions each of which encloses the pattern points

of a class[3]. The machine has to generate the decision boundaries separating those regions. Defining the decision boundaries by scalar and single-valued decision functions, also called discriminant functions, $d_1(\bar{x})$, $d_2(\bar{x})$, ..., $d_s(\bar{x})$; if $d_i(\bar{x}) > d_j(\bar{x})$ for $i,j = 1, 2, \ldots, m$, and $i \neq j$ then vector $\bar{x}$ belongs to pattern class w_i. In other words if the ith decision function, $d_i(\bar{x})$, has the largest value for a vector $\bar{x}$ then $x \in w_i$[3].

2.3.1 Bayes' decision theory

The design of a statistical pattern classifier is generally based on the Bayes' classification rule and it's variations. A general formulation of Bayes' theorem is given by[2]:

$$P(w_j|\bar{x}) = \frac{P(\bar{x}|w_j)P(w_j)}{\sum_{j=1}^{S} P(\bar{x}|w_j)P(w_j)}$$

where $w_1, w_2, \ldots, w_s$ are finite set of S states of nature, $\bar{x}$ a d-component feature vector, $P(\bar{x}|w_j)$ the state-conditional probability density function for $\bar{x}$, i.e. the probability density function for $\bar{x}$ conditional on w_j being the state of nature, and $P(w_j)$ the a priori probability that nature is in state w_j.

The Bayes' theory gives the relation between the a posteriori probability $P(w_j|\bar{x})$ and the probability density function $P(\bar{x}|w_j)$. Eliminating the scaling factor $\sum_{j=1}^{S} P(\bar{x}|w_j)P(w_j)$, the decision rule for the simple case of two pattern classes is:

Decide w_1 if $P(\bar{x}|w_1)P(w_1) > P(\bar{x}|w_2)P(w_2)$;

otherwise decide w_2.

In the design of optimal classifiers this concept could be extended to include actions other than classifications, for instance refusing to make a decision in close cases. This is achieved by introducing the loss function which also treats situations in which some mistakes are more costly than others. Let $\lambda(\alpha_i|w_j)$ be the loss incurred for taking action α_i when the state of nature is w_j. The expected loss is given by the conditional risk $R(\alpha_i|\bar{x})$ where

$$R(\alpha_i|\bar{x}) = \sum_{j=1}^{S} \lambda(\alpha_i|w_j)P(w_j|\bar{x})$$

Here the decision rule is: select action α_i for which $R(\alpha_i|\bar{x})$ is minimum. Classifiers could also be represented in terms of discriminant functions $g_i(x)$, for i = 1.., c, and these are given by

$$g_i(\bar{x}) = -R(\alpha_i|\bar{x})$$

The minus sign is introduced because maximum discriminant functions correspond to minimum conditional risk. From above analysis, to minimise probability of error one should always choose the state of nature that maximises the a posteriori probability. When all errors are equally costly then

$$g_i(\bar{x}) = P(w_i|\bar{x})$$

i.e. maximum discriminant functions correspond to maximum a posteriori probability.

One of the main problems in applying Bayes' theory in pattern recognition is that the conditional densities are not known. Sometimes the densities are known to be or assumed to be multivariate normal

but the values of the mean vectors and covariance matrices are not known. Under such assumptions it could be proved that the discriminant function is given by[2],

$$g_i(\bar{x}) = \bar{x}^t W_i \bar{x} + \bar{w}_i^t \bar{x} + w_{io} \quad ,$$

where

$$W_i = -\frac{1}{2} \sum_i^{-1}$$

$$w_i = \sum_i^{-1} \bar{\mu}_i$$

and $w_{io} = -\frac{1}{2} \bar{\mu}_i^t \sum_i^{-1} \bar{\mu}_i - \frac{1}{2} \log|\sum_i| + \log P(w_i)$

Here $\bar{\mu}_i$ is the ith component of the mean, $\sum_i$ the covariance matrix, $\sum_i^{-1}$ its inverse, $|\sum_i|$ its determinant, $\bar{x}^t$ and $\bar{\mu}^t$ are the transpose matrices. Fig. 2.3 shows the computational procedure for an optimal recognition scheme.

2.3.2 The statistical classification strategy

Bayes' decision rule yields an optimal classifier when the a priori probability $P(w_j)$ and the class conditional densities $P(\bar{x}|w_j)$ are known[2]. But in most practical applications this is not the case, as the only available information is a set of samples representative of the data to be recognised and a vague knowledge about the situation. The strategy followed in implementing the statistical decision theory depends upon the type of information which could be extracted from the representative samples and an outline of this strategy is shown in Fig. 2.4.

If the set of design samples are labelled to show their category membership i.e. representation patterns from each class are available, then supervised pattern recognition techniques are

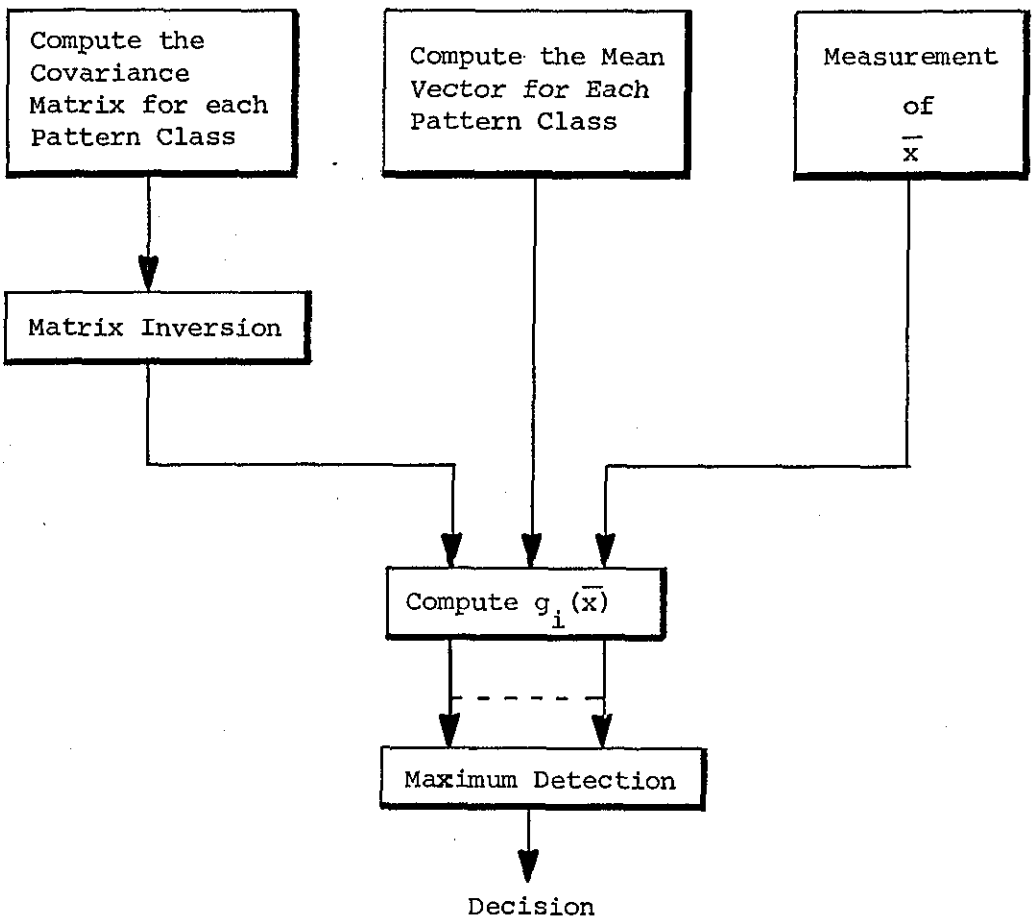


Fig. 2.3: Computational Procedure for an Optimal Recognition Scheme[3]

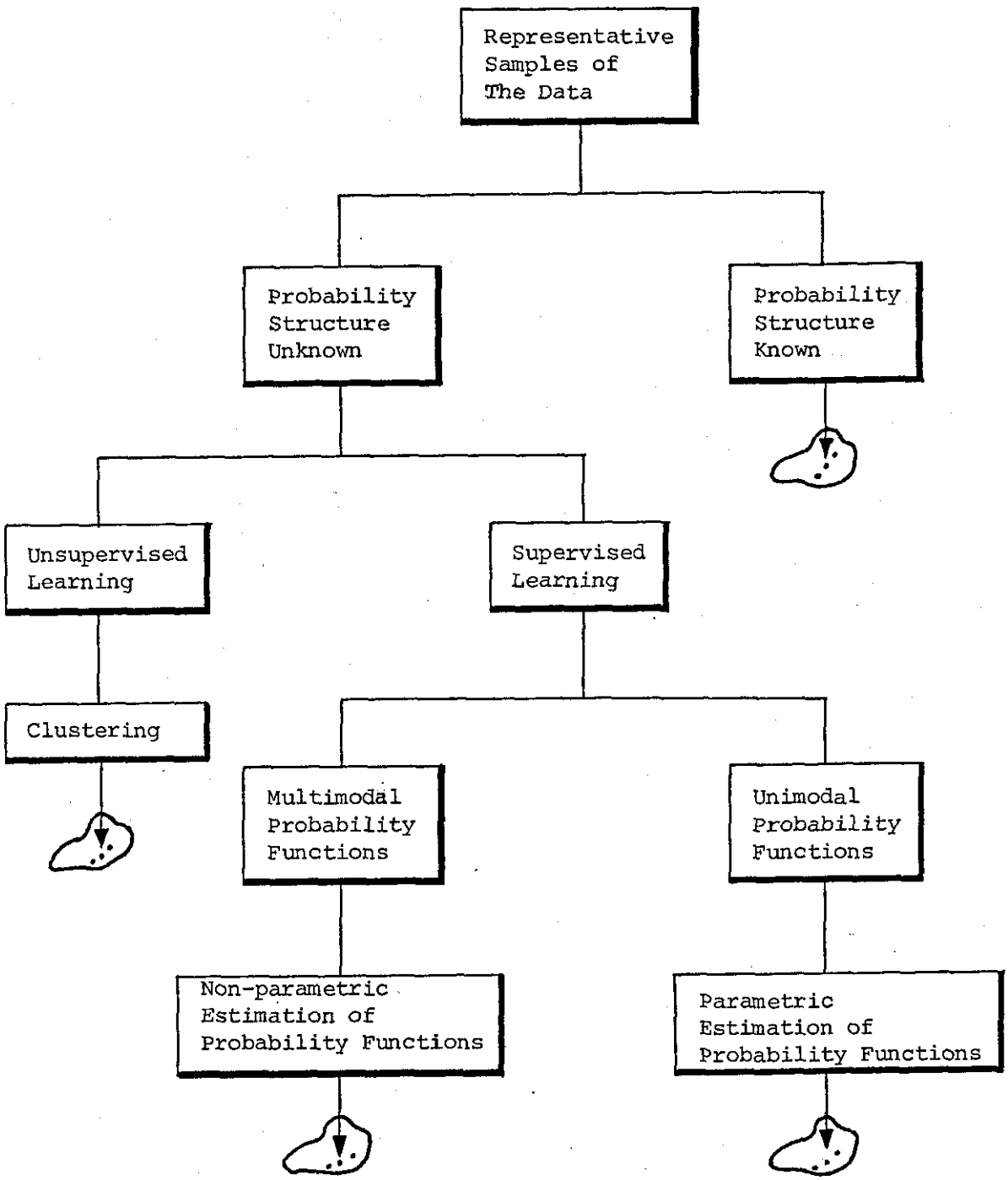


Fig. 2.4: A Block Diagram Summarising the Statistical Classification Strategy

adopted, where the system is taught to recognise the patterns by what is known as supervised learning. In supervised learning if the forms of the underlying density functions are known to be unimodal, parametric estimation of the probability density function $P(\bar{x}|w_j)$ is performed in which $P(\bar{x}|w_j)$ is assumed to be normally distributed with mean $\bar{\mu}_j$ and covariance matrix Σ_j, although the exact values of these quantities are not known. This simplifies the problem from one of estimating $P(\bar{x}|w_j)$ to one of estimating the parameters μ_j and Σ_j. Two common and reasonable procedures to do this is the use of the maximum likelihood and Bayesian estimation methods. When the underlying density functions are multimodal non-parametric techniques are applicable. Here $P(\bar{x}|w_j)$ can be obtained by estimating the probability of, say, k samples out of a total of n falling within a particular region in a d-dimensional space. Another approach is to directly estimate the a posteriori probabilities $P(w_j|\bar{x})$ using the nearest-neighbour rule.

In case the training samples are of unknown classification, unsupervised pattern recognition techniques are applied, in which data is partitioned into subgroups or clusters. The design of a classifier based on this concept is guided by the relative geometrical arrangement of various pattern clusters.

2.3.3 Remarks

In most pattern recognition problems which arise in practice, the determination of a complete set of discriminatory features is extremely difficult, if not impossible, and the dimensionality of the feature space tends to be very large; the number of dimensions ranging from 20 to 200[3]. As this great dimensionality is a major

cause of the cost and practical limitation of statistical recognition techniques, the choice of features and the reduction of the feature space are important problems. The available techniques for overcoming them fall into the following two categories:

1) Selection of a 'best' subset of features from a large initial set.

2) Transforming the feature space into a 'reduced' feature space that has fewer coordinates.

In general, if the performance obtained with a given set of features is inadequate, new features are added to facilitate separating class pairs most frequently confused. Increasing the number of features increases the cost and complexity of both feature extractor and classifier.

In a typical pattern classification problem, the estimation of the probability density functions is dependent on the samples available. Reliable results are obtained when the number of samples is large. However, the available number of learning samples is very much limited in most applications and the best features may not be selected. Another practical consideration is that the quality of the learning samples may not be uniform. Sample estimates of the probability density functions or distance functions may vary considerably with different sets of learning samples. Increasing the number of samples increases the storage requirement and processing time, particularly for the non-parametric methods where all samples might have to be stored.

If all possible discriminant features could be extracted and unlimited time is available for processing, then adequate pattern recognition could be achieved. In actual practice, however, restrictions in time, space and cost dictate the development of realistic approaches.

2.4 Computer Analysis of Pictures

The objective of scene analysis is classification and scene description. When the problem is that of classification, the mathematical tools described before are appropriate. When the problem domain is that of description, then the objective becomes description of line, description of shape and description of motion. Scene analysis is concerned not only with a wide diversity of problem domains, but also with the enormous variability of problems within a single domain and as in statistical methods the process is essentially one of simplification by extracting a set of features.

2.4.1 Feature extraction alternatives

Image features are derived from image measures which fall into three categories: point measures, local measures and global measures[8].

a - Point Measures:

Point features are represented as a matrix with a point measure value, usually of intensity, at each pixel location. One of the advantages of point measures is their availability directly from the image sensor. At the same time, one of their big disadvantages is their poor invariance characteristics for image transformations, for instance they can vary wildly with changes in illumination. It is possible to perform transformations to overcome the lack of invariance, but as a rule these transformations are computationally complex.

b - Local Measures:

Local measures include average intensity over an area, texture properties over an area, locally connected line segments and

curves and isolated line and curve intersections. Local measures have the advantage over point measures of greatly improved invariance characteristics. This improvement is at a cost of a moderate amount of processing.

c - Global Measures:

These are usually shape descriptions based on completely connected segmented boundaries and they include description of object shapes and the relation between them. The extraction of global features requires even more processing than local features, and because of the constraint of completely connected segmentation global feature extraction is nearly impossible for complex scenes.

2.4.2 Methodologies of scene analysis

There are many methods for scene analysis and the applicability of each depends upon the problem at hand. The block diagram of Fig. 2.5 shows the main techniques and their objectives. A brief description of these techniques is given in the following subsections. Here it is assumed that the picture occupies a plane whose coordinates are x and y, and it is characterised by a picture function g(x,y) which is proportional to the grey level at the point (x,y). Computers operate on digital picture functions obtained by digitising g(x,y).

2.4.2.1 Edge detection

Existing techniques for edge detection are very numerous and the treatment of this subject is given in Chapter 3. At this stage it should be emphasised that, excluding the simple approach of thresholding

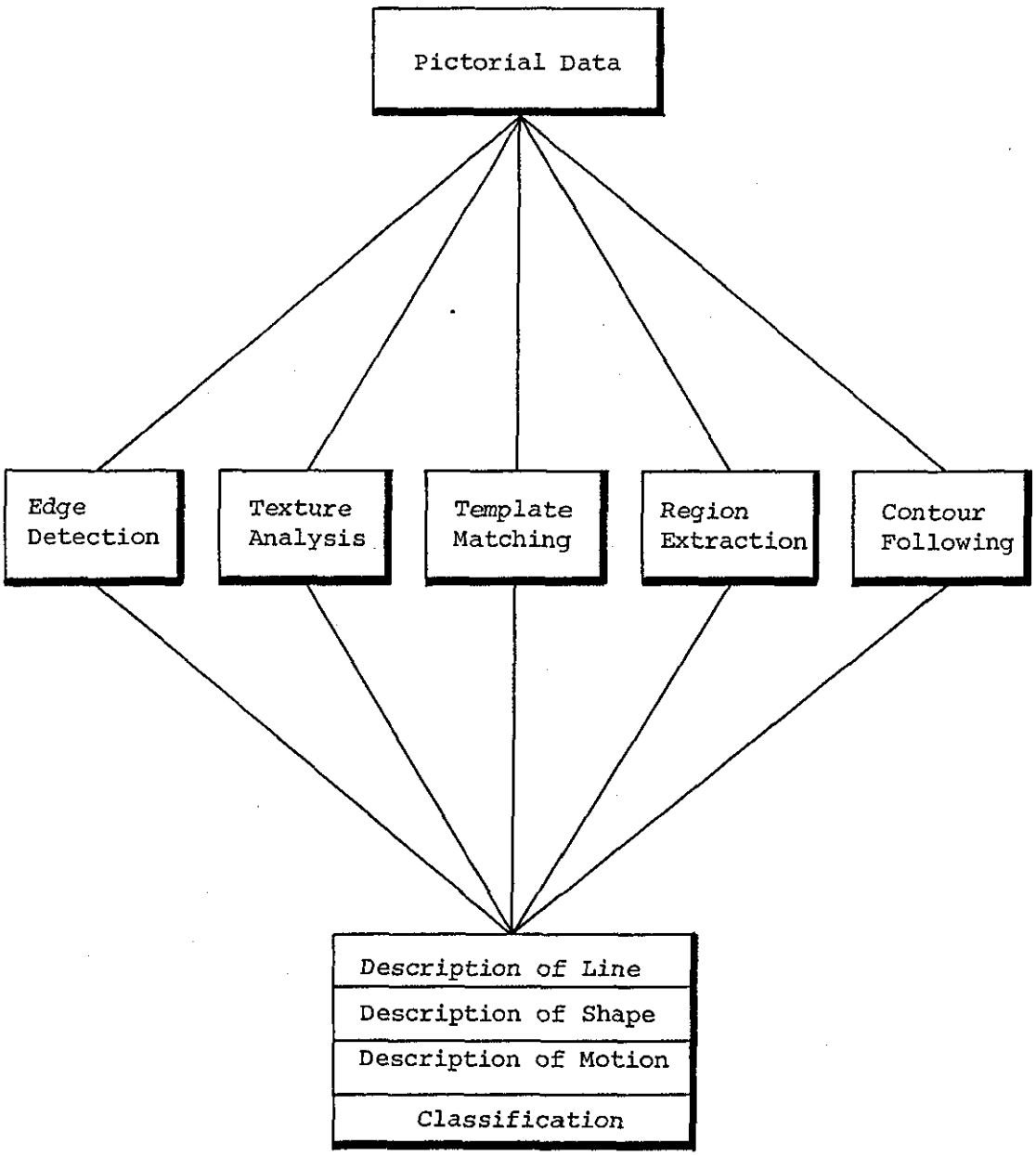


Fig. 2.5: Methodologies and Objectives of Scene Analysis

the picture function, these techniques vary greatly in their degree of complexity. Some edge detectors evaluate the magnitude of the gradient of the picture function using simple operators, while others compute products of the differences between the average grey levels for pairs of neighbourhoods of all sizes or use operators of high complexities utilising some characteristics of the picture under analysis.

2.4.2.2 Texture analysis

Texture may be classified as being artificial or natural. Artificial textures consist of arrangements of symbols (e.g. line segments, dots, stars etc.,) placed against a neutral background. Natural textures, as the name implies, are images of natural scenes containing semi-repetitive arrangements of pixels[9]. There are different methods of texture analysis, the applicability of each depends upon the type and characteristics of the texture[9,10]. One of the techniques is the evaluation of the 'texture' of various areas by going through a number of smoothing and subtracting cycles from the original picture. During each smoothing and subtracting cycle, texture descriptors (e.g. total area, contour length, annular width) are calculated. Some techniques for extracting textural properties employ autocorrelation functions, power spectra and local homogeneity measures, differences between grey-tone values of adjacent image elements and relative frequencies of various grey levels. There is no simple method for texture analysis and in most cases the computing involved is extremely time consuming and expensive.

2.4.2.3 Template matching

Template matching is used to determine whether a scene contains a previously specified object[2]. To do this a template, or mask, similar to the object is scanned systematically across the entire picture. If the object is in the scene a position will be found in which the template matches a portion of the scene. A measure of similarity between that portion and a template is given by the cross-correlation between them. Computational requirement for global template matching is severe as objects can appear in a wide range of sizes and at different orientations. Consequently, a whole family of templates are needed and each must be scanned across the scene. Individual parts of an object vary in appearance less than the entire object and to reduce the computational burden local templates can be designed to match individual parts of the object.

2.4.2.4 Region extraction

In region extraction attempts are made to simplify a digital picture by partitioning it into a set of disjoint regions. In the simplest case, adjacent pixels of identical brightness are grouped together, and then regions are merged if the difference in grey level between them is less than a specified threshold[2,11].

2.4.2.5 Contour following

Contour following is used to separate objects from the background by extracting their outlines. An algorithm performing contour following can be defined as follows[2]:

Scan the picture until a figure cell is encountered

Then:

If you are in a figure cell turn left and take a step.

If you are in a ground cell turn right and take a step

Terminate when you are within one cell of the starting point.

Fig. 2.6 illustrates the operation of this algorithm on a simple binary picture.

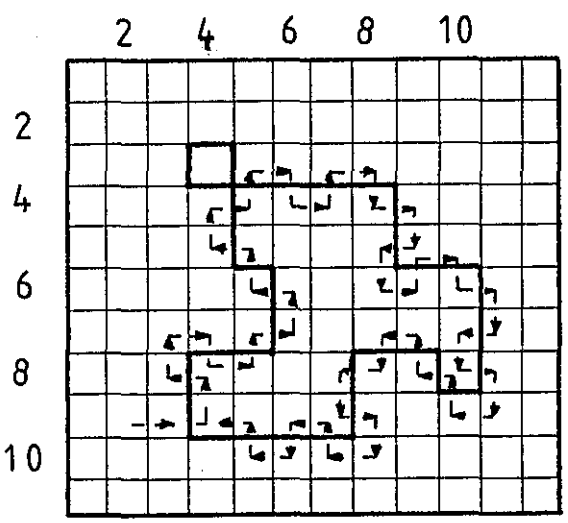


Fig. 2.6: Contour Following on a Digital Figure[2]

Prior to the application of a contour following technique, a picture has to be converted into the binary form by thresholding. Contour following is a serial operation and for this reason an error made in any step makes it more likely that subsequent steps will also be in error.

2.4.3 Picture representation and encoding

After features are extracted by one or more of the aforementioned techniques, the analysis is directed towards describing lines and shapes in the scene and position of objects and their motion.

2.4.3.1 Description of line

When figures are represented by lines the natural approach is to segment the figure and apply line fitting techniques. There are different methods for line-fitting and one of the most widely used is the minimum squared-error line fitting. Here a straight line is constructed such that the sum of squares of the vertical distances from each point to the line is minimum.

Another convenient method for representing an arbitrary curve is known as chain-encoding[2] in which a mesh is placed over the picture and the points where the curve crosses some line of the mesh are identified. The vertices of the mesh nearest to each intersection are taken to represent the curve. Fig. 2.7(a) shows the relevant vertices for the given curve. These vertices are encoded in an octal sequence by giving the direction from one vertex to the next according to the code shown in Fig. 2.7(b).

Starting at node A, the code is 1, 1, 2, 1, 0. This code specifies the angle as a function of line length, with the understanding that the diagonal lines are longer than vertical and horizontal ones by a factor $\sqrt{2}$.

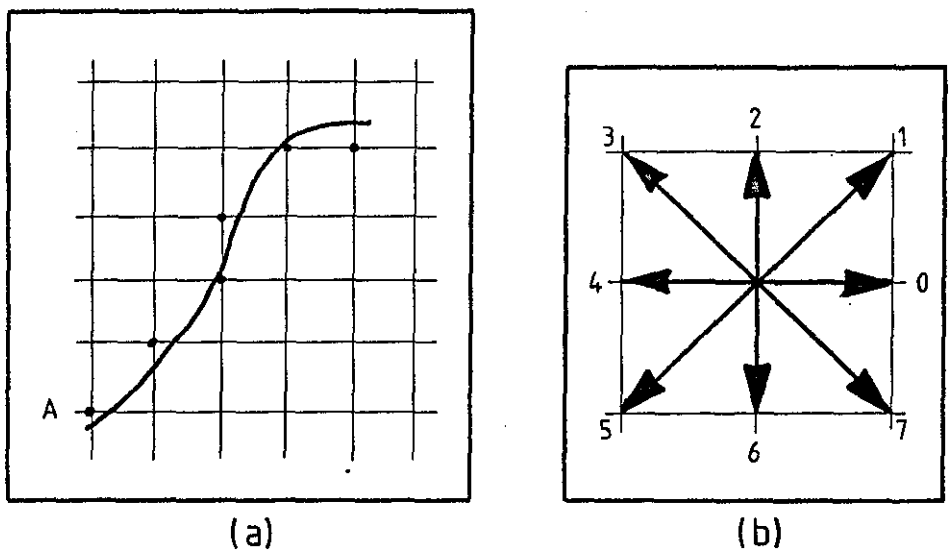


Fig. 2.7: Chain Encoding[2]

In some applications transformation techniques are used for mapping edge points in the image space into curves in the transform space on the basis of normal parameterisation of the curve. Concurrent edge points generate curves in the transform space that intersect at a common point indicating the slope and the y-intercept of the corresponding line.

2.4.3.2 Description of shape

The two methods most widely used are descriptions specifying topological[2] properties and the medial axis transform[2,12]. Topological properties are invariant to the so called rubber sheet distortions and consequently they are not a function of distance. One of the most commonly used topological descriptions of a set is the number of its connected subsets. Connectivity here means that

any two points in the subset can be joined by a line lying entirely within the subset.

The purpose of the medial axis transform is to obtain a locus or a skeleton describing the exact shape of a pattern.

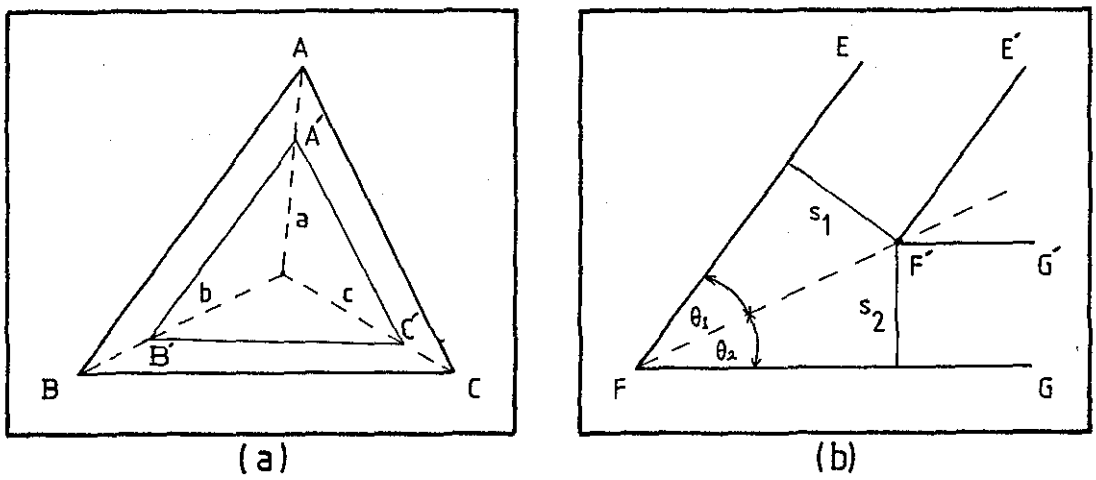


Fig. 2.8: Medial Axis Transform

Fig. 2.8(a) shows a triangle ABC and its transform. The transform is generated by causing the triangle to shrink down in size or in other words by allowing the edges to propagate with uniform velocity into the pattern. The medial axis is the locus of the intersection of the propagating edges A'B', B'C', C'A' and is given by sides a,b,c. Fig. 2.8(b) shows how a skeleton could be generated for a portion of a pattern consisting of two lines EF, FG meeting at angle $(\theta_1+\theta_2)$. After an interval of time the lines move to positions E'F' and F'G'. Since they are moving with constant speed, distances s_1 and s_2 are equal and therefore $\theta_1 = \theta_2$. Thus the skeleton FF' bisects the angle between the two branches.

2.4.3.3 Description of motion

Estimation of velocity of objects in a scene and description of their motion is useful in the fields of : image surveillance [13],

meteorological studies[14], process control[13], military applications[15], psychological studies[16], and image coding for redundancy reduction[17]. The manner in which motion is described depends upon the nature of the moving object and the reason for its tracking. For instance,in surveillance the need may be to detect any type of movement, movement in a particular direction or at a particular speed or movement of specific shape. In meteorological studies cloud motion is indicated by vector fields representing the displacement of clouds over a time interval using pictures taken by satellites. In process control the position and orientation of a moving object may be of importance, while the military, when tracking moving objects, might be interested in the azimuth and elevation pointing angles of significant points in the target. Eye movement as they pertain to psychological studies are mainly recorded as displacements of the eye point of fixation of an observer viewing certain patterns placed in front of him. In the frame-to-frame coding of television signals motion is measured in picture elements per frame (pefs).

Motion studies can be divided into two major parts: motion detection and motion analysis[18]. These divisions together with the main techniques employed are shown in Fig. 2.9.

The methods used in systems dealing with motion detection involve features which are in some sense global. In most cases the analysis yields a motion vector for an entire image or some subset of it. The analysis is usually based on cross-correlation, image differencing and statistical analysis techniques[18].

The cross-correlation technique is implemented by employing fast Fourier transform methods to estimate motion from a pair of

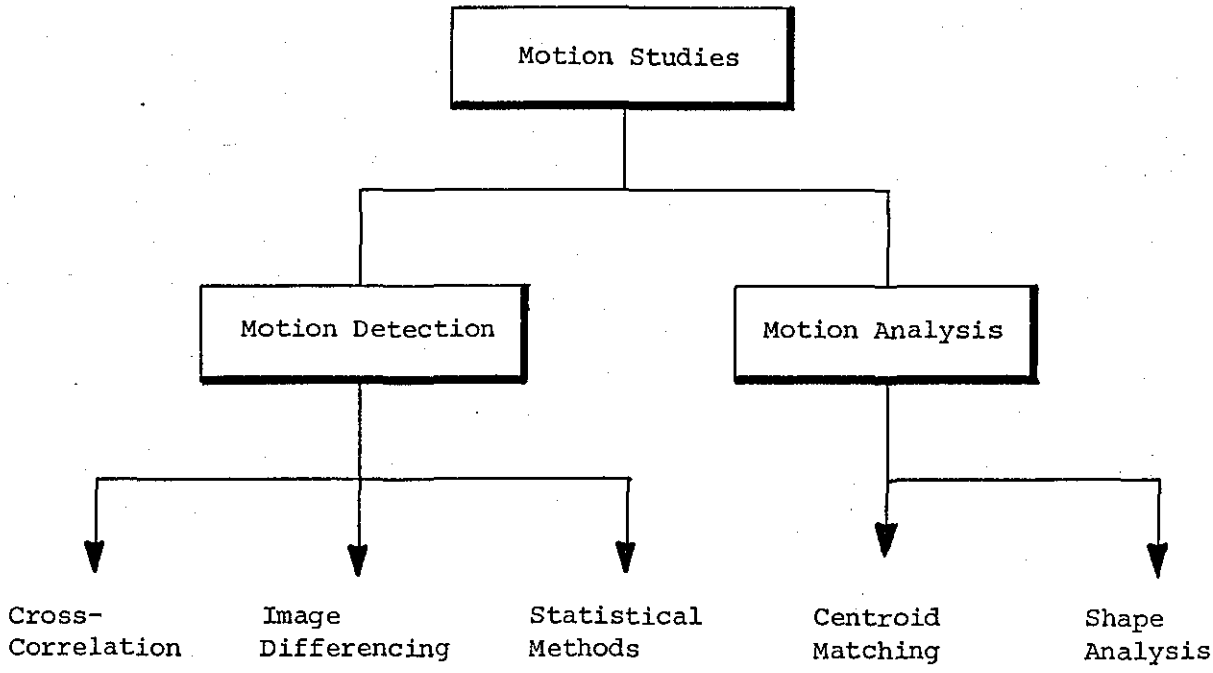


Fig. 2.9: Classification of the Motion Studies and the Techniques Employed

successive pictures, as for example in cloud motion analysis using satellite data[14,18].

Image differencing is concerned with the problem of determining areas of change between two images of the same scene[18,19,20]. The areas of difference are determined by a simple subtractive process performed prior to or after the processing of data representing the two images. Processing before subtraction involves careful alignment, by both spatial coordinates and intensity value, of the images. Spatial registration is achieved by employing cross-correlation techniques and intensity value alignment by matching the grey level values of subregions of the images. Processing after subtraction

entails the implementation of a normalising process to obtain a velocity estimate for the image. Techniques based on image differencing do not deal directly with motion analysis as they do not attempt to recognise any particular feature in either of the two successive images. They do, however, address the problem of determining areas of change between two images of the same scene or yield a velocity estimate for a picture frame as a whole. If a scene contains one moving object then the velocity obtained can be ascribed to it.

An example of a system employing statistical methods is a real time video tracking system proposed by Gilbert et al[21]. The system adopts adaptive statistical clustering and projection-based classification algorithms to identify and track a missile target.

The techniques used in motion analysis are: Centroid matching and shape analysis[18]. Centroid matching has been used in the analysis of the motion of clouds and microorganisms. At first a picture is thresholded to separate the image from the background, or in other words to extract points belonging to the image. Then clustering procedures are implemented to obtain the centroids of the extracted points. Motion analysis is based on pairing centroids in the first picture with those in the second.

Motion studies employing shape analysis have been performed on scenes containing moving opaque rigid polygons[22]. Each polygon is represented by a list of its boundary components, namely its vertices. The problem investigated is to determine from a sequence of scenes the linear and angular velocities of the polygons, and to decompose the scene in its component figures.

2.4.4 Remarks

Assuming no preprocessing is done, a computer performing picture analysis may need to allocate a large slice of memory for storing pictorial data. Taking as an example an ordinary TV picture satisfying the specifications for TV broadcasting in Britain, and selecting a sampling frequency just above 11 MHz, then such a picture contains about 570 x 570 pixels. If each pixel is represented by 8 bits, a total of about 2.6 megabits is required to describe this picture in full fidelity. Since a digital computer can work with only one point at a time (or at most, only the few points that can be packed in a word) the number of operations required to process the picture is formidable.

2.5 Syntactic Pattern Recognition

Automatic pattern recognition by linguistic or syntactic techniques has found many applications since Noam Chomsky[23] proposed in 1956 mathematical models of grammars related to his work in natural languages, and now these techniques are suitable in the analysis of bubble chamber pictures, chromosome and fingerprint pattern recognition[2,3,6].

In syntactic approach to pattern recognition, patterns are represented by a grammar, in which each pattern is described by a hierarchical structure of subpatterns (primitives), analogous to the syntactic structure of languages[3]. Since each pattern is represented by a string of primitives, the recognition of a pattern string is realised through a syntactic analysis or parsing procedure. Parsing is the procedure used to determine whether or not a string represents a sentence which is grammatically correct with respect to a given language.

Syntactic pattern recognition is particularly useful in dealing with patterns which cannot be conveniently described by numerical measurements, or are so complex that local features cannot be identified and global properties must be used. Primitive extraction and selection problems in the syntactic approach are similar to feature extraction and selection in analytical techniques, except that primitives in the syntactic approach represent subpatterns. One of the most serious problems in the syntactic approach is the detection of primitives in a scene[2]. The nature of the process makes it important to find primitives accurately since misrecognition of a single primitive can drastically alter the final result. There has not been any general mathematical approach for structural feature extraction. Finding a small but effective set of primitives still requires much human ingenuity. Also grammatical inference, i.e. learning a grammar from a set of sample sentences, is still in its infancy and the known grammatical inference schemes are limited in scope.

2.6 Requirements of Real Time Pattern Recognition Systems

In a digital context, a real time system receives and processes a set of data and gives results "sufficiently quickly" before receiving the next set of data. A real time system, makes special demands on both hardware and software and its design should be geared to solve the following;

1. Preparation of input data quickly and easily by careful design of the preprocessor and the interface units.
2. Once in the central processor, data must be processed quickly enough to avoid loss of incoming data.

The central processor unit should have fast input/output capability, efficient code and data handling, and reliability as failure at any stage might lead to loss of data. In the event of the need for many computations to be performed on a large set of data, as in picture analysis, the power of the central processor unit must be enhanced in one or more of three directions:

1. Increasing the size of the main or the backing stores.
2. Increasing the processing power of the arithmetic unit and its data handling capability.
3. Performing as much processing as possible in parallel.

To satisfy the requirements of a real time system the program can be written as a number of small logically independent modules. In this way complex computation could be done by modules each of which is responsible for a particular set of calculations. A multiprocessing system is a very suitable device as it could be arranged such that some routines are resident in the main store, while the rest are held in separate stores which can be accessed by other processors. With their powerful pipelined parallel computations the multiprocessors are very fast machines.

2.6.1 Array processors

A real time picture analysis system could be realised by implementing one of the parallel machines proposed by previous investigators[24,25,26]. The concept of parallel machines is based on local operations which are parallel in the sense that operators could be executed simultaneously, each one accepting a particular pixel and its neighbours. Unger's machine[24] is a direct implementation of this idea

and it consists of an array of small computers, each one operating on a particular area of the picture. All proposed parallel machines are highly redundant and construction has been hampered by economic constraints.

Recently, the advent of array processors as high speed computing devices has rendered the realisation of a general-purpose, real time picture analysis system feasible. They are termed array processors[27,28,29] after their ability to deliver high throughput on large arrays or vectors of data. Their computation rates greatly exceed that of a conventional sequential computer because they have one or more of the following features:

1. High level of parallelism, sometimes utilising a parallel multi-bus architecture to provide different paths for inputs and outputs.
2. Pipeline processing.
3. Fast instruction cycles.
4. Wide instruction words enabling the processor to execute many operations in a single processor cycle.
5. Two multipliers and two adders in the arithmetic unit for high speed complex multiplication and digital filtering.
6. Distribution of tasks to specialised components inside the array processor.
7. Use of more than one processing unit and more than one memory bank.

An array processor could be interfaced to a general purpose computer. It takes data and instructions from the host computer, performs the mathematical computations indicated by the instructions

and returns the processed data to the host or transfers it to a peripheral. An array processor takes over the burden of heavy computation and at the same time the host is freed to attend its own tasks. Fig. 2.10 shows the architecture of a "MAP" array processor which employs more than one processor and more than one memory bank[29].

Some array processors are supplied with a library of arithmetic subroutines for vector multiplication, evaluation of fast Fourier Transform (FFT), matrix inversion etc. These routines are called through Fortran from the host computer. Specific application routines can be created for the array processor in one of two ways. The first is by chaining routines from an existing library. The second is by creating new routines using a cross-assembler, simulator, debugger and the hardware diagnostics.

The ability of array processors to perform arithmetic operations in a very short time (38 bit floating point multiply every 500 nsec) coupled with a provision for fast transfer of data from the host[29] makes them suitable devices for real time video analysis. The only requirement is the construction of a suitable preprocessing device in order that data is transferred to the host computer in a suitable form for processing.

2.7 Comments

The ability of the human perception system to deal with a wide range of problems with enormous variability in a very short time, and in many cases in real time is very fascinating (it is possible for a human to recognise a familiar object with a tachistoscope exposure of only 4×10^{-6} sec.)[30]. Although psychological and physiological

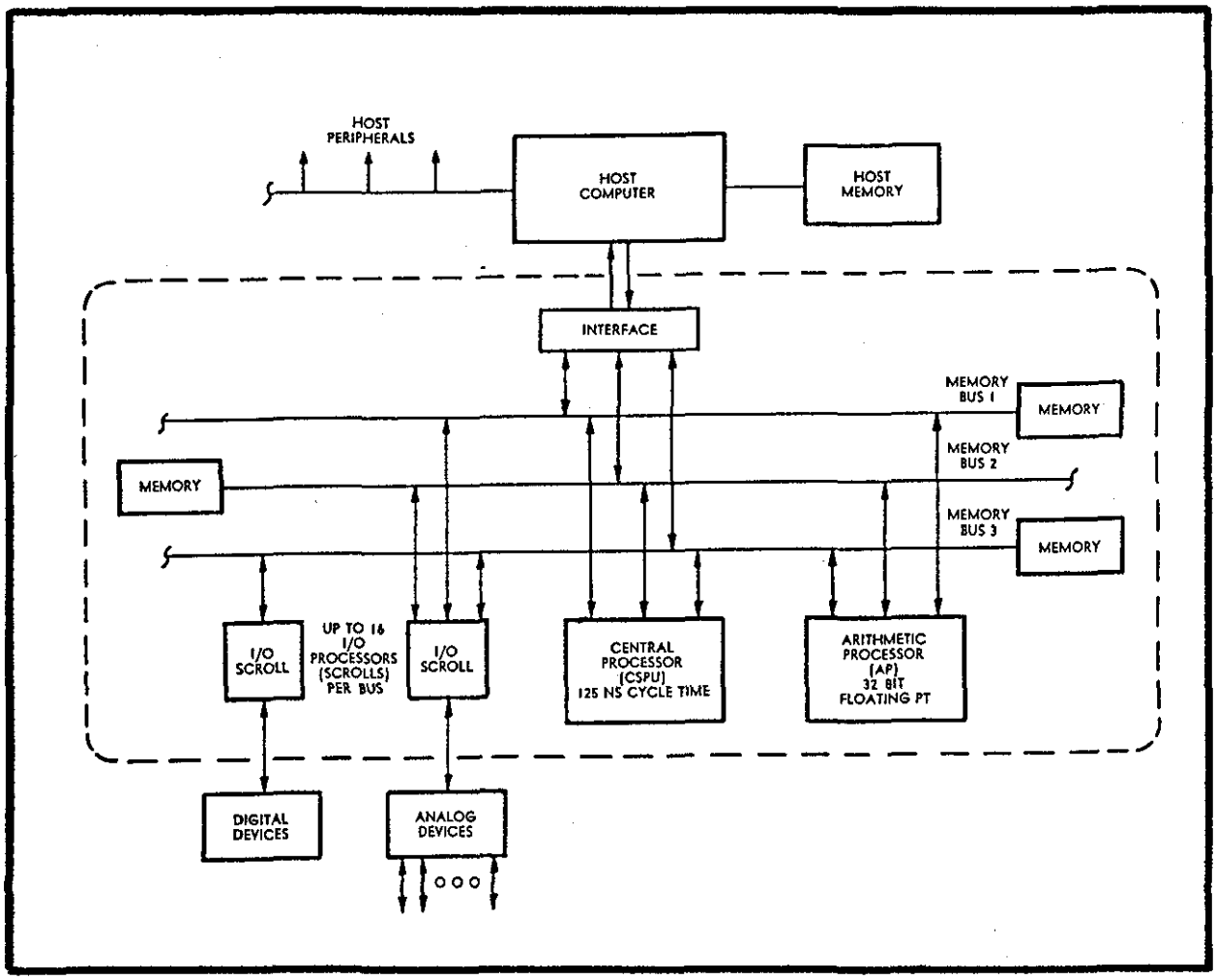


Fig. 2.10: "MAP" System Diagram[29]

studies have revealed similarities between the biological processes responsible for sensing and the properties of electronic systems,[31,32,33,34] not enough is yet understood to enable duplication of even a small part of the sensory system. The manner in which some of the nerve cells operate and the code of the nervous system are not yet known, and so researchers in their endeavours to design models of the human brain or even a small part of the sensory system, have to rely on a scant amount of information[35,36,37].

The difficulty of duplicating human perception is exasperated by the extensive parallelism in the biological processes. For instance, in the visual system the light receptors in each eye include more than 100 million rods and 7 million cones feeding into a million optic nerves[33]. The information is transmitted to the occipital cortex where there are overwhelming connections between enormous numbers of nerve cells. It has been discovered that the retina and the optic nerve perform elementary preprocessing of the field of view. Some of the most interesting revelations were the results of experiments performed by D.H.Hubel and T.N.Wiesel[38] on the functional architecture of the cats visual cortex. They found that cortical cells are feature detectors. How these features are processed in the human brain and how perception is achieved is not yet known, but there is strong evidence that when a person perceives a pattern he makes an inductive inference and associates the perception with some general concepts or cues which he has derived from his past experience. To make a machine perceive like a human brain is not yet conceivable, and consequently man remains the ultimate pattern recogniser.

2.8 Closing Remarks

To summarise, the techniques used in solving pattern recognition problems give reliable results if special characteristics of the problem at hand are well known. Any small complication in the problem might lead to enormous computational complexity. The number of applications for these techniques is quite large and they now include: radar detection, character recognition, diagnosis of radiographs, analysis of blood cells, fingerprint identification , analysis of bubble chamber photographs and description of motion. In many of these applications the major problems are:

1. Large storage requirement for data.
2. Complexity of computation, particularly in the presence of noise.
3. Time and cost of computation.

Very few of the existing systems operate in a real time environment. The array processors, with their high speed and computing power, can be used to perform some pattern recognition operations in real time. However, the greatest limitation here is their cost.

CHAPTER 3

A Survey of Parallel Edge Detection Techniques

3.1 Introduction

In a monochrome picture, an edge element is a picture element lying on the boundary between two objects or regions of different grey levels. Fig. 3.1 represents one-and two-dimensional noise free edges. The one dimensional edge is characterised by its height (h), slope angle (θ) and coordinate of the slope midpoint (x_e). An edge exists if both θ and h exceed specified values[9]. For the two dimensional edge the orientation with respect to x is also important. Edges are usually degraded by noise and irregularities of the surface structure of objects and the effect of these is depicted in Fig. 3.2.

A common approach to edge detection is based on thresholding a processed picture function. A functional block diagram of a monochrome threshold edge detector is illustrated in Fig. 3.3. The original monochrome image F(j,k) is processed to produce an image field G(j,k) with enhanced spatial brightness changes. This is followed by a thresholding operation to determine the pixel location of significant edges. A negative going edge exists if

$$G(j,k) < T_L(j,k)$$

and a positive going edge exists if

$$G(j,k) > T_U(j,k)$$

where $T_L(j,k)$ and $T_U(j,k)$ represent the lower and upper threshold values respectively. Thresholding of the original image function F(j,k)

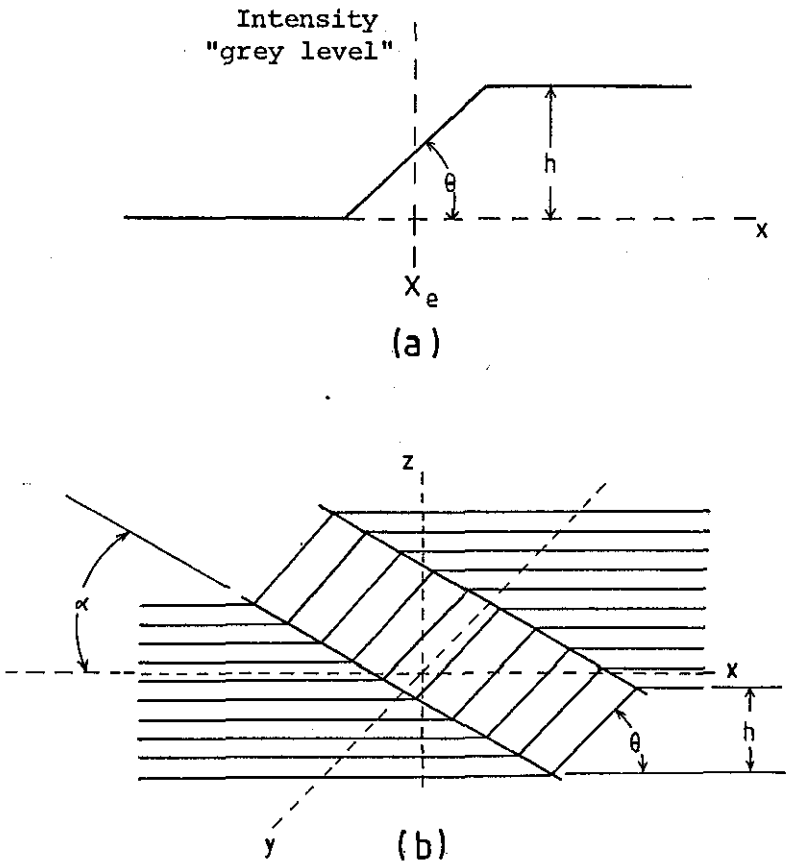


Fig. 3.1(a) One-Dimensional Edge

(b) Two-Dimensional Edge[9]

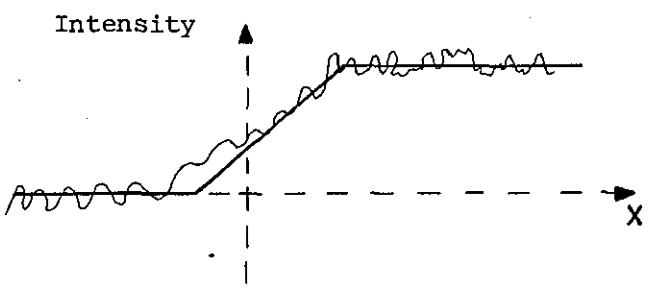


Fig. 3.2: A Real Noisy Edge Superimposed on a Noise Free Edge

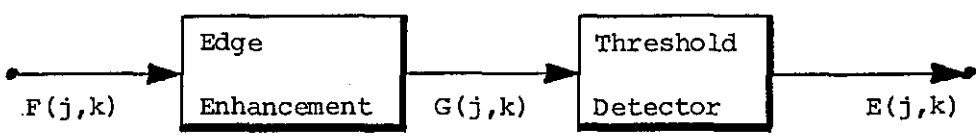


Fig. 3.3: Threshold Edge Detection System[9]

is historically the first method of segmentation applied to images[39]. In general, it produces very poor results and it has very limited applications.

Another approach to edge detection involves fitting a local region of pixel values to some ideal representation of a one- or two-dimensional edge. If the fit satisfies a certain criterion, an edge is assumed to exist, and it is assigned the same parameters attributed to the ideal edge.

A considerable number of edge detection schemes has been devised and these can be categorised as shown in Fig. 3.4. In parallel edge detection techniques, the decision whether an edge exists or not is based on analysis of grey levels of pixels in a neighbourhood, but the decision is not dependent on whether other neighbourhoods have edges. Consequently the edge detection operator can,in principle,be applied simultaneously everywhere in the picture. In contrast to parallel edge detectors, the result of a sequential edge detection operator at a point is contingent upon decisions made at previously examined points. Performance of a sequential edge detector depends upon[40]:

1. Choice of a good starting point. Usually some simplifying assumption is made to facilitate that choice, e.g. the edge begins in the top row of a picture.
2. Formulating the dependence structure, or in other words the manner in which decisions at previously examined points affect choice and results at the next point to be examined. Functions describing the dependence structure are usually complex and require high execution time for their implementation.

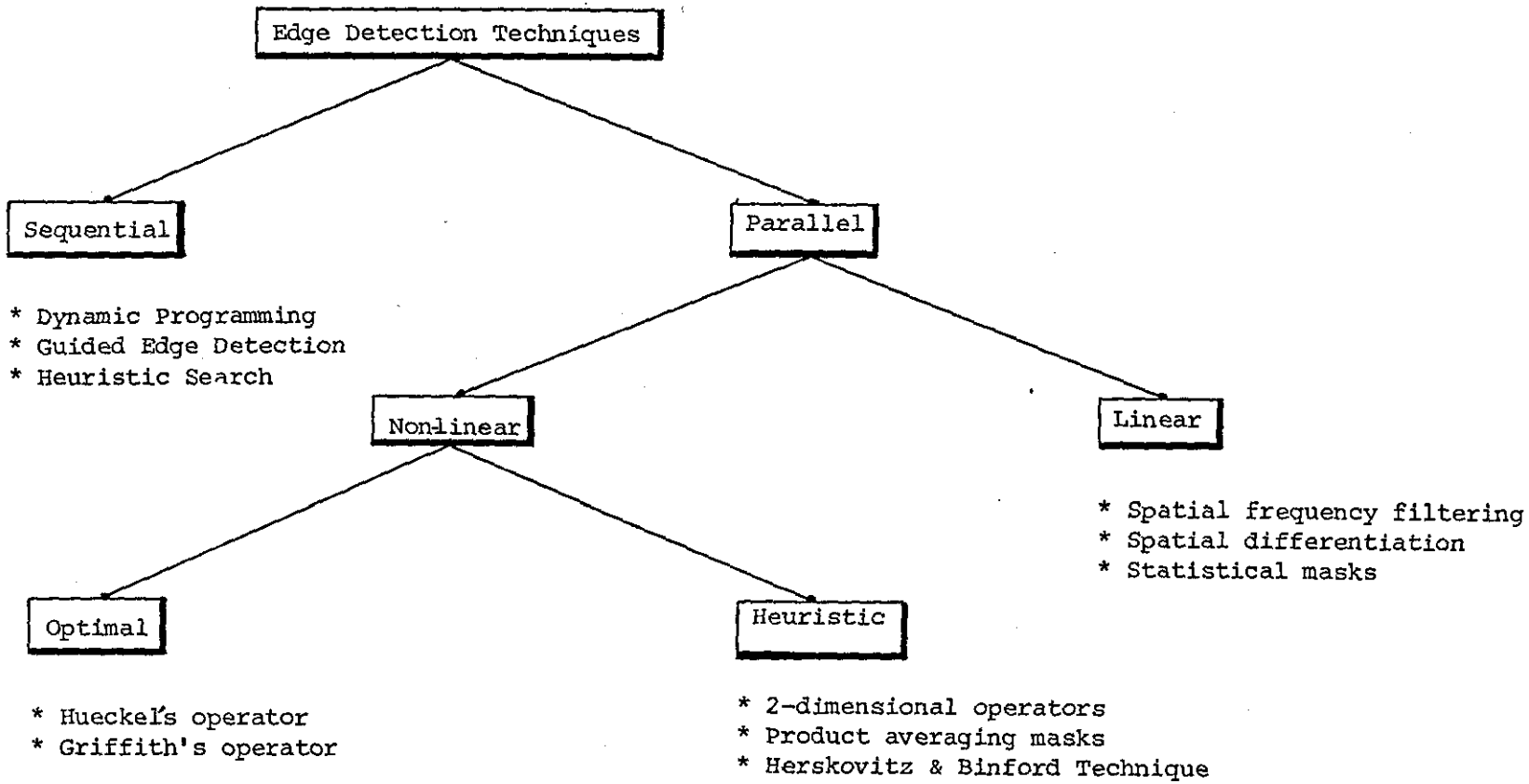


Fig. 3.4: Edge Detection Techniques

Since parallel edge detectors can be applied simultaneously everywhere in the picture, they have a great advantage, concerning speed and flexibility, over sequential edge detectors, particularly when they involve simple mathematical manipulations. The rest of this chapter is devoted to parallel edge detectors. Additional details about sequential edge detectors are available elsewhere[40].

3.2 Linear Edge Detectors

The output image field (G) of a linear operator σ is formed from linear combinations of pixel amplitudes of an input image field (F), i.e. $\sigma:F\rightarrow G$ is a linear operator if

$$\sigma(af + bf') = a\sigma(f) + b\sigma(f'), \text{ where } f, f' \in F$$

In linear edge detection techniques the operation is basically spatial frequency filtering, spatial differentiation, or statistical correlation.

3.2.1 Spatial frequency filtering

When Fourier transform is performed on a digital picture, edges in the picture introduce spatial frequencies along a line in the complex frequency plane orthogonal to the edge[2]. A spatial frequency filter utilises the fact that high spatial frequencies correspond to sharp edges, while low spatial frequencies correspond to the absence of edges, i.e. to regions of approximately uniform grey level. Fourier transform of the output of the filter $G_o(f_x,f_y)$ is given by

$$G_o(f_x,f_y) = G_i(f_x,f_y)H(f_x,f_y)$$

where $G_i(f_x,f_y)$ is the Fourier transform of the input picture function

$g_i(x,y)$, $H(f_x,f_y)$ the transfer function of the filter and f_x,f_y are the spatial frequencies. Using the convolution theorem, the output picture function $g_o(x,y)$ can be expressed as

$$g_o(x,y) = F^{-1}\{G_o(f_x,f_y)\}$$

$$= F^{-1}\{G_i(f_x,f_y)H(f_x,f_y)\}$$

$$= g_i(x,y) * h(x,y)$$

where $h(x,y) = F^{-1}\{H(f_x,f_y)\}$

Here F^{-1} is the inverse transform, $h(x,y)$ the impulse response of the filter, and $g_i(x,y) * h(x,y)$ is the convolution of the two functions $g_i(x,y)$, $h(x,y)$. As the frequency domain operation is a convolution in the space domain the filtering operator is linear and parallel.

Since high spatial frequencies are associated with sharp edges, a highpass filter enhances edges by removing the blurring effect of low frequency components. Noise fills up the very high frequencies and the filter may be designed to attenuate these as well. The basic problem is then to specify the requirement of a spatial filter that cleans up noisy pictures in an optimal fashion. Expressing the picture function $g(x,y)$ as

$$g(x,y) = s(x,y) + n(x,y)$$

where $s(x,y)$ is an ideal picture and $n(x,y)$ is a noise picture, then the problem is to estimate $s(x,y)$ given the actual picture function $g(x,y)$, or in other words to design a spatial filter characterised by

an impulse response $h(x,y)$ that accepts $g(x,y)$ as its input and produces an output $\hat{s}(x,y)$ which is an optimal approximation to the ideal picture $s(x,y)$. In Wiener filter the mean-square-error, $[s(x,y) - \hat{s}(x,y)]^2$ is used as the optimality criterion for evaluating the transform function, which is given by

$$H(f_x,f_y) = \frac{Psg(f_x,f_y)}{Pgg(f_x,f_y)}$$

where $Psg(f_x,f_y)$ is the cross-power spectral density of $s(x,y)$ and $g(x,y)$ and $Pgg(f_x,f_y)$ is the power spectral density of $g(x,y)$

Implementation of a Wiener filter as a two-dimensional infinite impulse response (recursive) digital filter for the extraction of edges is described by Fries and Modestino[41]. The Wiener operator they have employed shows good ability of detecting edge structure while preserving a considerable amount of noise immunity. A measure of its degree of complexity is indicated by the requirement of 22 multiplications and 39 additions per pixel for its implementation.

3.2.2 Spatial differentiation

In terms of the continuous picture function an edge exists at a particular picture point if the spatial derivative at that point is of greater magnitude than derivatives in the same direction taken at adjacent picture points. In a digital context discrete differencing is analogous to spatial differentiation. Edge accentuation in the horizontal direction at picture point (i,j) can be accomplished by performing the difference operation which produces an output image G(j,k) according to the relation[9]

$$G(j,k) = F(j,k) - F(j,k+1)$$

Similarly, in the vertical direction the relation is

$$G(j,k) = F(j,k) - F(j+1,k)$$

Enhancement of edges in diagonal direction can be obtained by subtraction of diagonal pairs of pixels. Edge sharpening can also be accomplished by subtracting slopes of the image amplitude along the same line at both sides of a pixel. For instance in the horizontal direction the operation is

$$G(j,k) = \left[F(j,k) - F(j,k-1)\right] - \left[F(j,k+1) - F(j,k)\right]$$

or equivalently

$$G(j,k) = 2F(j,k) - F(j,k-1) - F(j,k+1)$$

Two-dimensional discrete differentiation can be performed by convolving the original image array with the compass gradient masks such as those shown in Fig. 3.5. The compass names indicate the slope direction of maximum response; for instance the East gradient produces maximum output for horizontal intensity changes from left to right. Since the compass masks have zero weighting factors there is no output response over regions with a constant grey level.

North	North-east	East	South-east
$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$H = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$H = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$H = \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
South	**South-west**	**West**	**North-West**
$H = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$H = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$H = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$

Fig. 3.5: Typical Compass Gradient Masks[9]

Edge enhancement, regardless of the edge direction can be achieved by convolving an image array with a Laplacian mask. Three forms of 3 x 3 Laplacian masks are shown in Fig. 3.6. The first mask is a direct implementation of the difference equation form of the

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

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

(c)

Fig. 3.6: 3 x 3 Laplacian Masks

Laplacian operator[42]:

$$\nabla^2 g(i,j) = g(i,j-1)+g(i,j+1)+g(i-1,j)+g(i+1,j)-4g(i,j)$$

In Laplacian masks the sum of the array elements is zero, and consequently there is no output response over constant luminance regions.

There are various sizes of Laplacian operators, the small ones suitable for abrupt changes in grey level, while the large ones are more appropriate for slow changing luminance values[42]. Fig. 3.7 shows 4 x 4 and 5 x 5 Laplacian operators which are actually enlarged versions of the mask shown in Fig. 3.6(a).

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

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

Fig. 3.7: 4 x 4 and 5 x 5 Laplacian Masks[42]

Persoon[43] has proposed an edge detector implementing a modified definition of the derivative. Basically it is a five pixels by five pixels operator in which the two columns to the left and to the right of the central one are approximated by linear models. The gradient in the right direction is defined as a function of the two linear models, and the average grey level of the two left and right columns. The picture is rotated seven times over 45° and seven additional derivatives are computed for each orientation. An edge is assumed to be perpendicular to the direction of maximum gradient and has as its magnitude the value of the gradient. This edge detector solves some of the problems related to noise, but from a practical point of view it is very expensive in terms of computer time needed.

Wechsler and Kidode[44] have suggested a gradient like operator based on forward, backward and central finite differences of different orders (up to three); noting that as the order increases so does the

size of the masks required to implement the finite differences. The differences are evaluated in the horizontal and vertical directions. Assuming that the picture function F is defined over a rectangular grid, in which $\vec{w} = (j,k)$ stands for pixel location, and $\vec{h} = (h_j, h_k)$ is the interval at which the differences are taken; then the differences are defined as follows;

Forward Difference:

$$\Delta F(\vec{w}) = F(\vec{w} + \vec{h}) - F(\vec{w})$$

Backward Difference:

$$\nabla F(\vec{w}) = F(\vec{w} - \vec{h}) - F(\vec{w})$$

Central Difference:

$$CDF(\vec{w}) = F(\vec{w} + \vec{h}) - F(\vec{w} - \vec{h})$$

The corresponding nth order general formulae are:

$$\Delta^n F(\vec{w}) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} F[\vec{w} + (n - k)\vec{h}]$$

$$\nabla^n F(\vec{w}) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} F[\vec{w} - (n - k)\vec{h}]$$

$$CD^n F(\vec{w}) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} F[\vec{w} + g((n - 2k)/2)\vec{h}]$$

and $$g(\alpha) = \begin{cases} \lceil \alpha \rceil & \text{if } \alpha \geq 0 \\ \lfloor \alpha \rfloor & \text{if } \alpha < 0 \end{cases}$$

where $\lceil \alpha \rceil$ and $\lfloor \alpha \rfloor$ are the ceiling and floor operators, respectively. The ceiling and floor operators are defined as the first integer value above and below their argument, respectively. The finite differences

can digitally be implemented using masks which are weighted in accordance to coefficients of the general formulae.

3.2.3 Statistical masks

Edge sharpening can be accomplished by using statistical mask H, which gives a measure of the statistical correlation of pixel values[9]. H is defined by

$$H = \begin{bmatrix} \rho_c\rho_R & -\rho_c(1+\rho_R^2) & \rho_c\rho_R \\ -\rho_R(1+\rho_c^2) & (1+\rho_c^2)(1+\rho_R^2) & -\rho_R(1+\rho_c^2) \\ \rho_c\rho_R & -\rho_c(1+\rho_R^2) & \rho_c\rho_R \end{bmatrix}$$

where ρ_R and ρ_c represent the assumed Markovian correlation factors between adjacent row and column pixels. If there is no adjacent element correlation, i.e. $\rho_R = \rho_c = 0$, then the statistical mask has no effect. On the other hand if $\rho_R = \rho_c = 1$, the mask reduces to a Laplacian mask, similar to the one shown in Fig. 3.6(c) with the signs of the elements reversed.

Argyle and Macleod[9] have proposed masks with Gaussian-shaped weighting functions. The Argyle function is a split Gaussian function defined in one dimension as

$$h(x) = \exp\left\{-\frac{1}{2}\left(\frac{x}{p}\right)^2\right\} \qquad x \geq 0$$

$$h(x) = -\exp\left\{-\frac{1}{2}\left(\frac{x}{p}\right)^2\right\} \qquad x < 0$$

where p is a spread constant. The Macleod function is given by:

$$H(x,y) = \exp\left\{-\frac{1}{2}\left(\frac{y}{t}\right)^2\right\}\left[\exp\left\{-\frac{1}{2}\left(\frac{x-p}{p}\right)^2\right\} - \exp\left\{-\frac{1}{2}\left(\frac{x+p}{p}\right)^2\right\}\right]$$

where p and t are spread constants. For schemes employing Argyle or Macleod functions, if the mask is centred on an edge then points most likely to indicate the edge will be weighted most heavily.

3.2.4 Remarks

The linear edge detectors previously discussed, with the exception of those employing spatial frequency filtering and Persoon's technique, are sensitive to noise. A common limitation is their amplification of high spatial frequency noise as a result of the inherent differencing operations involved. Noise smoothing can be incorporated into the linear edge sharpening procedure by performing the linear masking on regions of pixels rather than on individual pixels[9]. This can be realised by forming a linear mask H(j,k) which is the result of convolving an edge enhancement mask $H_E(j,k)$ with a lowpass filter averaging mask $H_S(j,k)$, often called noise cleaning mask.

$$H(j,k) = H_S(j,k) * H_E(j,k)$$

Such spatial averaging leads to smoothing of edges as well as noise. Fig. 3.8 shows three noise cleaning masks which are normalised to unit weighting so that the noise cleaning procedure does not introduce a brightness bias in the processed image.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Fig. 3.8: Noise Cleaning Masks

Another method for reducing the effect of noise is to compute the difference of the average grey levels of two neighbourhoods on opposite sides of a point. This method is dealt with in Section 3.3.2.

3.3 Heuristic Non-linear Edge Detectors

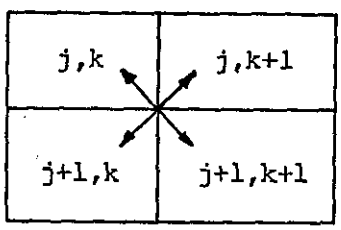
Non-linear edge detectors utilise non-linear combinations of pixels as a means of edge enhancement before thresholding.

3.3.1 Two-dimensional operators[2,9]

Most techniques employ 2 x 2 or 3 x 3 windows to estimate the gradient by evaluating derivatives of the picture function along two orthogonal directions. For instance, Roberts cross operator approximates the magnitude of the gradient at a picture point (i,j) according to the relation

$$|\nabla g(i,j)| \approx G_R(i,k) = ([F(j,k) - F(j+1,k+1)]^2 + [F(j,k+1) - F(j+1,k)]^2)^{\frac{1}{2}}$$

This is equivalent to a two-dimensional differencing operation in which a 2 x 2 window at cell(i,j) has its diagonal elements associated by the subtraction



For reasons of computational expediency, Roberts cross operator is simplified by using the magnitude cross-difference operator,

$$G_A(j,k) = |F(j,k) - F(j+1,k+1)| + |F(j,k+1) - F(j+1,k)|$$

It can easily be shown that

$$G_R(j,k) \leq G_A(j,k) \leq \sqrt{2}\, G_R(j,k)$$

and hence $G_A(j,k)$ behaves qualitatively very much like $G_R(j,k)$.

Sobel has proposed a 3 x 3 non-linear operator. With reference to Fig. 3.9, edge accentuation is obtained by evaluating

$$G(j,k) = \sqrt{X^2 + Y^2}$$

where $$X = (A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6)$$

and $$Y = (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4)$$

A_0	A_1	A_2
A_7	F(j,k)	A_3
A_6	A_5	A_4

Fig. 3.9: Numbering for 3 x 3 Edge Detection Operators

As in the previous case, Sobel's operator is often simplified for computational efficiency by calculating the absolute magnitudes.

$$S(e) = \left|(A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6)\right| + \left|(A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4)\right|$$

Kirsch has suggested a 3 x 3 non-linear operator. Referring to the numbering convention of Fig. 3.9, the operator is based on the relation

$$G(j,k) = \max\left\{1, \max_{i=0}^{7}\left[\left|5S_i - 3T_i\right|\right]\right\}$$

where $S_i = A_i + A_{i+1} + A_{i+2}$

$$T_i = A_{i+3} + A_{i+4} + A_{i+5} + A_{i+6} + A_{i+7}$$

An edge enhancement scheme has been proposed by Wallis, in which an edge exists if the magnitude of the logarithm of the image luminance at a pixel exceeds the magnitude of the average logarithmic luminance of its four nearest neighbours by a fixed threshold value. With reference to Fig. 3.9,the edge enhancement is accomplished by performing the operation

$$G(j,k) = \log[F(j,k)] - \frac{1}{4}\log(A_1) - \frac{1}{4}\log(A_3) - \frac{1}{4}\log(A_5) - \frac{1}{4}\log(A_7)$$

which is equivalent to a Laplacian mask operating on the logarithms of the pixel values. The operation is equivalent to

$$G(j,k) = \frac{1}{4}\log\left[\frac{(F(j,k))^4}{A_1A_3A_5A_7}\right]$$

Comparing G(j,k) against a threshold value is equivalent to comparing the fraction between the brackets against a modified threshold. Therefore logarithms need not be explicitly computed. The principal advantage of this detector is its computational simplicity and insensitivity to multiplicative changes in luminance level.

3.3.2 Product averaging masks

Rosenfeld[45,46,47] has developed a non-linear product averaging mask for edge enhancement, based on using neighbourhoods of many sizes at every point. As a first step,averages of the picture grey level over neighbourhoods of all sizes at every point are obtained. Then for

each size, at each point differences between pairs of averages corresponding to pairs of non-overlapping neighbourhoods just on opposite sides of the point are calculated. If these differences are obtained for more than one orientation, the orientation giving the highest absolute difference is chosen. Edge enhancement is accomplished by evaluating products of these differences. If $E^k_{i,j}$ is the difference of the average grey levels of a pair of $2^k \times 2^k$ neighbourhoods at (i,j), then

$$M(i,j) = \prod_{k=1}^{m} E^k_{i,j}$$

$M(i,j)$ will be large when all $E^k_{i,j}$'s are large. Conceptually, involving large k's in the product ensures the detection of major edges and provides some degree of noise suppression, while the small k's yield localised derivatives which are large when close to the actual edge.

Averaging based on a $2^k \times 2^k$ neighbourhood is quite practical computationally. Expressing the average values $A_k(i,j)$ over a neighbourhood of size $2^k \times 2^k$ at point (i,j) as

$$A_k(i,j) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} F(i,j)$$

Where $F(i,j)$ is the picture grey level, then averaging can be performed by alternatively shifting the picture successively, say, to the right and downward k times and pointwise adding the shifted pictures after each rightward shift and downward shift. The result at each point is the sum of the grey levels in a $2^k \times 2^k$ square having the point as the lower right-hand corner. Intermediate results give the average grey

level for the neighbourhoods of all sizes to the maximum size of $2^k \times 2^k$.

Rosenfeld has also proposed a simpler approach to edge enhancement which, rather than using the same size neighbourhood at each point, is to determine at each point a best size. This is achieved by evaluating the largest size for which the next smaller size does not yield a significantly higher absolute difference. As a result of this; minor edges due to noise near major edges will not be detected, but isolated minor edges from real objects will be detected. Conspicuous edges are determined by erasing the value at a point if there is a higher value at any other point within a distance of half the best size in a direction perpendicular to the best orientation.

3.3.3 Herskovitz and Binford's Technique

Herskovitz and Binford[40] developed a two dimensional edge enhancement technique in which differences, D(x), between two slopes about each point are calculated. This is followed by subtracting the sum of differences over a pair of non-overlapping neighbourhoods on opposite sides of each point.

For the step function shown in Fig. 3.10, the difference of slopes about point x is given by

$$D(x) = [g(x+\Delta) - g(x)] - [g(x) - g(x-\Delta)]$$

$$= -2g(x) + g(x+\Delta) + g(x-\Delta)$$

A plot of D(x) is illustrated in Fig. 3.11(a). Next the function $F_s(x)$ defined by

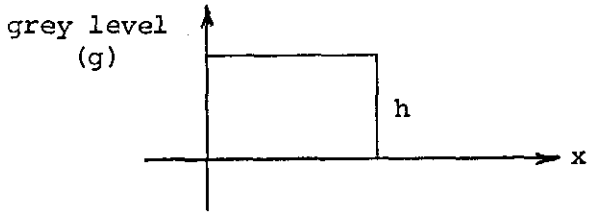


Fig. 3.10: A Step Edge of Height h

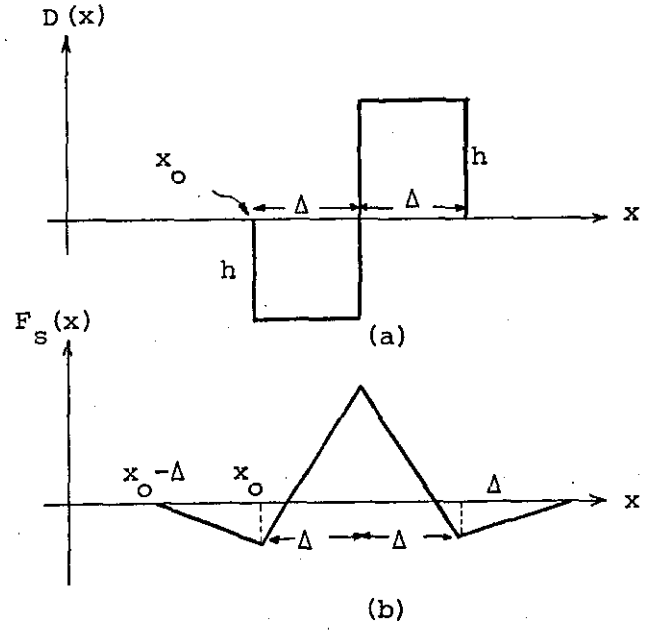


Fig. 3.11 (a) D(x) and

(b) $F_S(x)$ for the Step Function of Fig. 3.10

$$F_s(x) = \sum_{i=1}^{\Delta} S_g(D(x+i)) - \sum_{i=1}^{\Delta} S_g(D(x-i))$$

where $S_g(x) = 1$ if $x>0$, -1 if $x<0$, and 0 if $x = 0$, is computed at every point. This is actually done using two-dimensional neighbourhood, i.e.

$$F_s(x) = \sum_{x \in \text{right neighbourhood}} S_g(D(x)) - \sum_{x \in \text{left neighbourhood}} S_g(D(x))$$

Local maxima of $F_s(x)$ indicates position of an edge. Fig. 3.11(b) shows $F_s(x)$ for the step function of Fig. 3.10.

3.3.4 Remarks

The two-dimensional operators described in Section 3.3.1 involve a small number of pixels and consequently they are extremely sensitive to noise and surface irregularities. One way of reducing the effect of noise is to compute the difference D(j,k) of the average grey levels of two one-dimensional neighbourhoods on opposite sides of a point according to the relation[40]

$$D(j,k) = \left|\frac{1}{\Delta}\sum_{n=1}^{\Delta} g(i,j+n) - \frac{1}{\Delta}\sum_{n=1}^{\Delta} g(i,j-n)\right|$$

$$= \frac{1}{\Delta}\left|\sum_{n=1}^{\Delta}\left[g(i,j+n) - g(i,j-n)\right]\right|$$

Averaging over many points reduces the effect of noise to a great extent. Better results can be obtained by employing two-dimensional neighbourhoods. The size of the neighbourhood is critical. Large

neighbourhoods can overcome the effect of noise, surface irregularities and textural properties of a region and detect major edges, but they may overlook minor edges. Rosenfeld's solution of using neighbourhoods of different sizes overcomes this problem and has proved to some extent effective in finding edges between textured regions.

Herskovitz and Binford's procedure for edge detection is the outcome of their analysis of scenes containing untextured polyhedral objects. The motivation is to detect edges in a particular domain, so there is an in-depth analysis of the sources and characteristics of noise. Still their technique is general and is insensitive to minor edges that are close to major edges.

3.4 Optimal Non-linear Edge Detectors

The previously discussed edge enhancement techniques are heuristic in nature as there is no formal model of 'edge' associated with them and their performance can only be measured not predicted. The following sections deal with optimal approaches to boundary detection based on formal models of edges.

3.4.1 Hueckel's operator

Hueckel's[9] operator uses as input a grey level function derived from a circular area within the image and tries to fit this input to a member within a set of ideal edge lines such that a minimum mean-squared error criterion is satisfied.

Considering the two-dimensional edge shown in Fig. 3.12, the ideal step edge S(x,y) can be defined as

$$S(x,y) = \begin{cases} b & x\cos\theta + y\sin\theta < \rho \\ b+h & x\cos\theta + y\sin\theta \geq \rho \end{cases}$$

where b, b+h are the grey level values at opposite sides of the edge, and θ,ρ represent the polar distance from the centre of the circular region to the normal point of the edge.

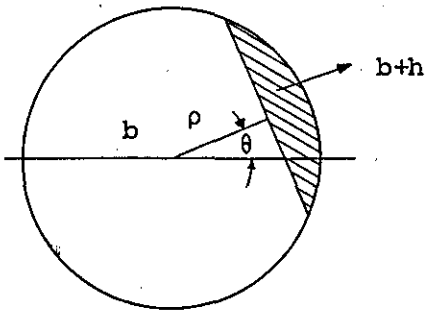


Fig. 3.12: Two-Dimensional Edge Fitting

The edge fitting error ξ is evaluated according to the equation

$$\xi = \int\int_{\substack{\text{over}\\ \text{circle}}} \left[F(x,y) - S(x,y)\right]^2 dx\, dy$$

Hueckel's operator is an efficient solution to the minimisation of ξ. Specifically, he has developed a two-dimensional edge fitting procedure, in which the image points within the circle of Fig. 3.12 are expanded in a set of two-dimensional basis functions by a Fourier series in polar coordinates. Denoting the basis function by $H_i(x,y)$, then the weighting coefficients for the expansion of the image and the

ideal step edge become

$$a_i = \iint H_i(x,y)\ F(x,y)\ dx\ dy$$

$$b_i = \iint H_i(x,y)\ S(x,y)\ dx\ dy$$

minimising ξ is equivalent to minimising

$$\sum_{i=0}^{\infty} (a_i - b_i)^2$$

In Hueckel's algorithm only the first eight coefficients are used for computational economy and to provide some noise smoothing as noise predominates in the high spatial frequencies. The minimisation procedure gives an indication of the best edge and a measure of the goodness of the edge.

3.4.2 Griffith's operator

Griffith[40] has suggested the optimal use of intensity information to detect edges assuming no knowledge of a scene other than the type of edges (step,spike) that is expected. He has considered the input picture function $J(x,y)$ to be the noisy, blurred version of the original picture function $I(x,y)$. The assumption here is that the noise is white noise and is normally distributed about the intensity with a constant variance independent of the intensity. The picture is divided into a set of long narrow rectangular bands and within each band analysis is directed towards estimating $P(J(x,y)\,|\,I(x,y))$, abbreviated for convenience as $P(J|I)$, which is the probability of getting the noise distorted picture function J given the original picture function I. An optimal decision procedure is employed to obtain the probability that an

edge is exactly centred in the rectangular band.

Specifically Griffith has assumed that at most one edge or line exists in every rectangular band and has defined the following functions:-

1. $P(I_i,J_i)$: the joint probability of occurrence of the ith noise-free sample and the jth noisy sample
2. $P(I_i)$, $P(J_i)$: the a priori probabilities of I_i and J_i
3. $P(I_i|J_i)$: the conditional probability of I_i given J_i
4. False Positive Error (FPE) : the decision procedure says there is an edge in a region when in fact there is none.
5. True Negative Error (TNE) : missing an edge that is in the picture
6. $G:\{I\}\rightarrow\{0,1\}$: G maps all I_i with edges or lines down their centres into 1, and all other I_i into 0. G is called an identification function
7. $Q(J)$: the probability that the noisy region J corresponds to a line or an edge

Griffith has proved that Q(J) can be thresholded at a point that will keep the FPE rate constant while minimising the TNE rate. So the problem is to determine a solution for Q(J). Using Bayes' rule he has expressed Q(J) as,

$$Q(J) = \sum_{G(I_i)=1} P(I_i|J)$$

$$= \frac{\sum_{G(I_i)=1} P(J|I_i).P(I_i)}{\sum_{G(I_i)=1} P(J|I_i).P(I_i) + \sum_{G(I_i)=0} P(J|I_i).P(I_i)}$$

By invoking some simplifying assumptions Griffith has found a solution for this equation [40].

3.4.3 Remarks

Although there is no analytical model for the relationship between noise and the performance of Hueckel's operator, yet experimental evidence indicates that the operator performs quite well as an edge detector even for noisy and sometimes textured image fields. The major weakness of Griffith's operator is its tendency to indicate edges where there are supposed to be none. Also, it is not clear whether Griffith's analysis can be extended to objects that are inherently noisy or textured. A common feature of both Hueckel's and Griffith's operators is their computational complexity.

3.5 Performance and Complexity of Edge Detectors

The computational complexity of edge detectors is difficult to define in terms of a quantitative measure, as the cost of an operation depends upon the precision required and the type of operation (addition/subtraction or multiplication/division). Also the relative cost of the operations varies from machine to machine and depends on the level of the programming language. Under all circumstances, amongst the edge detection techniques surveyed in this chapter, the ones with the least computational burden are the linear edge detectors, with the exception of those employing optimal spatial filtering, and the two dimensional non-linear operators. Here the mathematical manipulations entailed are simple particularly when no noise cleaning procedure is involved.

Relatively few studies of edge detectors performance have been reported in literature[8,39,41,48]. Performance evaluation is difficult because of the large number of proposed methods, difficulties in determining the best parameters associated with each technique and the lack of definitive performance criteria. The evaluation of edge detectors is somewhat heuristic as the particular application dictates the seriousness of missed, spurious, or noisy edges.[41] So the visual evaluation, however subjective this may be, remains the main criterion to assess the accuracy of edge detectors.

Qualitative studies of the performance of edge detectors have been reported by Fries and Modestino[41], and Bullock[8,39]. Fries and Modestino have performed an empirical evaluation of the performance of Wiener and Hueckel's operators, using digitised versions of actual scenes and synthetic images with additive white Gaussian noise. As a measure of the complexity of the algorithms they have implemented; the cost of employing Hueckel's operator is 109 operations per pixel, while for the Wiener operator it is 105 operations per pixel. Table 3.1 contains the values they have used to provide a measure of the relative computational costs

operation	cost
16 bit addition or subtraction	1 operation
16 bit multiply	3 operations
31 bit add-subtract	2 operations
31 bit multiply	12 operations
31 bit square root	20 operations

Table 3.1

Assignment of Relative Computational Costs[41]

Concerning performance they have found that the Wiener operator exhibits: relative insensitivity to background noise and/or spurious image detail; ability to discern between soft edges or shading (gradual transitions) and hard edges (step transitions); ability to discriminate between closely spaced and more widely spaced edge structure and a tendency to produce connected edge segments. The Hueckel operator, on the other hand, is found to be quite sensitive to spurious image detail, suffers from the inability to discern soft edges and/or closely spaced edge structure, and tends to produce disconnected edge segments.

Results from Bullock's[8] quantitative comparison of the complexities of some of the edge detectors are reproduced in Table 3.2. Compared to the edge detectors listed in the table, Heuckel's operator has the highest complexity but best performance on low contrast edges.

EDGE OPERATOR	PERFORMANCE RANK	COMPUTATIONAL COMPLEXITY
ROBERTS CROSS	4	N(3a)
HIGHPASS FILTER	4	N(9a)
LAPLACIAN	4	N(9a)
SOBEL	3	N(14a)
KIRSCH	2	N(72a)
HUECKEL	1	54(a+m)=270a

N = Number of Image Elements

a = Machine Add Cycle Time

m = Machine Multiply Cycle Time
(assume $m \approx 4a$)

Table 3.2

Edge Detector Comparison

CHAPTER 4

A New Edge Detection Technique and its Implementation

4.1 Introduction

In solving the problem of edge detection in monochrome video images, the main objective is the implementation of a detection scheme as a hardware preprocessing unit. The rationale for this is to free the microprocessor of this task and to reduce the enormous storage requirement of digitised video signals. This approach calls for the employment of an edge detector having a simple structure and characterised by a relatively small amount of computation to make its realisation feasible through a reasonable amount of hardware.

As may be inferred from the foregoing chapter, excluding the trivial procedure of thresholding a picture function, the available edge detection techniques vary greatly in their degree of complexity. On one hand some detectors are based on the evaluation of the magnitude of the gradient of a picture function, and on the other hand some employ rather complex algorithms. The latter include optimal non-linear edge detectors, Hueckel's and Griffith's operators, Herskovitz and Binford's technique, product averaging masks and spatial frequency filtering. Implementation of a complex operator as a hardware preprocessing unit is a formidable task because of the processing circuitry and size of storage required. Moreover, hard-wired versions which have been proposed for some of the operators are hundreds of times slower than required for real time video analysis[44,49]. The other edge detection techniques are

somewhat easier to implement as hardware units, but here the greatest disadvantage is the requirement to store, at least,part of the video image.

A new edge detection procedure which allows for interactions to take place among spatial differentials at neighbouring pixels has been developed and built as a hardware processor. This procedure utilises the serial nature of the image formation and evaluates differentials at each pixel of the incoming video signal. Simultaneously,comparison functions are derived from differentials at adjacent pixels and a decision criterion is applied to extract true edges. This criterion is guided by intuition and is based on solving logical functions which are formulated using signs of the differentials, the comparison functions, and thresholding operations. Solution of the logical functions involves a thinning operation, an essential procedure when edges are characterised by gradual transitions rather than sharp ones. Thinning consists of retaining only those edge points that reach a local maximum difference.

This chapter opens with a brief description of monochrome video signals and features of digitised video signals which have some bearing on the performance of the edge detector. This is followed by presentation of the edge detection procedure, its hardware implementation and results obtained.

4.2 Analogue Video Signals

Complete analysis of video signals is presented in great detail in many textbooks. In this section a very short and qualitative description of the format of the analogue monochrome video waveform is given.

A TV camera converts a two-dimensional electronic image of a scene into a single dimension time waveforn. This is achieved by using the principle of scanning shown in Fig. 4.1. For simplicity only a few lines of an interlaced pattern are displayed.

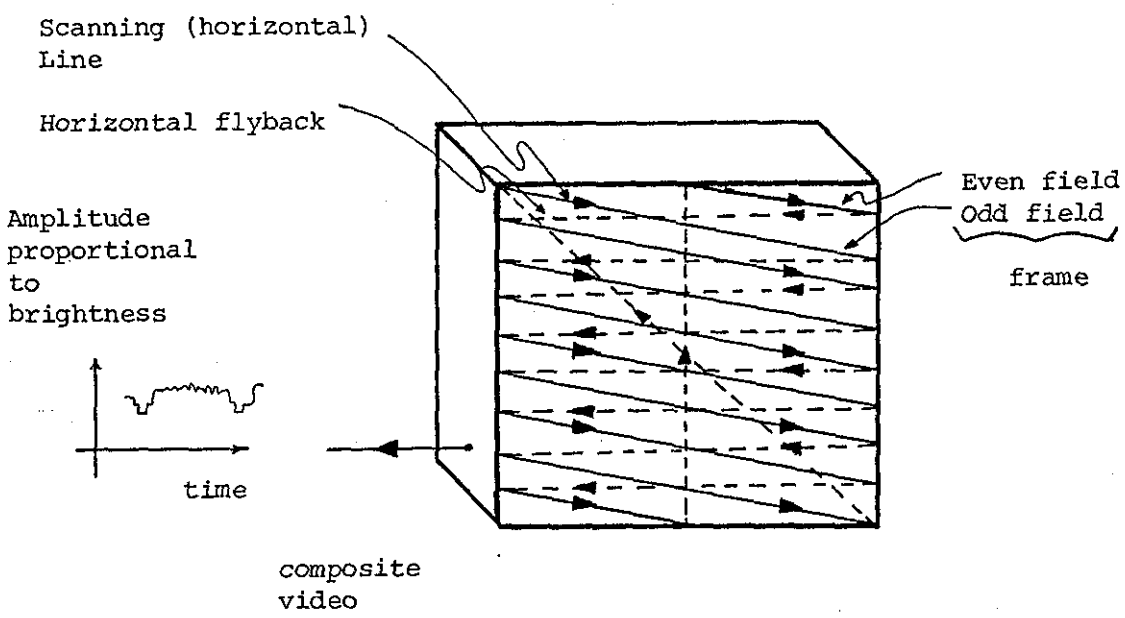
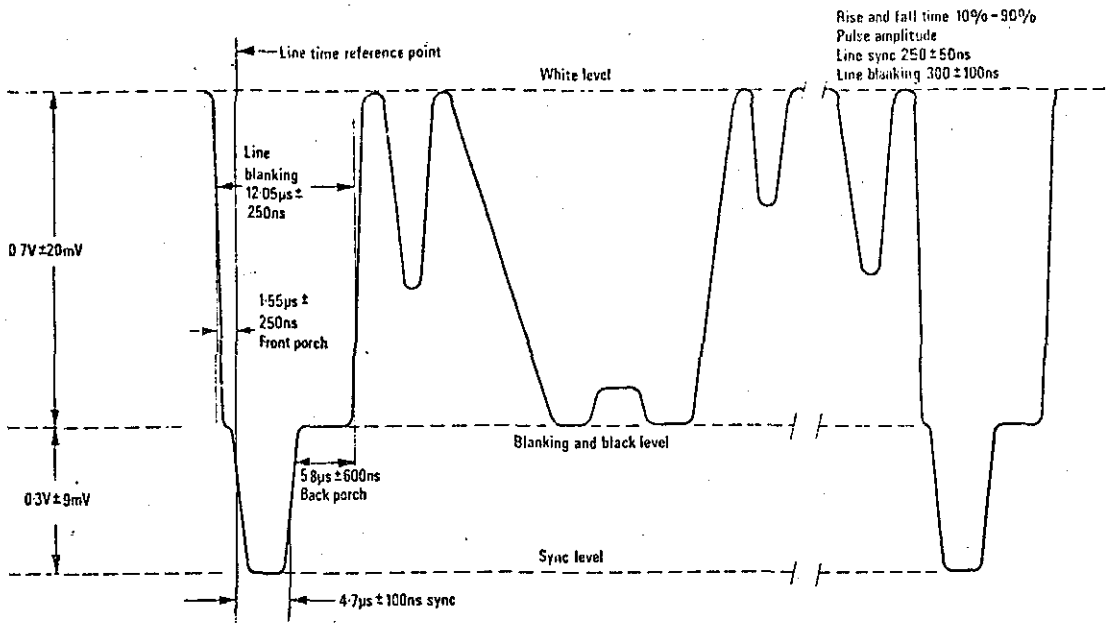


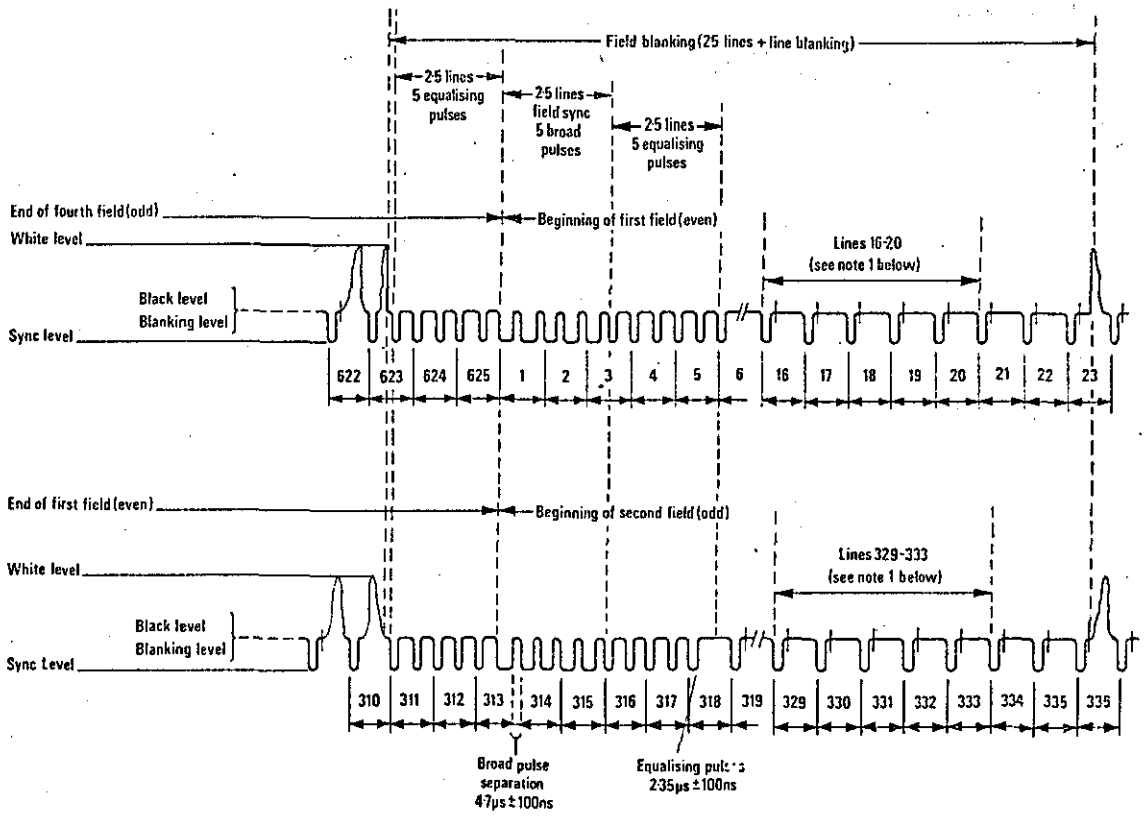
Fig. 4.1: Interlaced Scanning

In the British TV system each frame is represented by 625 lines[50]. The line time base is 15625 lines per second and the field time base is 50 fields per second. The complete video signal waveform is shown in Fig. 4.2. Here the number of lines containing information per frame is 575. The line period is 64 μsec and the line blanking period is 12 μsec. The information content of a line is obtained in 52 μsec. The video signal feeding the edge detector is digitised at the rate of 10.25 MHz* so the resolution of each frame is 533 x 575.

* *Throughout this thesis the 10.25 MHz will be referred to as 10 MHz for brevity.*



(a) The waveform of a typical line showing synchronising signals. Pulse duration is measured at half-amplitude points. Blanking duration is measured at half-amplitude points with a white level signal of line duration and for this reason the picture signal has been shown starting and finishing at the white level.



(b) Vertical synchronising and blanking waveforms for a typical signal. Lines 7–14 and 320–327 have been omitted. Rise and fall times are from 10% to 90% of the pulse amplitude and for the field blanking are 300 ± 100 ns, and for the field sync pulses and the equalising pulses the rise and fall times are ± 50 ns.

Note 1: Lines 16–20 may contain identification control or test signals.

Note 2: The first and second fields are identical with the third and fourth in all respects except burst blanking.

Fig. 4.2: Composite Monochrome Video Waveform (50)

4.3 Digital Video Signals[51-54]

If each sample of a digitised monochrome video signal is represented by a binary number containing 'n' digits, the analogue signal obtained by decoding the digital signal is quantised into 2^n separate levels. The types of picture impairment caused by this quantisation process can conveniently be divided into two:

a) Contouring Effects:

Areas of the picture in which the brightness varies slowly with position are represented by patches of uniform brightness separated by sharp transitions. The appearance of this effect is reminiscent of shaded contours on a map.

Contouring occurs whenever a slowly changing analogue image signal traverses the threshold between two coarsely spaced quantisation levels. The effect is accentuated with TV type images because the contour often has a high degree of spatial and temporal correlation. Contouring tends to be masked by fine patterns which cause the signal to cross quantum levels at closely spaced intervals.

b) Increase in Noise

Random noise on the video signal at the input of the digital coder tends to break up contouring. The quantising errors then appear as additional noise on the picture.

Increasing the number of bits per sample rapidly reduces the above effects as each additional bit halves the quantising error.

If normal subjective standards of broadcast quality are to be maintained then 8 bits per sample are required to produce a picture sufficiently free of quantising noise. 'Dither' signals can be added to the video signal prior to digitising it as a means of reducing the required number of bits per sample. The effect of dither signals is to break up coherent quantising errors such as contouring. However, dither can only save one bit. Fig. 4.3 shows the effect of adding a half sampling frequency dither with an amplitude of $\frac{Q}{2}$ to a video signal, where 'Q' is the increment between successive quantising levels.

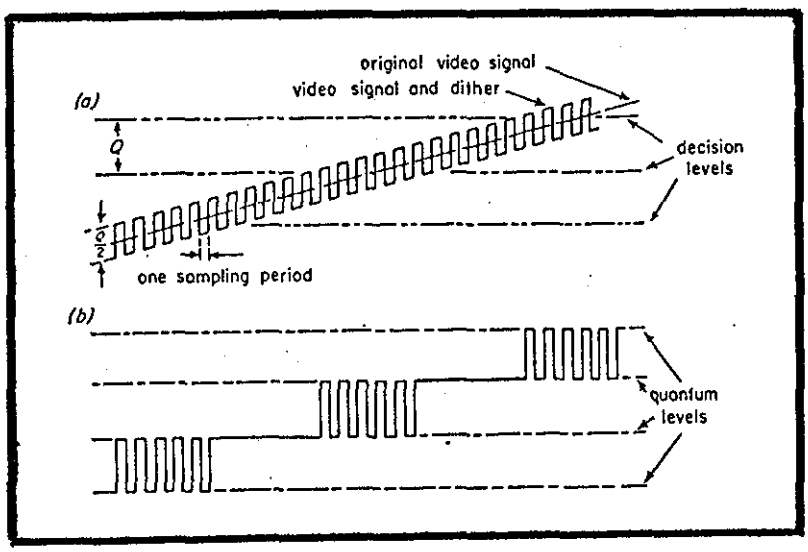


Fig. 4.3: Effect of Adding Half Sampling Frequency Dither to Slowly Changing Video Signal[51]

(a) Relative Positions of Video Signal and Decision Levels in Coder.

(b) Output of Coder.

The edge detector operates on a video signal quantised to 4-bit accuracy which is less than required for acceptable picture quality.

Addition of dither to randomise the selection of quantisation levels may impair the performance of the edge detector because of extra noise superimposed on edges.

In addition to quantising noise, several kinds of noise exist on a digitised video signal. Some of these noises are due to the picture input device. These include random noise of the device, difference in sensitivity between left and right or upper and lower portions of the vidicon camera, blur and/or distortion in the peripheral area. A noisy sensor may generate noise which appears as discrete isolated pixel variations which are not spatially correlated. Pixels that are in error often appear markedly different from their neighbours,"salt-and-pepper-noise". A lowpass filter prior to digitisation acts to remove some of the random noise[9].

4.4 Edge Detection Procedure

First order spatial differentials of a two-dimensional picture function can be obtained by convolving a window of the form

0	-1
-1	2

over the whole area covered by the picture. When this window is overlapping the image array

j-1,k-1	j-1,k
j,k-1	j,k

the horizontal and vertical components of the differential (D_H, D_V) are given by

$$D_H = g(j,k) - g(j,k-1)$$

$$D_V = g(j,k) - g(j-1,k)$$

where g is the picture function.

To perform two-dimensional differentiation the first horizontal line of a video scan must be stored before the commencement of the operation and this line has to be updated as the window scans the picture. Starting from the upper left corner of the scene, the window moves horizontally all the way to the right, encompassing the entire stored first line and the incoming second line. Then it jumps back, sets itself at the left end of the display and moves again towards the right, but this time working on the second line, which has been stored in the place of the first , and the incoming third line. This scan pattern continues until differentials at all pixels representing the picture are evaluated.

To avoid storage of a horizontal line and to simplify processing and control circuitry it has been decided to perform spatial differentiation in the horizontal direction only, i.e. reducing the window to the form

$$\begin{bmatrix} -1 & 1 \end{bmatrix}$$

As the window is convolving the picture, the decision procedure, utilising differentials at 3 adjacent pixels and the relationship between them, is executed to extract edges.

Signs of differentials and their values relative to each other within the neighbourhood of 3 pixels give information about the shape of different edge profiles. These signs are very valuable in determining the behaviour of the edge or more explicitly a positive differential (+) indicates an increase in the grey level i.e. a positive going edge, while a negative differential (-) implies a negative going edge. When there is no edge

the value of the differential is equal to zero (0). The signs (+, -, 0) can be combined to give 18 different edge profiles and a reasonable representation of these profiles is illustrated in Fig. 4.4. In all of these profiles it is assumed that an edge exists at the centre, and as a consequence profiles with 0's at that position are excluded.

The decision procedure is a combination of a number of logical AND functions which are formulated from the following variables:-

1. Signs of derivatives.
2. Variables obtained by comparing the magnitude of the differential at the centre pixel with those at adjacent pixels on both sides of the central one.
3. Result of thresholding the differential at the centre.

Assuming D_l, D_c, D_r represent absolute values of differentials at the left, centre and right pixel respectively, and T is the magnitude of the threshold, then the variables from which the AND functions are formulated are defined as follows:-

b is true if $D_c \geq T$

c is true if $D_c \geq D_r$

d is true if $D_c \geq D_l$

p_1 is true if D_r is positive

p_2 is true if D_c is positive

p_3 is true if D_l is positive

s_1 is true if D_r is negative

s_2 is true if D_c is negative

s_3 is true if D_l is negative

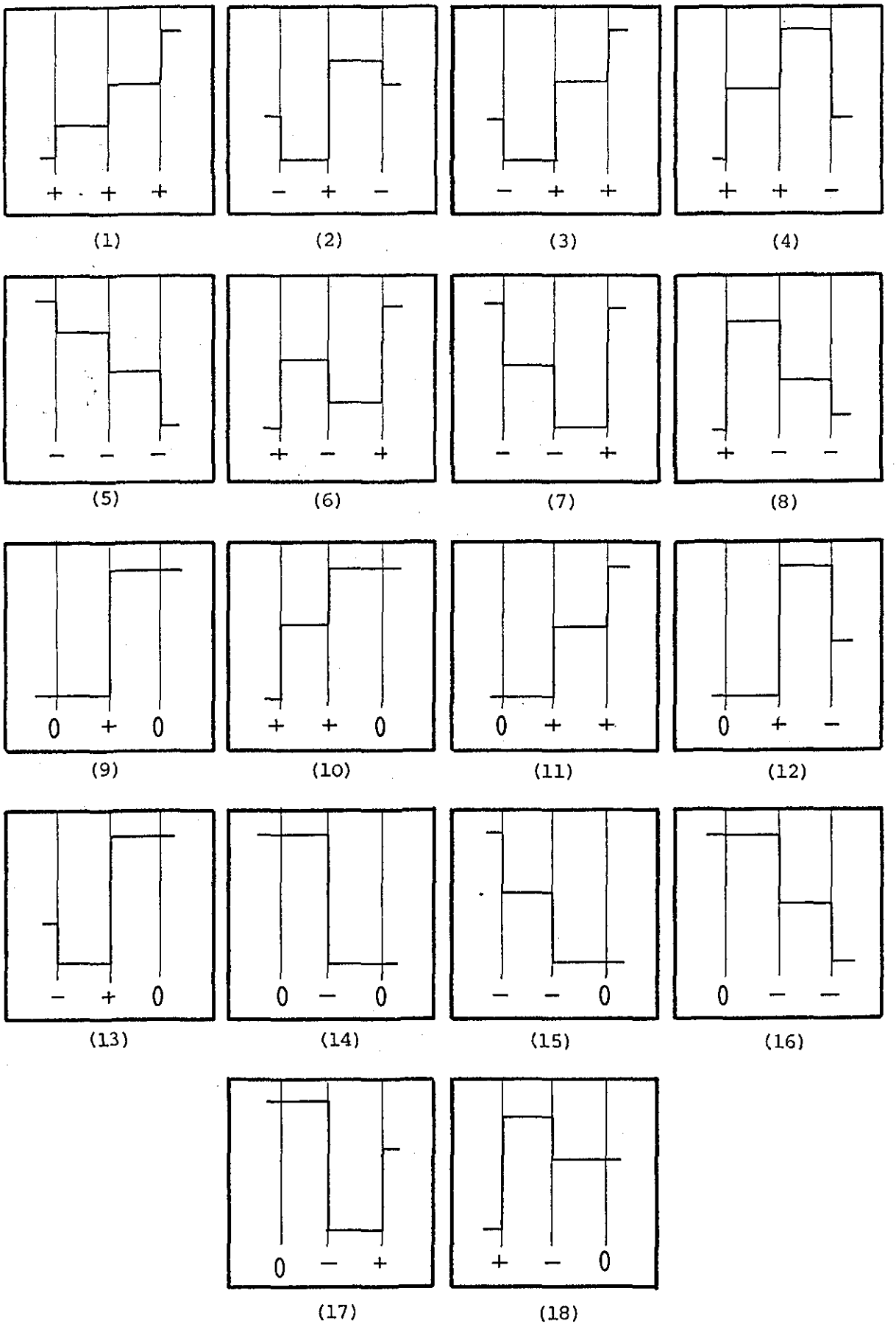


Fig. 4.4: Paradigm Edge Profiles

A positive going edge exists whenever one of the products shown in Table 4.1 is true. Products for negative going edges are given in Table 4.2. In both tables $(\overline{s_1 + p_1})$ and $(\overline{s_3 + p_3})$ are true if differentials at the left and right pixels respectively, are equal to zero.

AND Functions	Signs of Differentials
$b.c.d.p_1.p_2.p_3$	+ + +
$b.s_1.p_2.s_3$	- + -
$b.s_1.p_2.p_3$	- + +
$b.p_1.p_2.s_3$	+ + -
$b.c.d.(\overline{s_1 + p_1}).p_2.(\overline{s_3 + p_3})$	0 + 0
$b.c.d.p_1.p_2.(\overline{s_3 + p_3})$	+ + 0
$b.c.d.(\overline{s_1 + p_1}).p_2.p_3$	0 + +
$b.c.d.(\overline{s_1 + p_1}).p_2.s_3$	0 + -
$b.c.d.s_1.p_2.(\overline{s_3 + p_3})$	- + 0

Table 4.1

AND Functions	Signs of Differentials
$b.c.d.s_1.s_2.s_3$	- - -
$b.p_1.s_2.p_3$	+ - +
$b.p_1.s_2.s_3$	+ - -
$b.s_1.s_2.p_3$	- - +
$b.c.d.(\overline{s_1 + p_1}).s_2.(\overline{s_3 + p_3})$	0 - 0
$b.c.d.s_1.s_2.(\overline{s_3 + p_3})$	- - 0
$b.c.d.(\overline{s_1 + p_1}).s_2.s_3$	0 - -
$b.c.d.(\overline{s_1 + p_1}).s_2.p_3$	0 - +
$b.c.d.p_1.s_2.(\overline{s_3 + p_3})$	+ - 0

Table 4.2

The decision procedure extracts edges and at the same time performs a thinning operation. For instance, when consecutive differentials are all positive and above the threshold value, the edge point obtained lies at the pixel location where the differential is greater than adjacent ones. Thus the system avoids extracting multiple edges when the border line between two regions of two different grey levels is not sharply distinct.

The procedure also takes account of irregularities which might exist in edge profiles as for example when a positive going edge has some negative differentials in its vicinity.

An important feature of edge detectors based on spatial differentials is their insensitivity to reasonable variations in the brightness level of the input video signal. The main drawback of the procedure adopted here is its inability to resolve horizontal edges as a result of evaluating differentials in the horizontal direction only. Because analysis is performed over a limited number of pixels it is sensitive to noise, and it cannot be used for extracting edges between two textural regions, unless differentials at these regions are of reasonably low magnitudes. However, the procedure performs some noise cleaning in the vicinity of true edges by taking into consideration the irregularities superimposed by noise on an edge profile.

4.5 Hardware Realisation of the Edge Detector

A functional block diagram of the edge detector is shown in Fig. 4.5. As data representing a digitised video image (A_0,A_1,A_2,A_3) arrives serially from the analogue-to-digital converter (ADC), it is

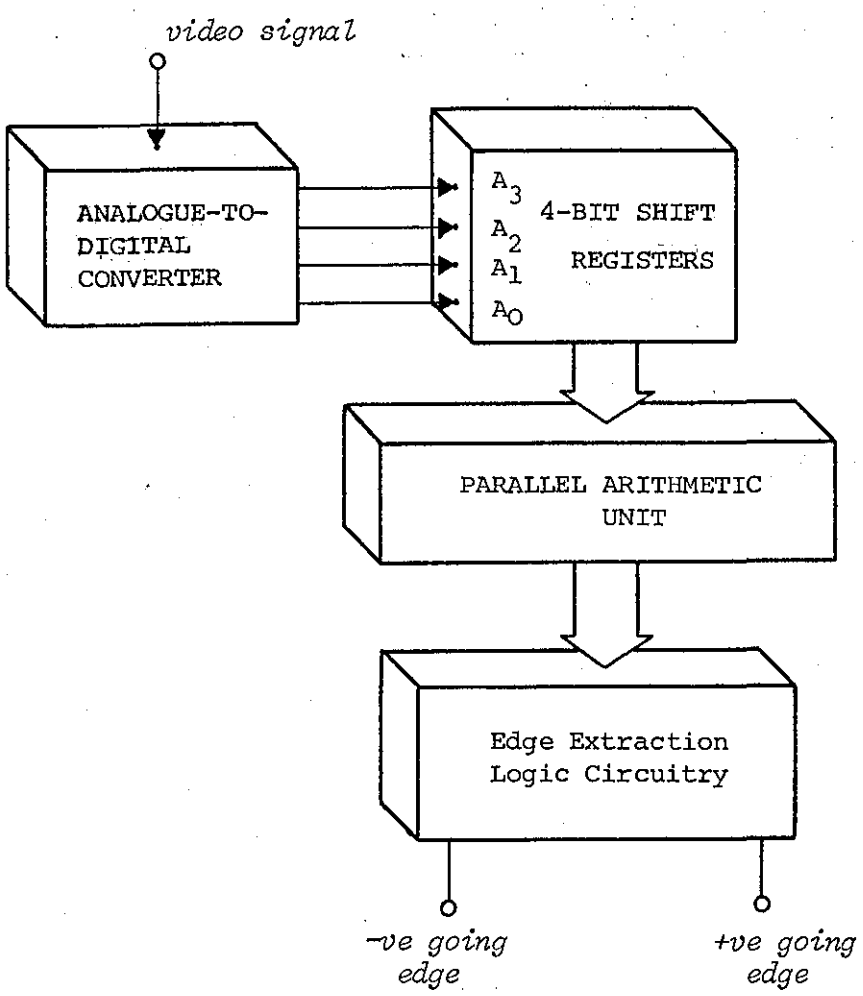


Fig. 4.5: Functional Block Diagram of the Edge Detector

loaded into a bank of 4-bit shift registers. Data stored in the shift registers is accessed by the parallel arithmetic unit, and used to evaluate differentials of the digitised video signal at 3 adjacent pixels, and generate the variables defined in the previous section. The output of the arithmetic unit is relayed to a combinational logic which solves the logical AND functions and extracts edges. All components of the edge detector were built using standard TTL logic circuits. To avoid errors due to various delays imposed

on signals flowing in parallel, the outputs of different stages are latched.

4.5.1 The parallel arithmetic unit

The shift registers are clocked by a 10 MHz master clock to render data representing 4 consecutive pixels available for subsequent parallel arithmetic operation (Fig. 4.6). Data representing the first pixel is denoted by A_{0n1}, A_{1n1}, A_{2n1} and A_{3n1} or more briefly A_{n1}. Similarly the second, third and fourth pixels are denoted by A_{n2}, A_{n3} and A_{n4} respectively. Differentials are evaluated by subtracting the 4-bits characterising a pixel from those of the preceding one. The hardware is designed to generate absolute values of the differentials and the signs attached to each differential separately. This approach is adopted in preference to the popular two's complement method (where the most significant bit is used as a sign bit and the whole operation is performed in one step) for the following reasons:-

1. The input to the arithmetic unit is represented by 4-bit words. To implement the two's complement notation, an extra bit must be included to serve as a sign bit. The arithmetic operations have to be performed in 5-bits which is not economical due to additional hardware required.

2. The logical AND operations make use of the three conditions: two differentials are equal, the first is greater than the second and the first is less than the second. When subtraction is performed in two's complement, the sign bit in the result does not indicate whether the minuend and the subtrahend are equal or not. To determine whether the two are equal the lower

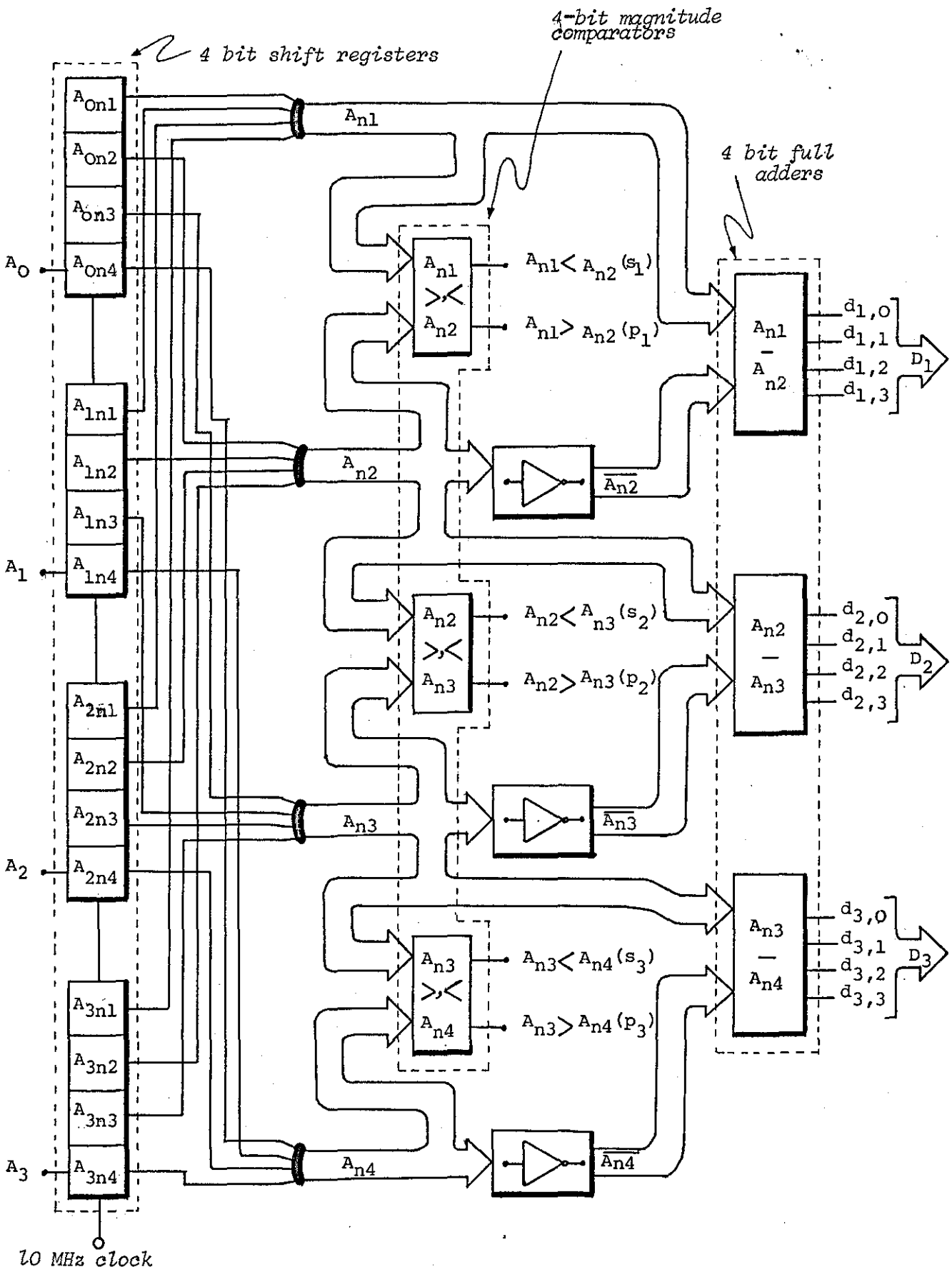


Fig. 4.6: Functional Diagram of the Edge Detector

bits of the result must be examined to determine whether they are equal to zero or not.

3. When operations are performed in two's complement, two thresholds are required, one for negative values and the other for positive values of the differentials.

Referring to Fig. 4.6, the shift registers feed their outputs to 4-bit magnitude comparators, to obtain the signs of the differentials (s_1,s_2,s_3 and p_1,p_2,p_3). Subtraction is accomplished by 1's complement addition. The outputs of the shift registers are connected to 4-bit adders to provide the minuends. The subtrahends are formed by inverting corresponding outputs of the shift registers. Another set of adders and inverters is employed to obtain the magnitudes of the differentials by manipulating the results of the additions (Fig. 4.7). When the minuend is greater than or equal to the subtrahend, the absolute value of the differential is obtained by adding +1 to the sum, whereas if the minuend is less than the subtrahend the magnitude is equivalent to the inverse of the sum.

The parallel arithmetic unit performs both operations on the sum, i.e. addition of +1, and inversion of the result, and relays the two values to the data selectors. Control signals (s_1, s_2 and s_3) are used to select the proper values of the magnitudes. These values, denoted in the diagram by D_R, D_C and D_L are fed to the combinational logic, together with the corresponding signs of the differentials and the variables b,c,d generated by the comparators shown in Fig. 4.8. The switches connected to the comparator which generates the variable b, allow an operator to select different values of the threshold T.

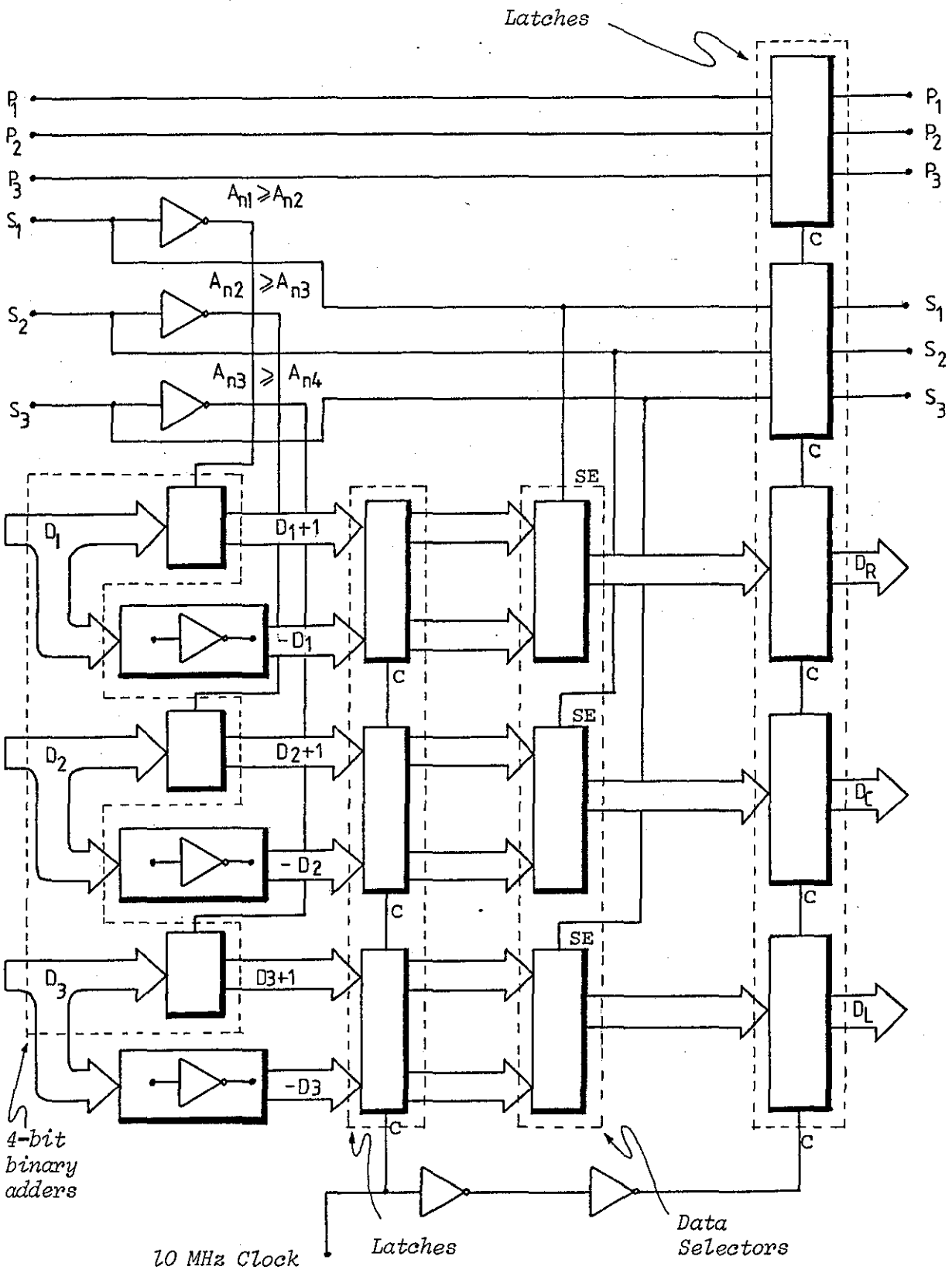


Fig. 4.7: Functional Diagram of the Edge Detector (continued)

4.5.2 The combinational logic unit

Minimisation of the logic circuit has been achieved by reducing the last five products in Tables (1) and (2) by making use of the simplification theorems of Boolean algebra. Considering for instance the last five expressions in Table (1), and assuming that E_p is true whenever a positive going edge exists, then:

$$E_p = bcd\left[(\overline{s_1+p_1})p_2(\overline{s_3+p_3})+p_1p_2(\overline{s_3+p_3})+(\overline{s_1+p_1})p_2p_3+(\overline{s_1+p_1})p_2s_3+s_1p_2(\overline{s_3+p_3})\right]$$

$$= bcd\left[(\overline{s_1+p_1})p_2((\overline{s_3+p_3})+p_3+s_3) + (\overline{s_3+p_3})p_2(s_1+p_1)\right]$$

According to DeMorgan's theorem:

$$\overline{s_3+p_3} = \bar{s}_3.\bar{p}_3$$

$$\therefore E_p = bcd\left[(\overline{s_1+p_1})p_2(\bar{s}_3.\bar{p}_3+p_3+s_3)+(\overline{s_3+p_3})p_2(s_1+p_1)\right]$$

but $\bar{s}_3.\bar{p}_3+p_3 = \bar{s}_3+p_3$

Then

$$E_p = bcd\left[(\overline{s_1+p_1})p_2(\bar{s}_3+p_3+s_3)+(\overline{s_3+p_3})p_2(s_1+p_1)\right]$$

Since $\bar{s}_3+s_3 = 1$, then the expression reduces to

$$E_p = bcd\left[(\overline{s_1+p_1})p_2(p_3+1)+(\overline{s_3+p_3})p_2(s_1+p_1)\right]$$

$$= bcd\left[(\overline{s_1+p_1})p_2 + (\overline{s_3+p_3})p_2(s_1+p_1)\right]$$

$$= bcd\left[p_2((\overline{s_1+p_1}) + (\overline{s_3+p_3}).(s_1+p_1))\right]$$

But $(\overline{s_1+p_1}) + (\overline{s_3+p_3})(s_1+p_1) = (\overline{s_1+p_1}) + (\overline{s_3+p_3})$

$$\therefore \quad E_p = bcd\ p_2\ ((\overline{s_1+p_1}) + (\overline{s_3+p_3}))$$

Similarly, the last five products in Table (2) can be expressed as;

$$E_n = bcd\ s_2\ ((\overline{s_1+p_1}) + (\overline{s_3+p_3}))$$

where E_n is true whenever a negative going edge exists.

The first four products, shown in Table (1) and (2), and the last five expressions, simplified as shown above, are implemented in the logic circuit illustrated in Fig. 4.8. Detected edge elements are fed to monostables to adjust the edge pulse durations to about 100 nsec. the same duration as that of the digitised picture elements. Before sending the edge information for analysis by the microprocessor, pulses indicating positive and negative going edges (B/W, W/B) are combined together by an OR gate.

4.6 Performance of the Edge Detector

When evaluating the edge detector no attempt was made to obtain quantitative measures to indicate its performance. Such measures are somewhat difficult to define because evaluation of edge detector performance is basically heuristic. The particular application generally dictates the seriousness of missed or spurious edges (Section 3.5). For real-world images, the differentiation between true edge structures and spurious details is to some extent arbitrary. Nevertheless, it is expedient to rely on the visual evaluation criterion, however subjective this may be, to assess the accuracy with which the edge detector output represents the essential"edge information".

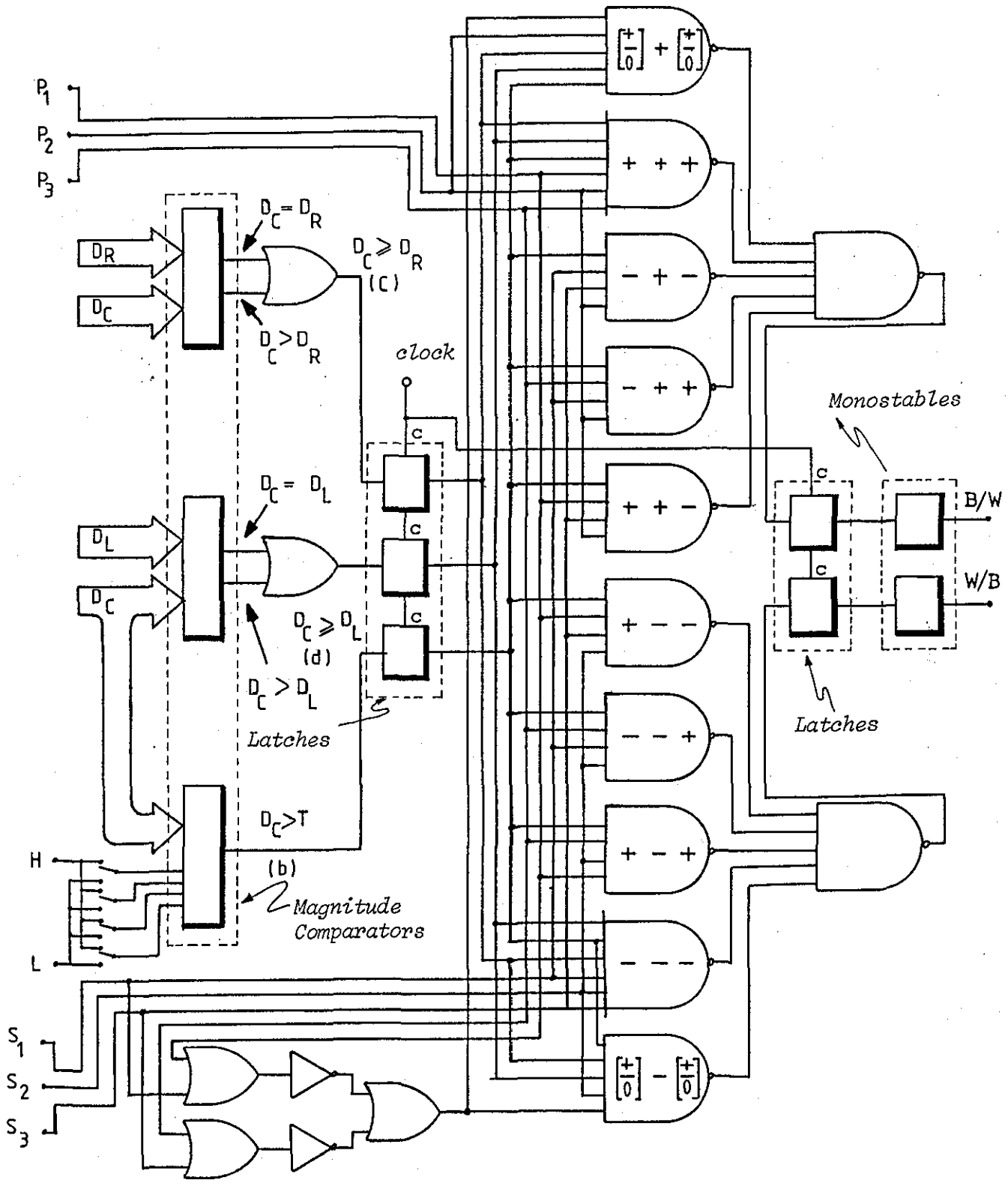


Fig. 4.8: Functional Diagram of the Edge Detector (continued)

Test patterns were provided by a set of pictures and real-life images, viewed by a TV camera feeding the ADC. Experiments were set up in a small space and the lighting conditions were rather poor. In image processing the lighting is an important factor and in general, images viewed by a TV camera should be uniformly illuminated, a condition which was difficult to realise using ordinary ceiling and table lamps. Inadequate lighting and lack of facilities to provide a specially generated set of images containing measured amounts of noise made it very difficult to study quantitatively the effect of noise on the performance of the edge detector. However, all available resources were exploited in the best possible manner to perform extensive subjective evaluation of the edge detector. Input images and respective results, as they appear on a TV monitor, were reproduced on 35mm photographic film and are shown in subsequent diagrams.

The output of the edge detector is dependent on three parameters: the level of the video signal, the level of the reference voltage of the ADC (V_{ref}) and the magnitude of the threshold T. The first two parameters determine the difference between the top and the bottom quantum levels of the ADC, and the subsequent value of the increment (Δ) between successive quantising levels (Fig. 4.9). In addition, they determine the position of the quantising levels on the dynamic range of the video signal.

Since only the information pertaining to the part of the video signal between the top and bottom quantum levels is available for the edge detector, the whole dynamic range of the signal or any smaller part of it can be processed. This is a very useful procedure when all essential information is confined within a small

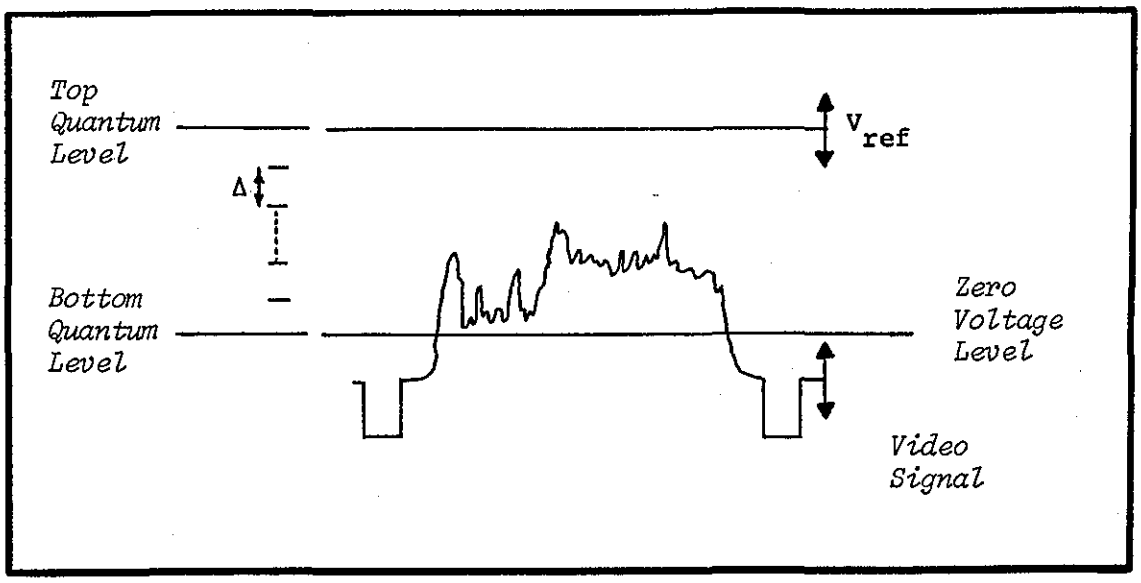


Fig. 4.9

part of the video signal. For example, consider a signal representing a surface with many irregularities as shown in Fig. 4.10 and assume that only the outer edges of the surface are of interest. The levels of the signal and V_{ref} can be manoeuvred such that the top and bottom quantum levels contain the part shown shaded in the diagram. This results in pronounced reduction in the values of differentials within the surface compared to those at its perimeter. As a consequence it becomes a much easier task for the threshold T to discriminate against edges within the surface. Moreover, the value of Δ decreases and this in

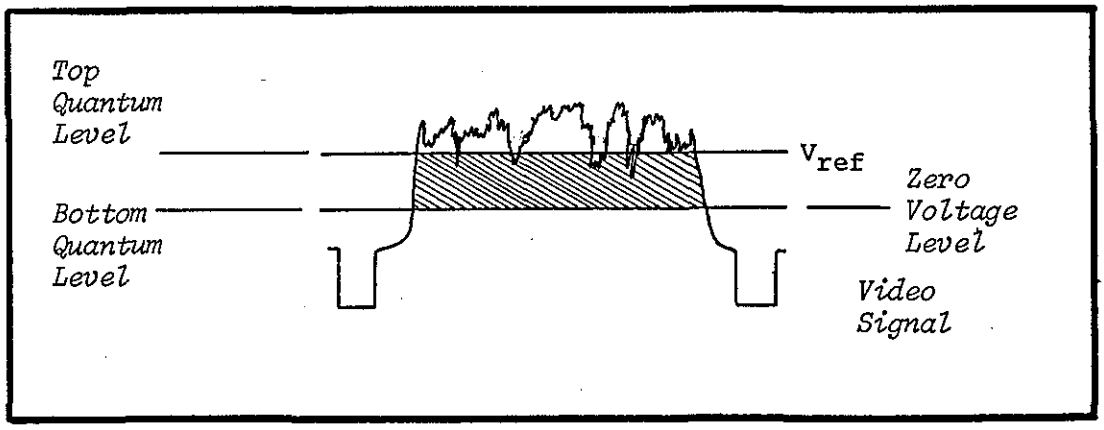


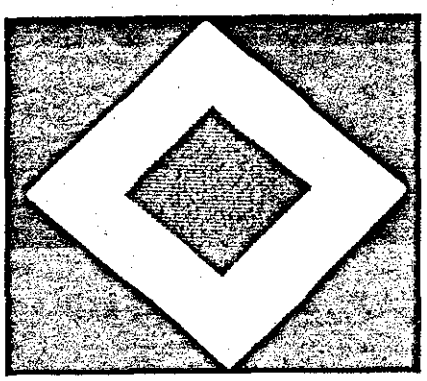
Fig. 4.10

turn leads to reduction in the contouring effect. Adjustments of the voltage levels and the threshold are performed through controls on the front panel of VAS. For each image, an operator has to make proper manipulations to achieve the best subjective performance.

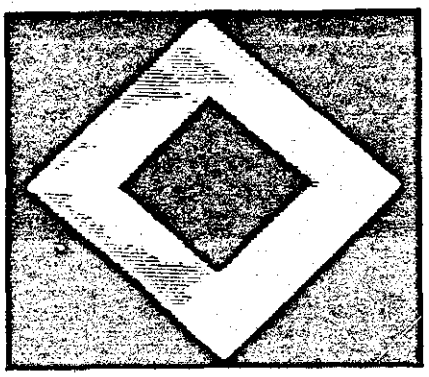
4.6.1 Results

The test pattern shown in Fig. 4.11(a) consists of two coaxial diamonds of different grey levels, viewed against a black background. The contrast between the outer diamond and the background is greater than between the outer and inner diamonds. Plate (b) illustrates the digital image of the pattern, with the dynamic range of the video signal adjusted between the zero voltage level and V_{ref}. Here the shaded contours on the white diamond are due to quantising errors (Section 4.3). Plate (c) depicts the edge pattern with the threshold set to zero. When Plate (c) is overlaid with Plate (b), the edges of the diamonds coincide with the corresponding outlines in the edge pattern. The rest of the edge pattern follows closely the shaded contours but is not, in the strict sense, identical to them. This is because shaded contours drift slowly about a central position and are sensitive to slight variations in the lighting conditions. Light reflected from the operator, which depends on his position in the room, sometimes causes noticeable change. Only when the digital image and the edge pattern are both photographed at the same instant will the shaded contours and their corresponding edges demonstrate exact coincidence.

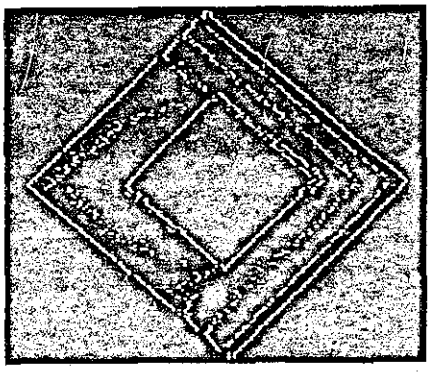
When the threshold is set to 3, most of the edges representing the contours disappear (Plate d). By setting the threshold to 6 the



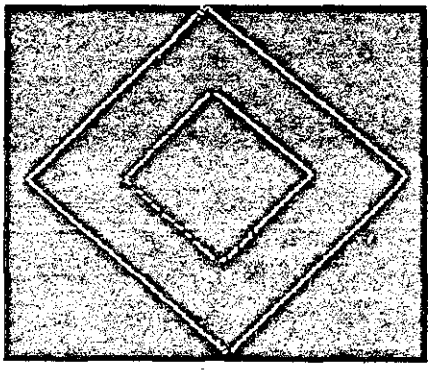
(a) Original Image



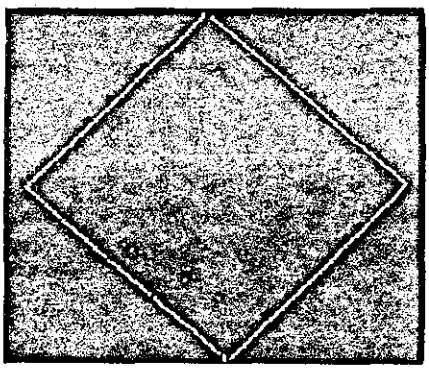
(b) Digital Image



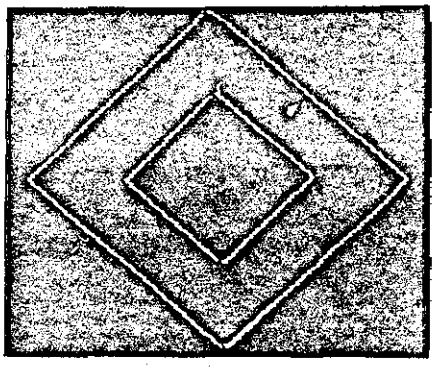
(c) Edge Pattern (T=0)



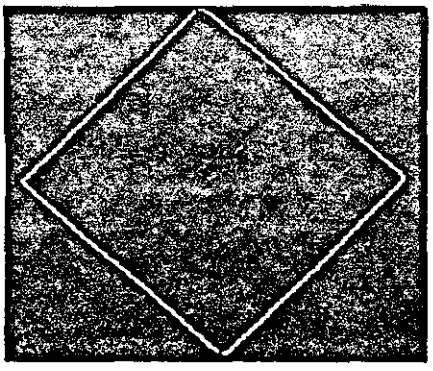
(d) Edge Pattern (T=3)



(e) Edge Pattern (T=6)



(f) Edge Pattern. Video Signal and V_{ref} shifted down (T=4)



(g) Edge Pattern. Video Signal and V_{ref} shifted down (T=8)

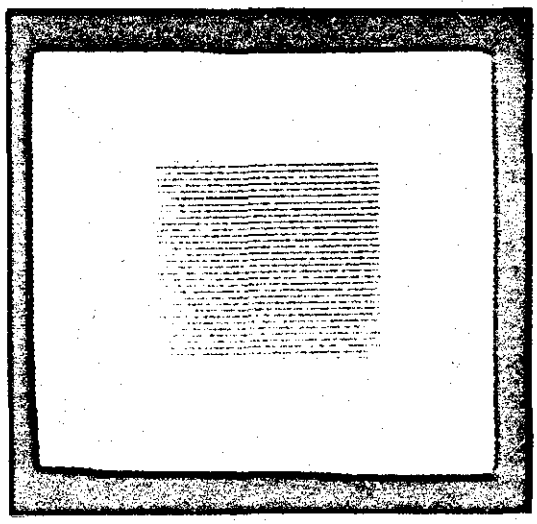
Fig. 4.11

edges of the inner diamond are eliminated (Plate e). This is due to the fact that differentials at the perimeter of the inner diamond are smaller than those at the perimeter of the outer diamond and as the threshold is increased they are excluded first. In this plate spurious points, due to some contouring effect, still exist. By shifting the video signal and V_{ref} downwards and setting the threshold to 4, spurious points are eliminated altogether (Plate f). For the same setting of voltage levels, the edges of the inner diamond can be excluded by increasing the threshold to 8 (Plate g).

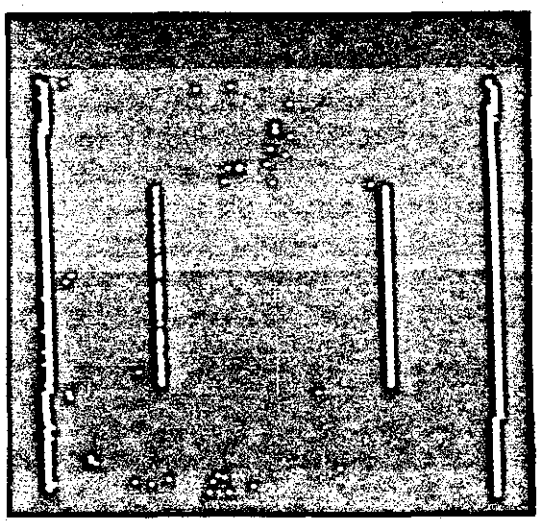
Similar results are shown in Fig. 4.12, but here the inability of the edge detector to discern horizontal edges is demonstrated.

Fig. 4.13 (a) is a photograph of a falconer. Plates (b), (c) and (d) are the corresponding digital images taken for different settings of the brightness of the monitor to enhance various levels of shaded contours. Plate (e) is the edge pattern obtained when the dynamic range of the video signal is adjusted between the zero voltage level and V_{ref}, and T is set to zero. Plate (f) illustrates the ability of the edge detector to extract essential edge information and eliminate spurious points. This is accomplished by shifting slightly the video signal downwards and setting V_{ref} to a low value.

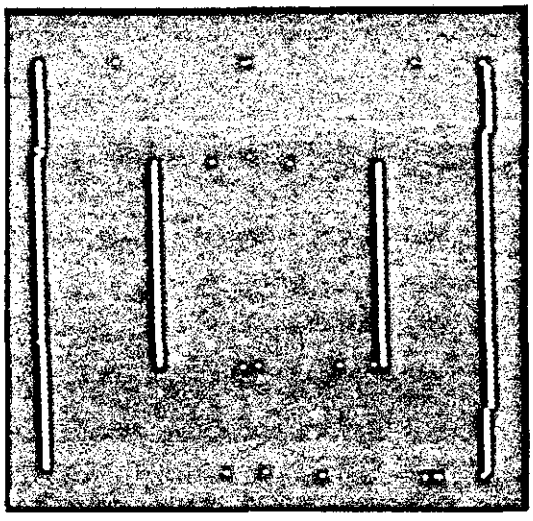
A chair viewed by the TV camera constitutes the test pattern shown in Fig. 4.14(a). It is an example of a 3-dimensional image with noticeable texture characteristics. The edge pattern for T=0 shows many edges on areas, particularly the seat, where texture is conspicuous (Plate c). Again contouring effects have added some distortion to the edge pattern. When T is set to 1 most of the edge points in regions with texture disappear (Plate d). By manipulating the levels of the video signal and V_{ref} a better edge pattern can be obtained as shown



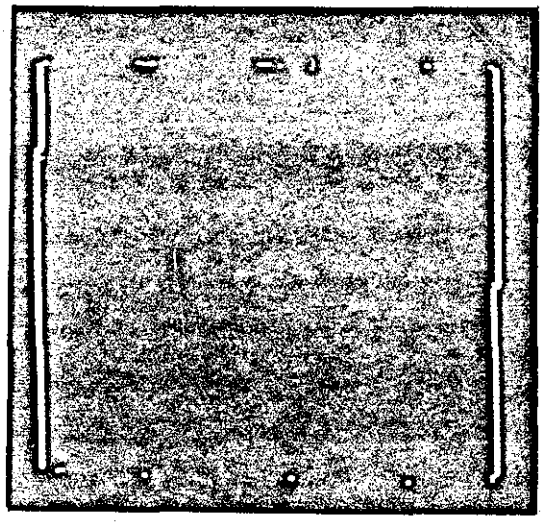
(a) Original Image



(b) Edge Pattern (T=3)

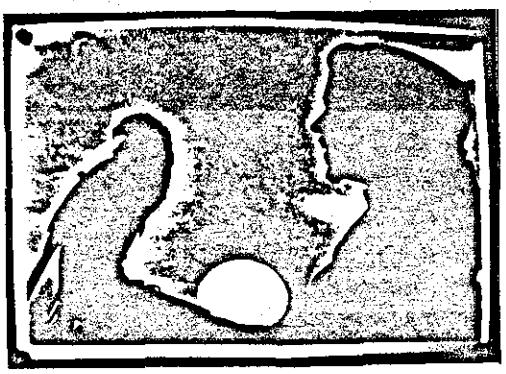


(c) Edge Pattern. Video Signal and V_{ref} shifted down (T=3)

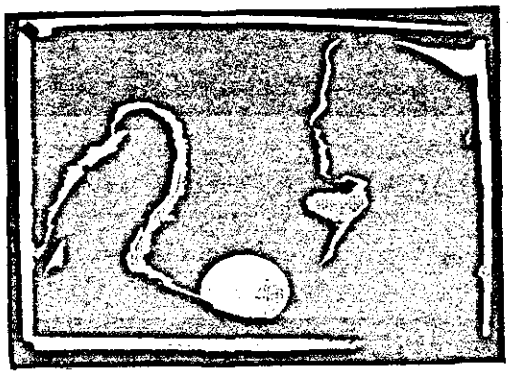


(d) Edge Pattern. Video Signal and V_{ref} shifted down (T=7)

Fig. 4.12

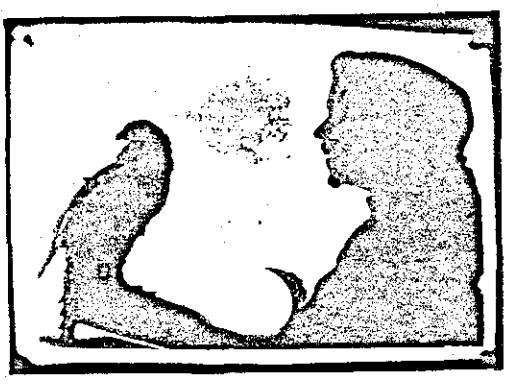


(a) Original Image

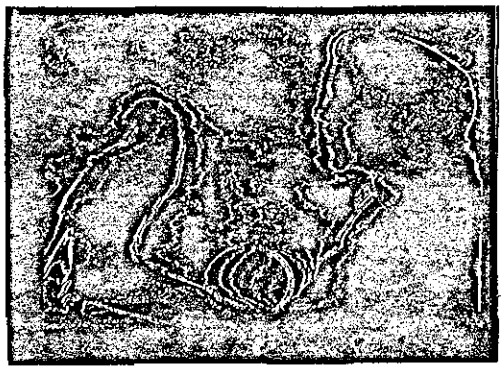


(b) Digital Image

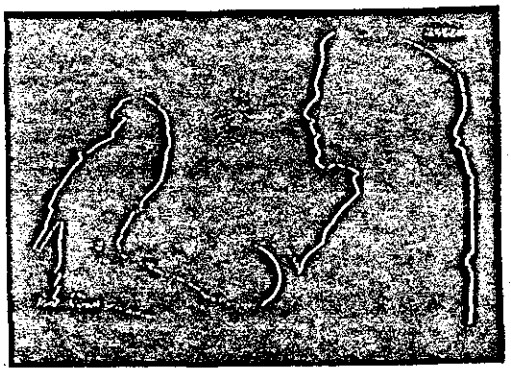
(c) Digital Image



(d) Digital Image

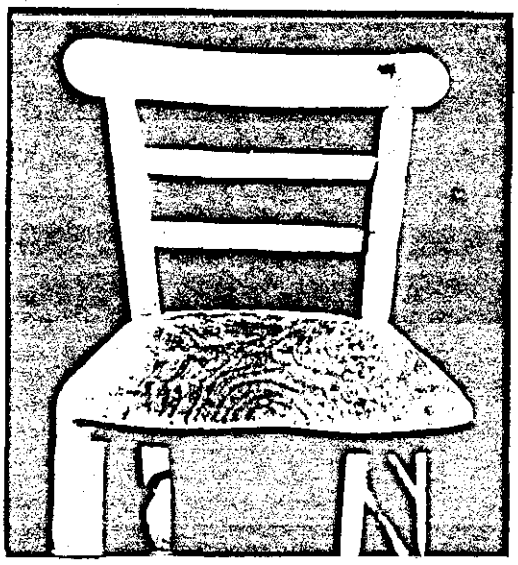


(e) Edge Pattern (T=O)

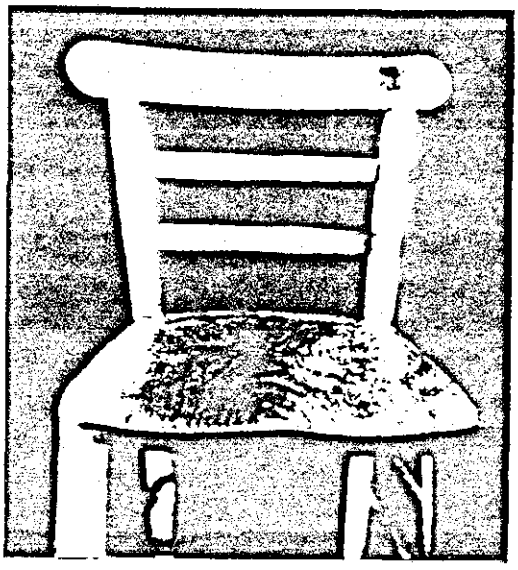


(f) Edge Pattern. Video Signal slightly shifted down. V_{ref} set to a low value

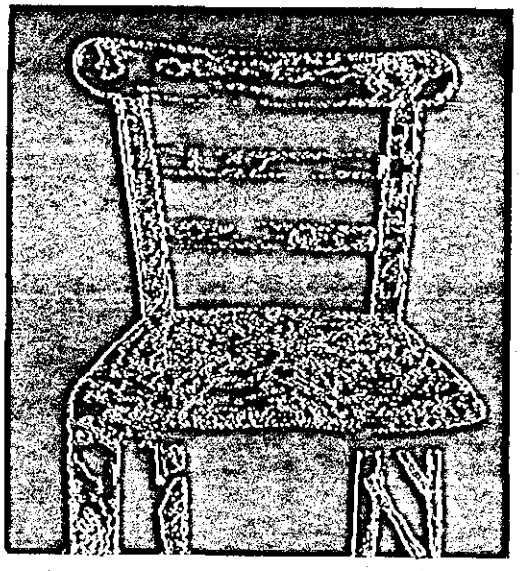
Fig. 4.13



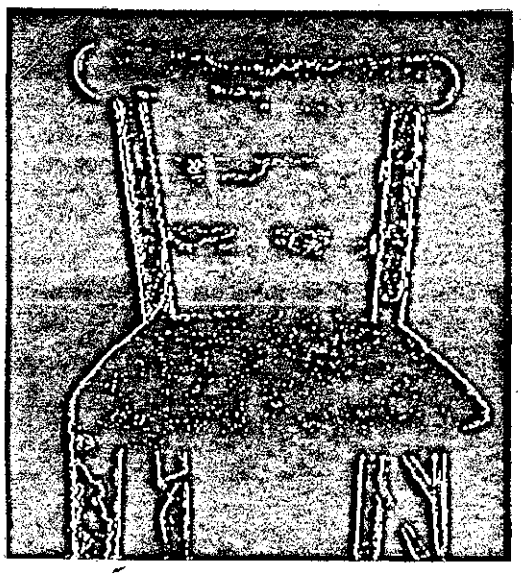
(a) Original Image



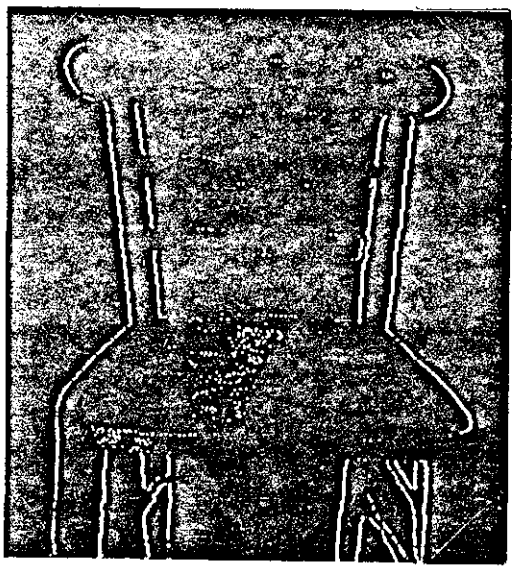
(b) Digital Image



(c) Edge Pattern (T=0)



(d) Edge Pattern (T=1)



(e) Edge Pattern. Video Signal slightly shifted down and V_{ref} set to a low value (T=12)

Fig. 4.14

in Plate(e). Since variations in the grey level within the textural surfaces is less than the variation between the chair as a whole and the background, then proper adjustments of the video signal, V_{ref}, and T makes it possible to exclude texture information from the results.

The input image shown in Fig. 4.15(a) is a photograph of a girl's face. Plates(b)and(c)are the digital images for two different settings of the brightness control of the monitor to enhance different contours. The dynamic range of the video signal is kept between the zero volt level and V_{ref}. Plate(d)illustrates the edge pattern for T=0. Because transitions between different grey levels are not large, most of the edge points disappear when T is set to 1 (Plate e). In Plate(f)the video signal is kept at the same level as before, but V_{ref} is shifted above the peak of the video signal. This causes the contours to spread into black areas, e.g. the hair, which they did not cover before. The remainder of the Plates (g-k) illustrates the edge patterns for different settings of the video signal, V_{ref} and T. In Plate(k) spurious points tend to spread all over the pattern because V_{ref} is at approximately half the dynamic range of the video signal.

Fig. 4.16 shows another set of edge patterns for the test image of the preceding figure. Here the one or two least significant bits of the digitised video signal are set to zero before feeding the signal to the edge detector. Contouring artifacts are conveyed by these lower bits and by grounding them a clearer edge pattern can be obtained, but at the risk of omitting details which might be of value. For instance, in Plate(a)and(b) the same setting of the video signal and V_{ref} is used, but in (a) the least significant bit of the digitised video is grounded, while in (b)the two least significant bits are grounded.

(a) Original Image

(b) Digital Image. Brightness at high level

(c) Digital Image. Brightness at low level

(d) Edge Pattern (T=0)

(e) Edge Pattern (T=1)

(f) Edge Pattern. V_{ref} shifted up (T=0)

(g) Edge Pattern. Video Signal shifted down (T=0)

(h) Edge Pattern. Video Signal shifted down and V_{ref} shifted up (T=0)

(i) Edge Pattern. Video Signal shifted down and V_{ref} at approximately half the dynamic range

(j) Edge Pattern. Video Signal shifted down and V_{ref} set to a low value (T=1)

(k) Edge Pattern. V_{ref} at approximately half the dynamic range (T=3)

Fig. 4.15

(a) Edge Pattern. V_{ref} shifted up (A_3=0)

(b) Edge Pattern. V_{ref} shifted up (A_3=A_2=0)

(c) Edge Pattern. Video Signal shifted down (A_3=0)

Fig. 4.16

Although the outlines depicted in (b) are clearer and easier to perceive than those in (a), the amount of information contained is less. Actually, the same outlines in Plate(b) can be traced in Plate(a). Plate(c) is taken with the same experimental set-up as that of Plate(g) in Fig. 4.15. Similarly, more information is available in the latter plate.

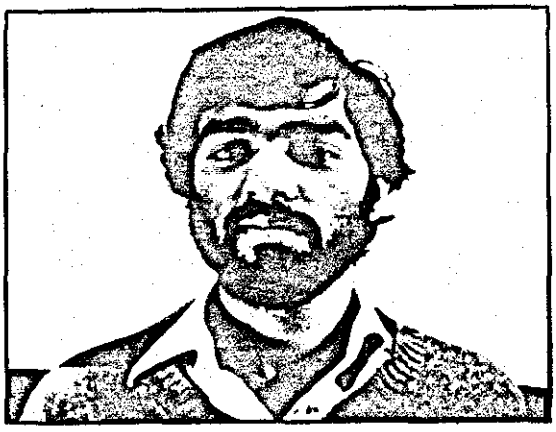
Plates displayed in the subsequent figures (4.17 and 4.18) are images of real-life human faces. In Fig. 4.17 contours appear on the background. This is avoided in Fig. 4.18 by setting a black background. The edge detector worked well and extracted relevant edges of the faces. More details about the eye and nose could have been obtained under appropriate lighting conditions.

4.6.2 Remarks

The results obtained show the ability of the edge detection scheme to extract essential edge information from different types of input images. The outstanding features of the edge detector are that: it does not store any part of the incoming image data, it has a reasonable amount of hardware and is compatible with real time video analysis.

The flexibility afforded by the system is that by changing the level of the video signal and the reference voltage of the ADC, irrelevant parts of the scene can be excluded. In addition, by changing these two signals the quantising levels can be squeezed in small areas of the dynamic range of the video signal, which in effect reduces the contouring artifacts.

The performance of the edge detector can be improved by increasing the number of bits per sample to 8. This eliminates the



(a) Original Image

(b) Edge Pattern

(c) Edge Pattern

(d) Edge Pattern

Fig. 4.17

(a) Original Image

(b) Digital Image

(c) Edge Pattern

(d) Edge Pattern

(e) Edge Pattern

Fig. 4.18

contouring effects and increases the range of the magnitudes of the differentials. Then more edges can be extracted and better selectivity will be accrued to the threshold. The 4-bit ADC feeding the edge detector is a parallel one and has been built using ordinary TTL logic circuitry. The hardware requirement for an 8-bit version of such a converter is prohibitively large (Section 5.5.4). At the same time, during the early stages of the research it was difficult to obtain one of the commercially available 8-bit ADC due to economic considerations.

CHAPTER 5

A MICROPROCESSOR-BASED VIDEO ANALYSIS SYSTEM DESIGN

5.1 Introduction

The next step after building the edge detector is to incorporate it in a system which collects the edge points sequentially, generates their coordinates and allows for their storage and retrieval as well as analysis and re-display in real time. This necessitates the introduction of a processor into the system. One possibility was to interface the edge detector to one of two processors available in the Department of Electronic and Electrical Engineering, University of Loughborough. These processors are a Modular One computer and an Interdata minicomputer. The advent of a single-chip, 16-bit microprocessor offers by far a more convenient alternative. 8-bit microprocessors, which have been on the market since the early seventies[55], are inadequate for handling the coordinates of edge points of a video image as these are expressed in a 10-bit format.

A major advantage of microprocessors is that of cost[56,57]. They are now inexpensive for many applications and their prices are dropping, while their capabilities are increasing. A distinct feature of microprocessors vis-à-vis large processors is that; whereas large processors are shared by many users and tasks, microprocessors are dedicated to one user and/or one task. Here there is a pronounced decrease in the operating system complexity. Also the software required to keep track of a multi-user environment presents a huge overhead cost

on top of the rather formidable hardware cost. With their extensive computing capability, contained in a very few small integrated circuits, it is possible to build a self-contained stand-alone, portable device.

An overview of the evolution of the real time Video Analysis System (VAS) is presented in Fig. 5.1. The heart of the system is a TMS9900 microprocessor, manufactured by Texas Instruments Incorporated. Two other 16-bit microprocessors were available on the market when options for interfacing the edge detector to a processor were considered in early 1977. These were the National Semiconductor's INS8900 (PACE) and the Data General's mN601 (MicroNova). Both are characterised by limited instruction sets and multiplexed bus structures. On the other hand TMS9900 has full minicomputer instruction set capability including multiply and divide,advanced memory-to-memory architecture; and separate memory, input/output, and interrupt-bus structures rendering it easy to interface with peripherals.

The following guidelines were placed within the context of the overall design process:

a - cost effectiveness,

b - simplicity and portability,

c - modularity and expandability,

d - adequate software tools ,

e - fast display of results,

f - feedback between operator and system,

g - system image resources available to be previewed by the user.

This chapter is devoted to the rudimentary building blocks of VAS, and whenever it is deemed essential the details are presented. Detailed hardware description and circuitry are postponed to the Appendix.

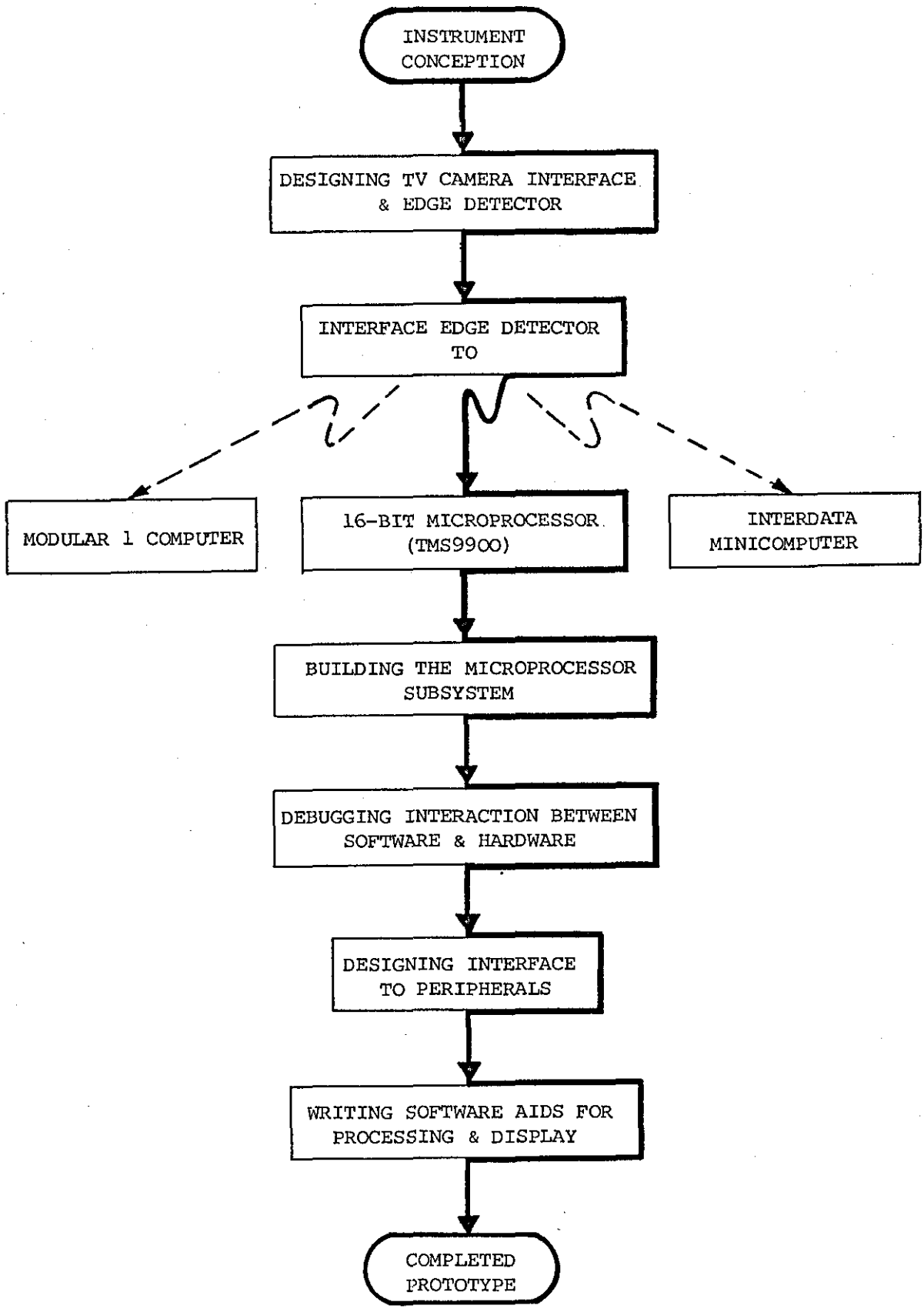


Fig. 5.1: Overview of the Evolution of VAS

5.2 Basic Modules

The basic modules of which VAS is composed and the interconnections between them are shown in Fig. 5.2. Information transfers occur between the system modules and within the system modules. In addition to the TMS9900 , the microprocessor system comprises 1K words of ROM containing the system software, 9K words of RAM, and input/output (I/O) interface circuitry. The system software consists of a debug monitor and an assembler derived from the Technico Inc. TEC-9900-SS single board, microprocessor development system[58]. The edge detector is interfaced to a TV camera (PYE Super Lynx, Type LD M0001). Output of the edge detector is routed to the microprocessor system via the coordinate generator and interface. The window generator is used to blank the whole raster except for a rectangular area whose size and location can be varied by the operator. Monitor-1 is directly connected to the TV camera. Monitor-2 displays digitised video or edge patterns. Stored data and results are displayed on a storage oscilloscope.

Some modules are constructed from P.C. boards, but most of them are built using wire-wrap technique. The modules and the system power supply are assembled in one unit shown in Fig. 5.3. The data terminal used in conjunction with VAS is a Texas Instruments' "Silent 700 ASR" which is provided with a cassette magnetic tape unit for storage of programs and data.

A functional diagram of VAS is shown in Fig. 5.4. Here the dotted lines enclose the units from which the modules shown in Fig. 5.2 are constructed. In the subsequent sections a more detailed description of each module is presented.

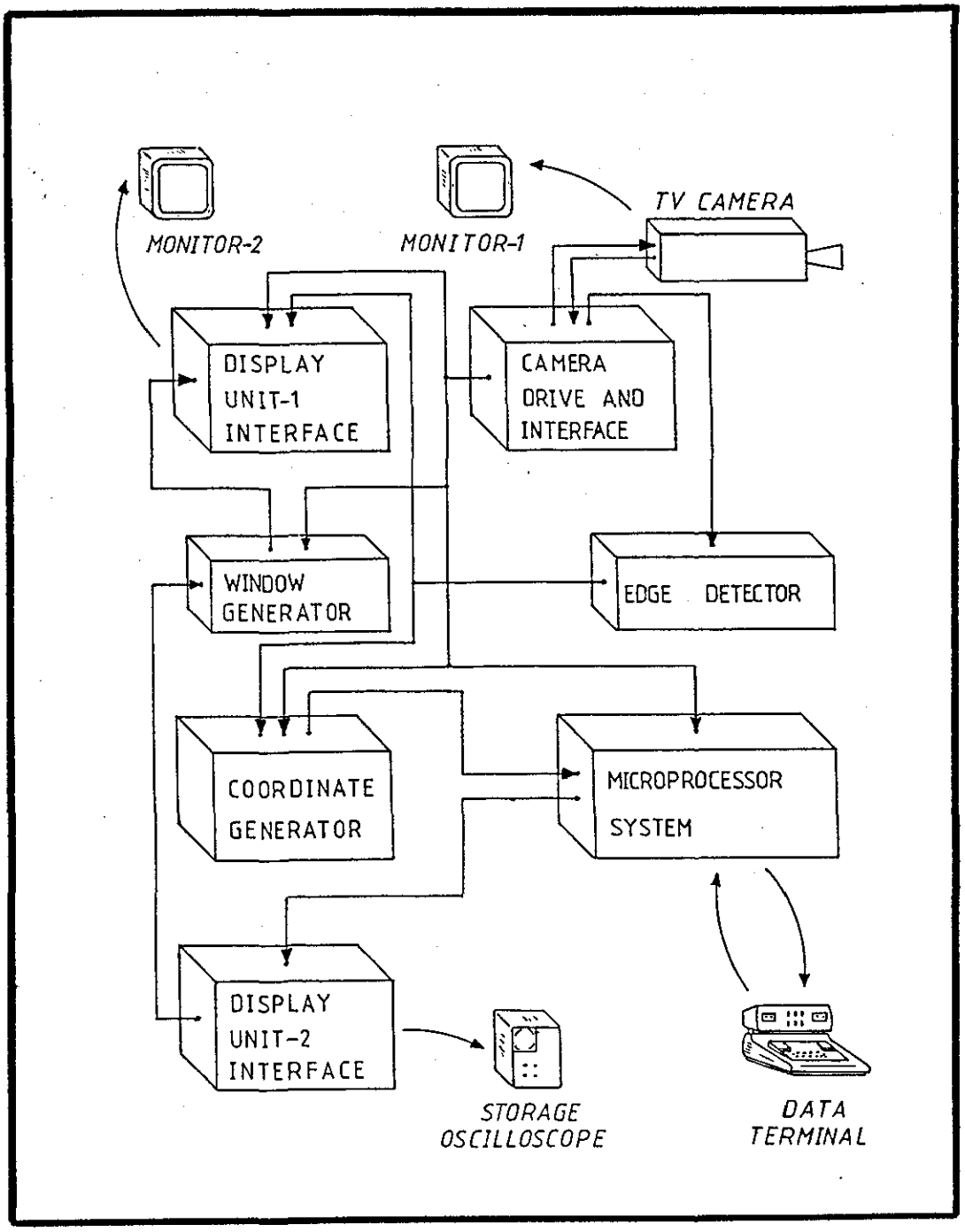
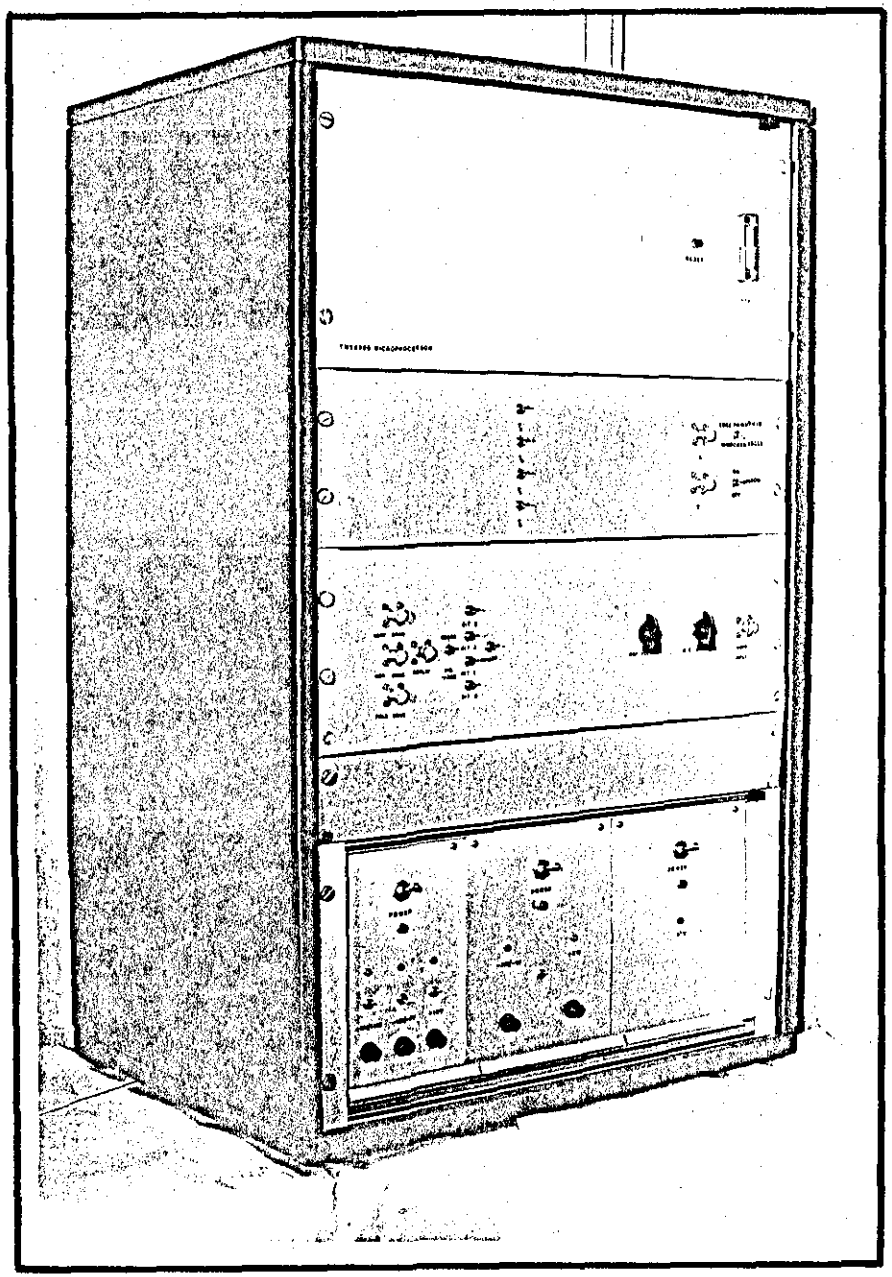
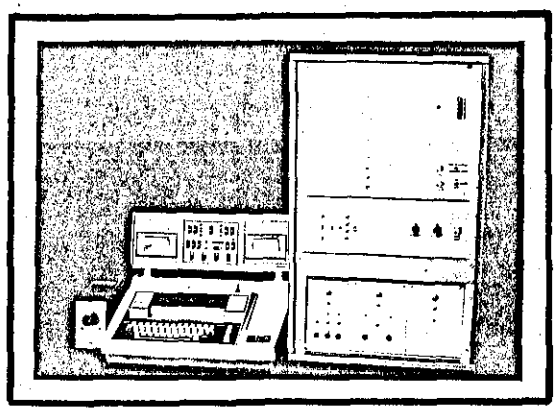


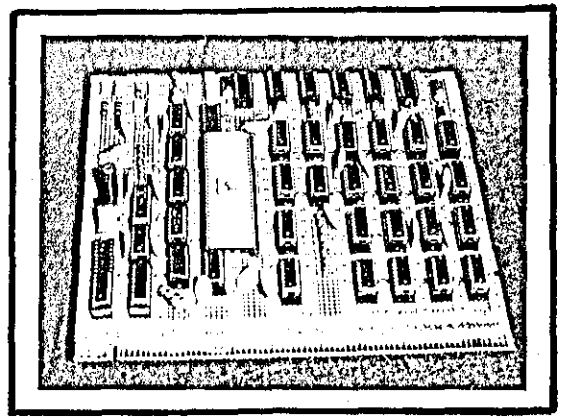
Fig. 5.2: Block Diagram of System Configuration



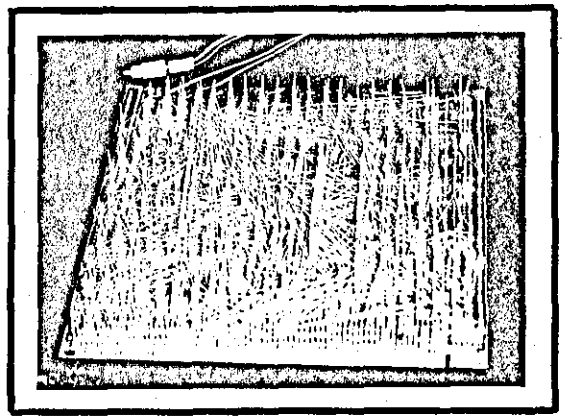
(a) Front View of VAS



(b) The System



(c) The Microprocessor Board (Top View)



(d) The Microprocessor Board (Underside)

Fig. 5.3

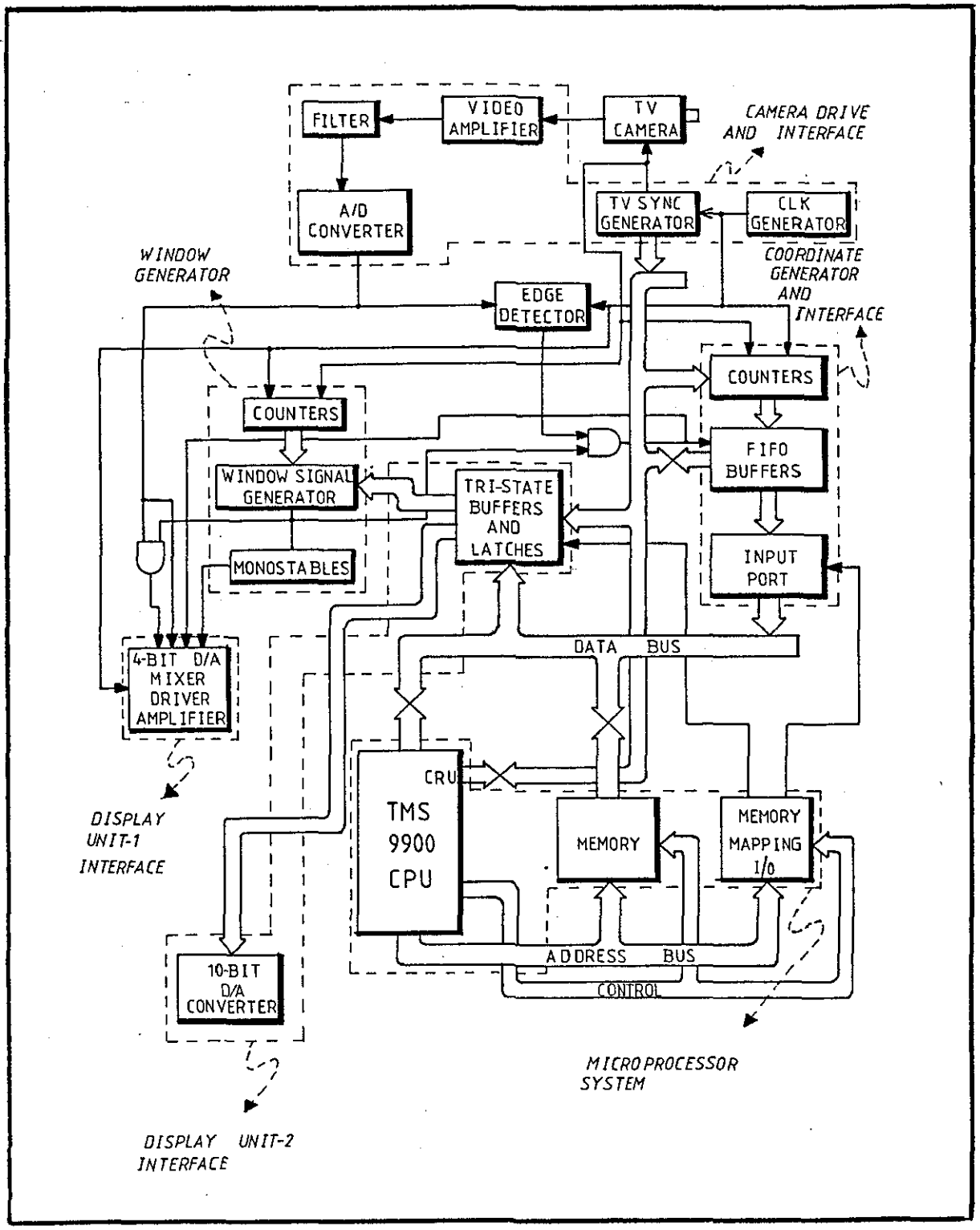


Fig. 5.4: Functional Diagram of VAS

5.3 Description of TMS9900[59-62]

The details given in this section represent a small fraction of the data taken from the manufacturer's literature. However, the information presented is sufficient for a reader to obtain an understanding of the design of the TMS9900 microprocessor system.

5.3.1 Introduction

The TMS9900 microprocessor is a 16-bit central processing unit (CPU) mounted within a 64-pin IC, and is produced using N-channel silicon gate MOS technology. It is capable of working with 16-bit instruction words on a 16-bit data bus. In addition it has the following key features:-

* Full minicomputer instruction set (69 commands including multiplication and division)
* Up to 65,536 bytes of memory (32,768 words)
* 3 MHz speed
* Advanced memory-to-memory architecture
* Separate memory, I/O, and interrupt-bus structures
* Sixteen general registers
* Sixteen prioritized interrupts
* Programmed and DMA I/O capability

5.3.2 Word organisation

Fig. 5.5 shows the relationship of 16-bit memory words and byte operands. Each of the 16-bit memory words can also be defined as two bytes of 8 bits each. The instruction set for the TMS9900 allows both word and byte operands. Thus, all memory locations are

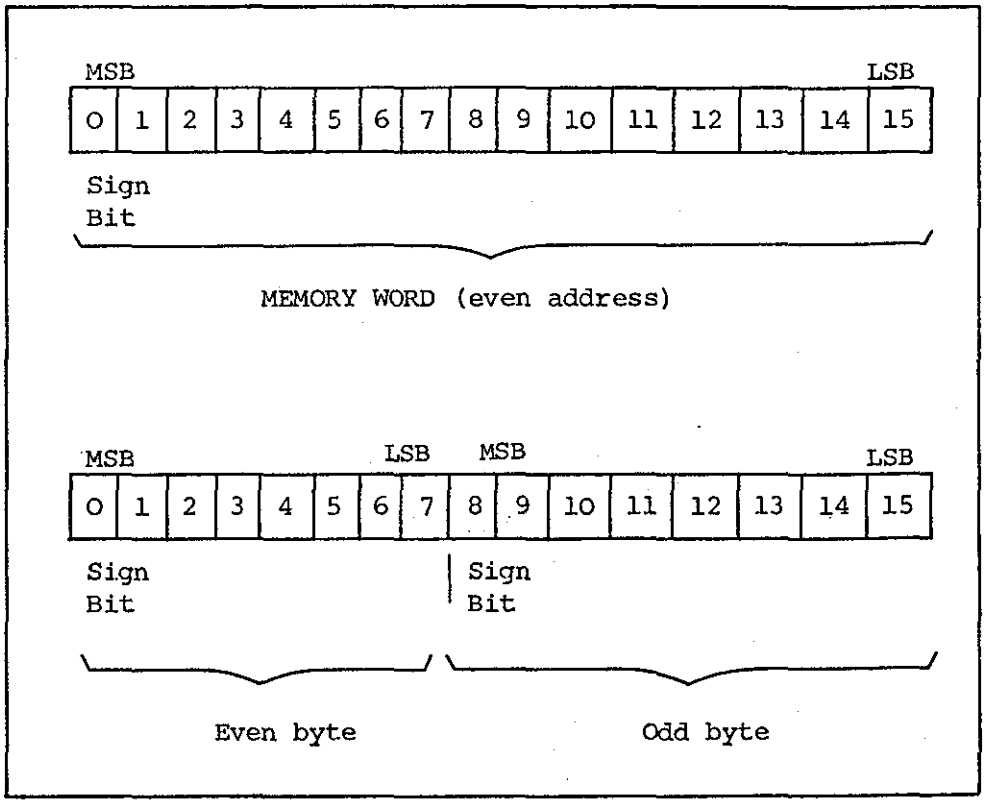


Fig. 5.5: Relationship of 16-bit Memory Words and Byte Operands

on even-address boundaries, and byte instructions can address either the even or the odd byte.

5.3.3 Registers

Blocks of memory, designed as workspace, replace internal hardware registers with program-data registers. A workspace-register file occupies 16 contiguous memory words in the general memory area and its location is defined by a single internal register called the Workspace Pointer. When a different set of registers is required the program simply reloads the workspace pointer with the address of the new workspace.

The workspace concept is particularly valuable during operations that require a context switch, which is a change from one program environment into another (as in the case of an interrupt) or to a subroutine.

$\overline{\text{RESET}}$, and $\overline{\text{LOAD}}$ also cause a context switch. The purpose of the context switch is twofold. Firstly, the present internal registers(Workspace Pointer(WP), Program Counter (PC) and Status Register (ST)) are stored in memory; secondly, new values for the WP and PC are loaded; thus setting up a different workspace and starting point for program execution.

5.3.4 Interrupts

The TMS9900 uses 16 interrupt levels, with the highest priority level 0 and the lowest level 15. Level 0 is reserved for the $\overline{\text{RESET}}$ function, and all other levels may be used for external devices.

5.3.5 Input/output

Three I/O methods are implemented in the TMS9900: memory-mapped I/O, DMA I/O and Communication Register Unit (CRU) which is a versatile direct command-driven interface for serial data transfer. The memory-mapped I/O and the DMA are similar in operation to those used in other microprocessor systems, but the CRU transfer is unique to the TMS9900, and deserves a closer examination.

Texas Instruments reasoned that board layout and parallel-data-bus complexity could be reduced if some sort of serial data transfer capability was incorporated into the microprocessor[63]. In addition to being able to read or write a bit-stream of data, Texas Instruments wanted its microprocessor to be able to selectively manipulate or test the bits sent or received.

Up to 4096 directly addressable input bits and 4096 directly addressable output bits are provided by the CRU. Both input and output bits can be addressed individually or in fields of from 1 to 16 bits. The TMS9900 uses three dedicated I/O pins (CRUIN , CRUOUT and

CRUCLK) and 12 bits (A_3 through A_{14}) of the address bus to interface with the CRU system. The processor instructions that drive the CRU interface can set, reset or test any bit in the CRU array or move between memory and CRU data fields.

5.3.6 Bus structures

The TMS9900 has separate memory, I/O, and interrupt bus structures as shown in Fig. 5.6. The uniform width memory words are transferred in parallel between the CPU and memory. The memory reference instructions operate on parallel data words or bytes, with additional masking instructions to isolate individual bits.

5.3.7 Memory

Standard memories can be interfaced easily without external address latches. The 16-bit address word describes a 64K x 8 bit address space. The least significant address bit is used internally by the CPU to select the even or odd byte, and the other 15 address bits are passed to external memory to describe a 32 x 16 bit address space.

Besides register files the system memory is used for program and data storage. For maximum design flexibility only the locations of the transfer vectors are restricted. A memory map for the TMS9900 is shown in Fig. 5.7.

5.3.8 Memory control

The TMS9900 uses three signals to control the use of the data bus and address bus during memory read or write cycles. These are: Memory Enable ($\overline{\text{MEMEN}}$), Data Bus In (DBIN) and Write Enable ($\overline{\text{WE}}$). A basic

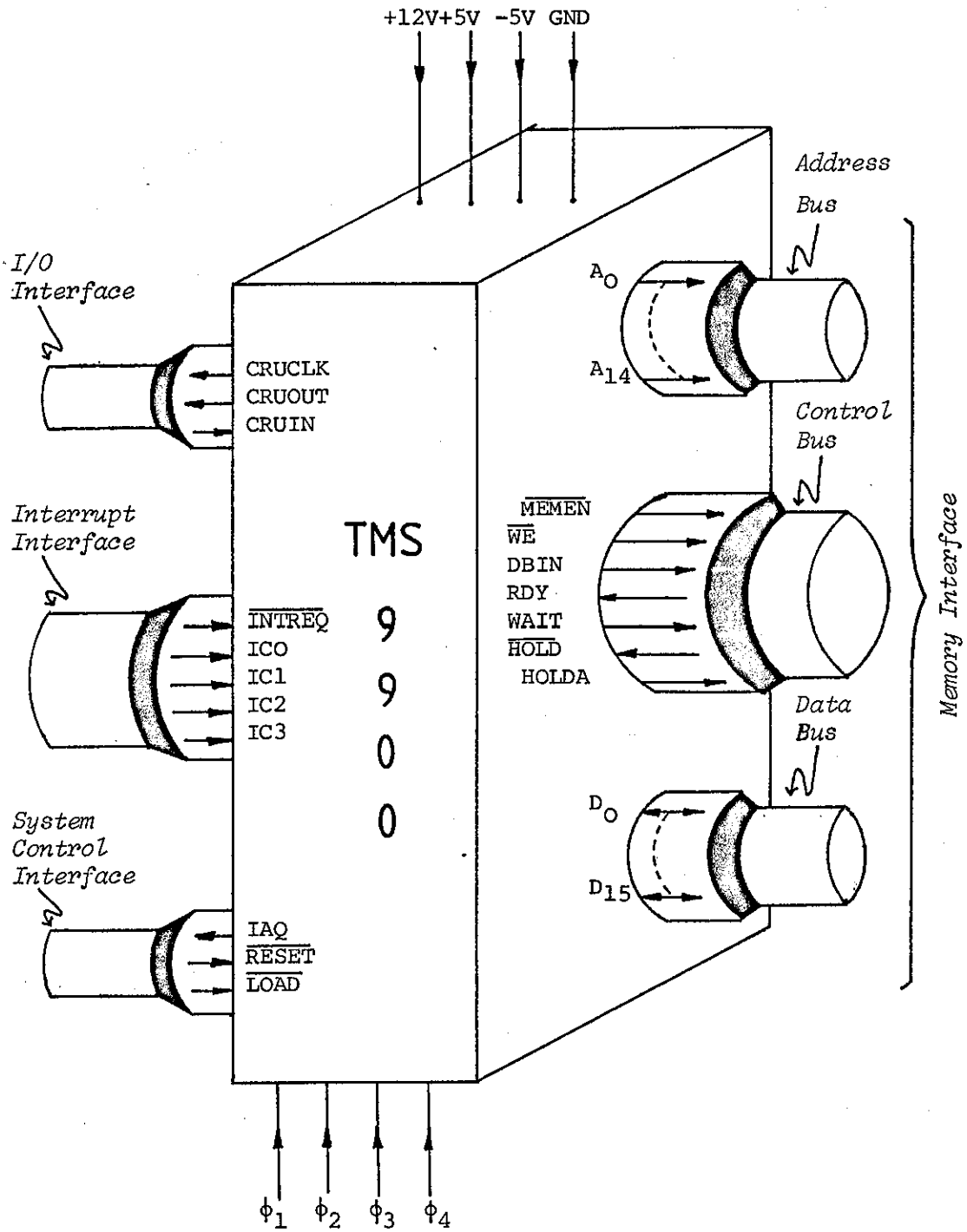


Fig. 5.6: TMS9900 Bus Structure

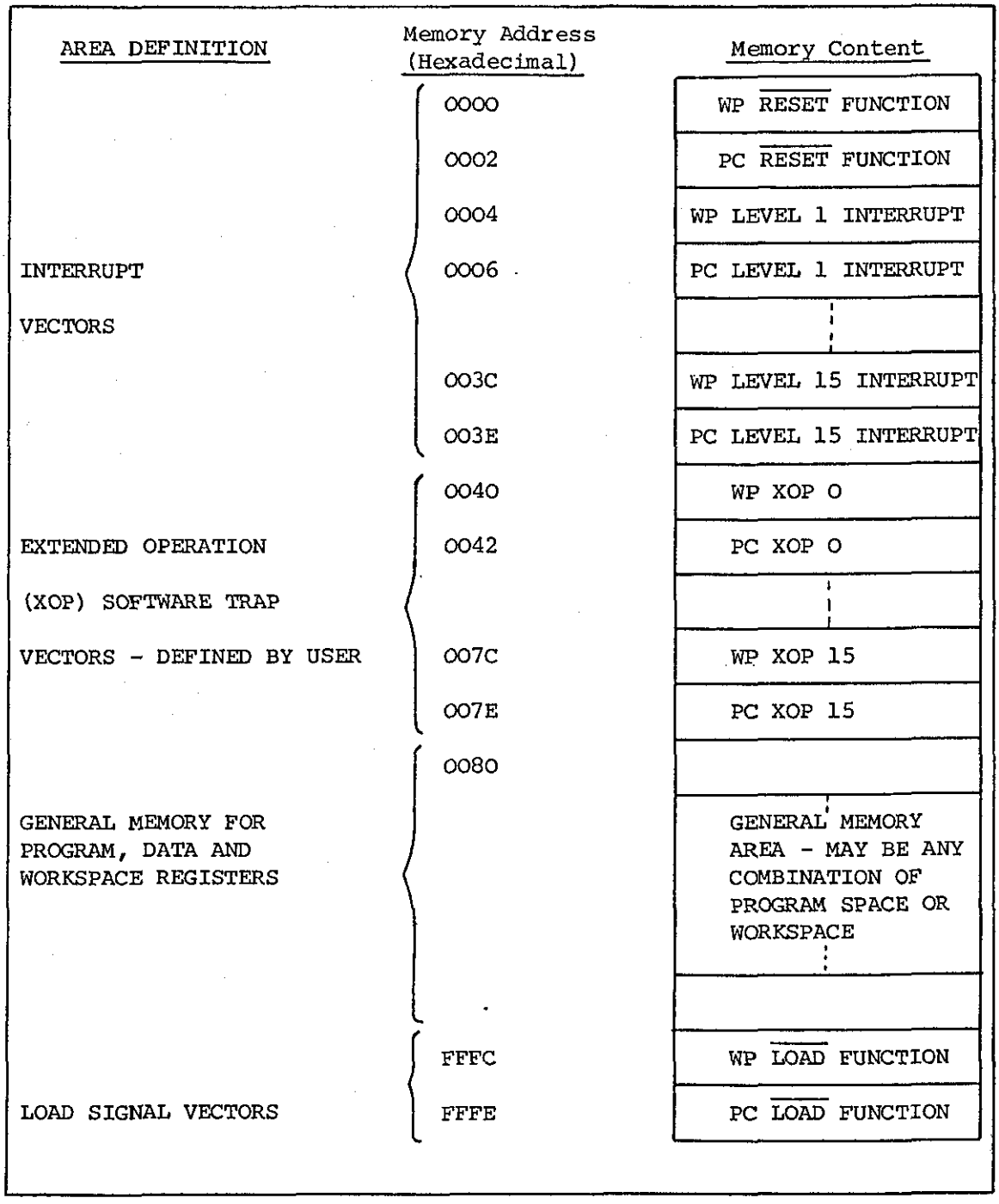


Fig. 5.7: TMS9900 Memory Map

memory read and write cycle is shown in Fig. 5.8. The read cycle is shown with no wait states and the write cycle is shown with one wait state.

$\overline{\text{MEMEN}}$ goes active (low) during each memory cycle. At the same time that $\overline{\text{MEMEN}}$ is active, the memory address appears on the address bus bits A_0 through A_{14}. If the cycle is a memory-read cycle, DBIN will go active (high) at the same time $\overline{\text{MEMEN}}$ and A_0 through A_{14} become valid. The memory write signal $\overline{\text{WE}}$ will remain inactive (high) during a read cycle. If the read cycle is also an instruction acquisition cycle, IAQ will go active (high) during the cycle.

The READY signal, which allows extended memory cycles, is shown high during Ø1 of the second clock cycle of the read operation. This indicates to the TMS9900 that memory-read data will be valid during Ø1 of the next clock cycle. If READY is low during Ø1, then the TMS9900 enters a wait state suspending internal operation until a READY is sensed during a subsequent Ø1. The memory read data is then sampled by the TMS9900 during the next $\emptyset_1$, which completes the memory-read cycle. At the end of the read cycle, $\overline{\text{MEMEN}}$ and DBIN go inactive (high and low, respectively). The address bus may also change at this time, however, the data bus remains in the input mode for one clock cycle after the read cycle.

A write cycle is similar to the read cycle with the exception that $\overline{\text{WE}}$ goes active (low) as shown and valid write data appears on the data bus at the same time the address appears. The write cycle is shown as an example of a one-wait-state memory cycle. READY is low during Ø1, resulting in the WAIT signal shown.

A DMA interface utilises the TMS9900 memory bus by using the hold operation (illustrated in Fig. 5.9) of the TMS9900. When $\overline{\text{HOLD}}$ is active (low), the TMS9900 enters the hold state at the next available non-memory cycle, and raises HOLDA to acknowledge the $\overline{\text{HOLD}}$ request.

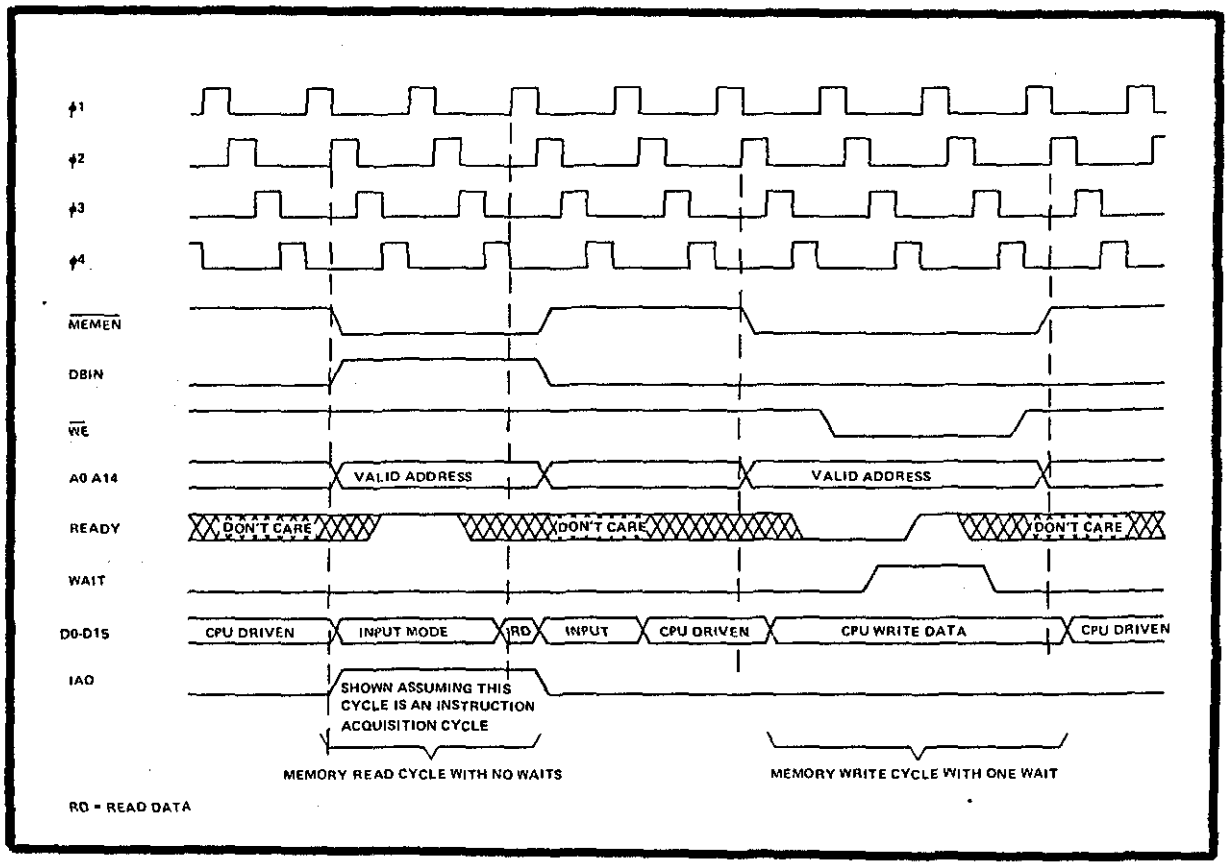


Fig. 5.8: TMS9900 Memory Bus Timing

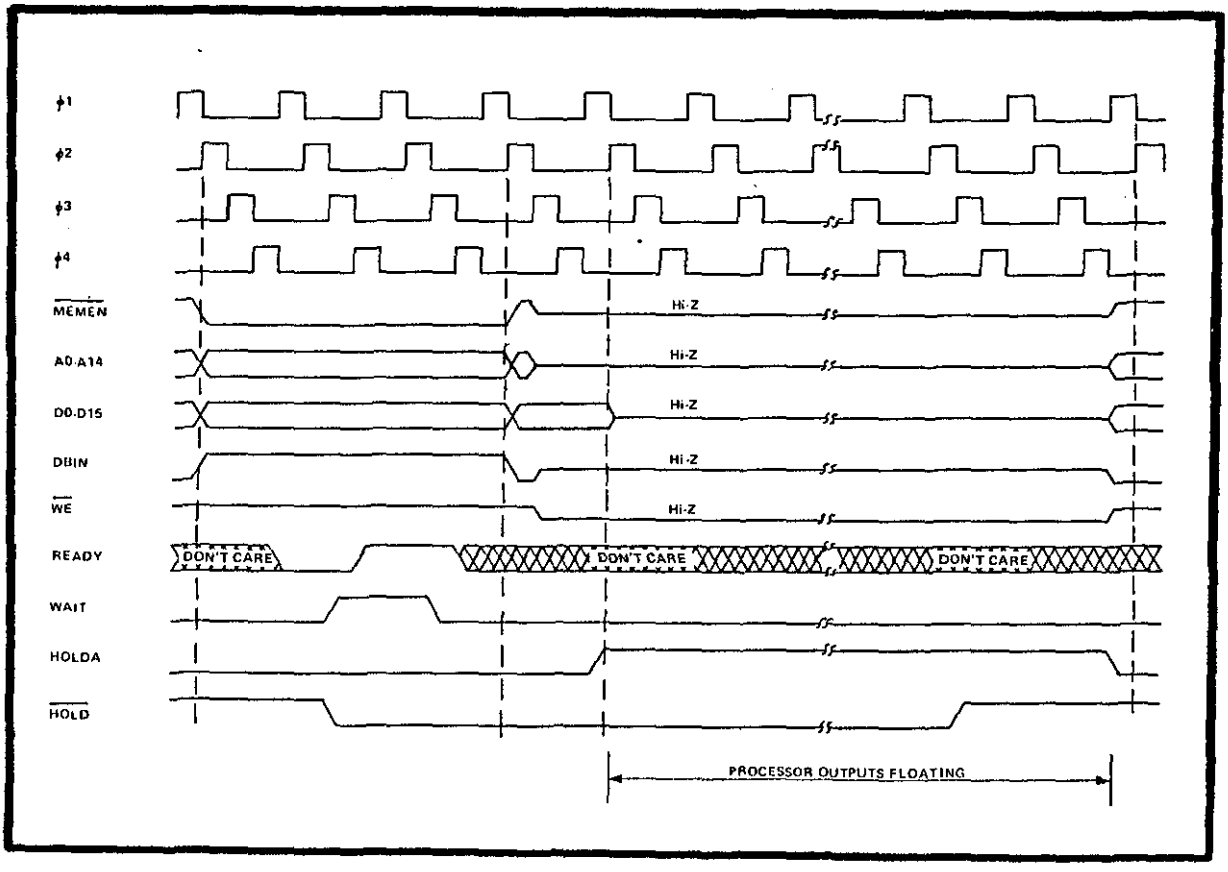


Fig. 5.9: TMS9900 HOLD Timing

The maximum latency time between the hold request and the hold acknowledge is equal to three clock cycles plus three memory cycles. When HOLDA is active, A_0 through A_{15}, D_0 through D_{15}, DBIN, $\overline{\text{MEMEN}}$, and $\overline{\text{WE}}$ go into a high impedance state to allow other devices to use memory buses.

5.3.9 Addressing modes

The 69 commands of the TMS9900 provide 26 arithmetic, logic and data manipulation instructions, 14 internal register-to-memory operations, 5 data transfer commands and 24 control functions. The instructions contain the following addressing modes;

1. Workspace register addressing.
2. Workspace register indirect addressing.
3. Symbolic memory addressing.
4. Indexed memory addressing.
5. Workspace register indirect autoincrement addressing.

5.4 Microprocessor System Design

A block diagram of the TMS9900 system built as an integral part of VAS is shown in Fig. 5.10. The system consists of;

1. TMS9900 microprocessor.
2. 1K words of ROM containing the system software and 9K words of RAM, with provisions for expanding the RAM by plugging in 4K words memory boards.
3. Memory mapped I/O for interfacing with other modules of VAS.
4. 16-bit CRUIN interface.
5. 16-bit CRUOUT interface.

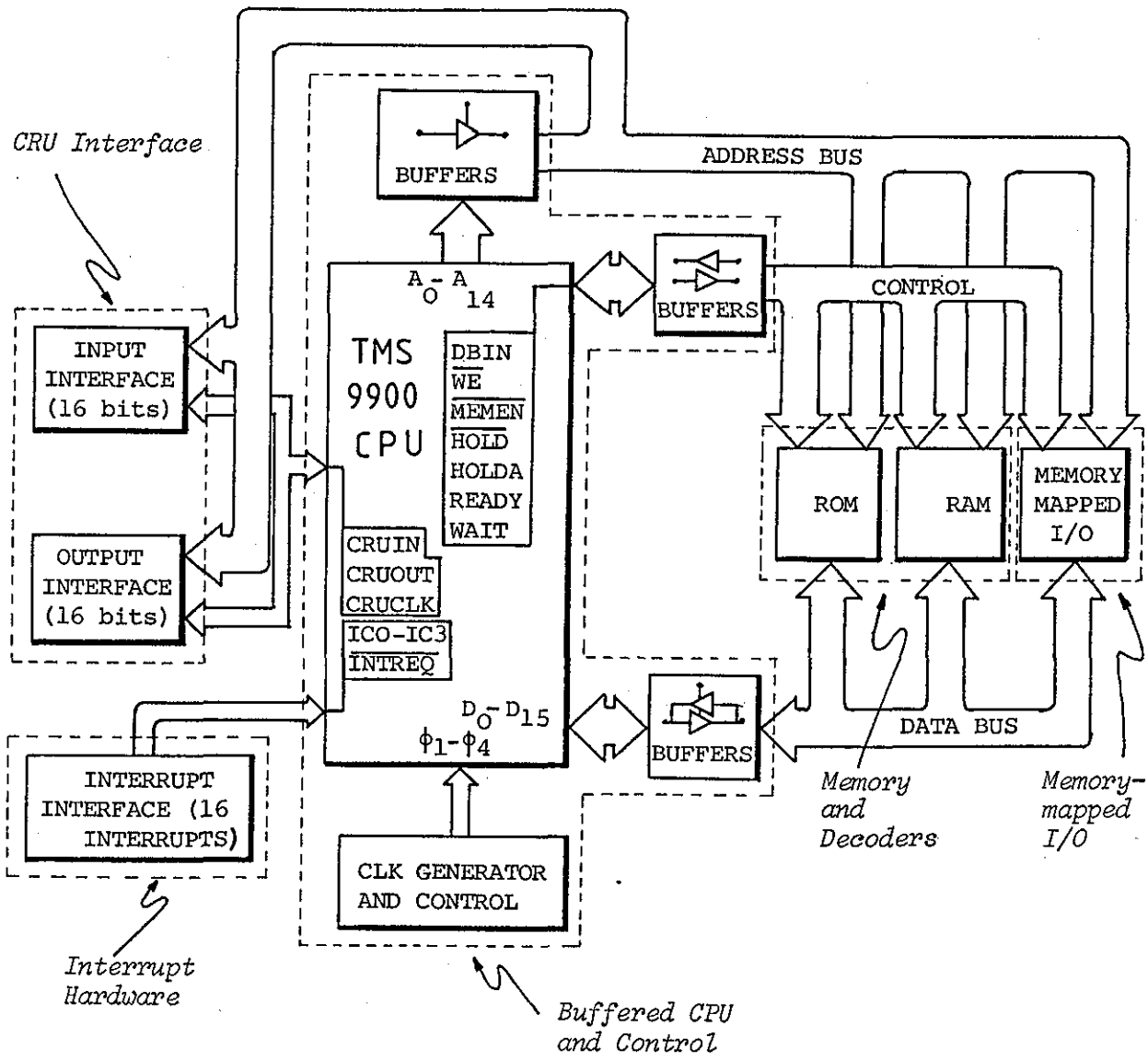


Fig. 5.10: Overall TMS9900 System

6. Interface to Texas Instruments Model 700 ASR electronic data terminal.

7. 8-level interrupt interface.

The block diagram is divided by dotted lines into subsystems. These subsystems are dealt with briefly in subsequent sections.

5.4.1 Buffered CPU and control

Fig. 5.11 shows the buffered TMS9900 CPU and control. An SN74LS362 clock generator provides four phase MOS timing signals for the CPU. The clock frequency (3 MHz) is determined by a quartz crystal reference, and a simple LC network is used to control the frequency overtone. The SN74LS362 also provides TTL compatible clock outputs. An RC network connected to the Schmitt-triggered D-input of the SN74LS362 maintains an active $\overline{\text{RESET}}$ signal (low) for a short time immediately following the power-on, providing a power-on reset for the system in addition to the manual reset.

The $\overline{\text{RESET}}$ signal initialises the CPU by inhibiting $\overline{\text{WE}}$ and CRUCLK, and putting the CPU memory bus and control lines in a high impedance state. When the $\overline{\text{RESET}}$ signal is released, the CPU fetches the restart vector from locations 0000 and 0002, stores the old WP, PC and ST into the new workspace, resets all status bits to zero and starts execution at the new PC. The $\overline{\text{RESET}}$ signal must be held active for a minimum of three clock cycles.

The LOAD signal is normally used to implement a restart ROM loader or front panel functions. When active (low) the $\overline{\text{LOAD}}$ signal causes the CPU to perform a non-maskable interrupt. The $\overline{\text{LOAD}}$ signal can be used to terminate a CPU idle state.

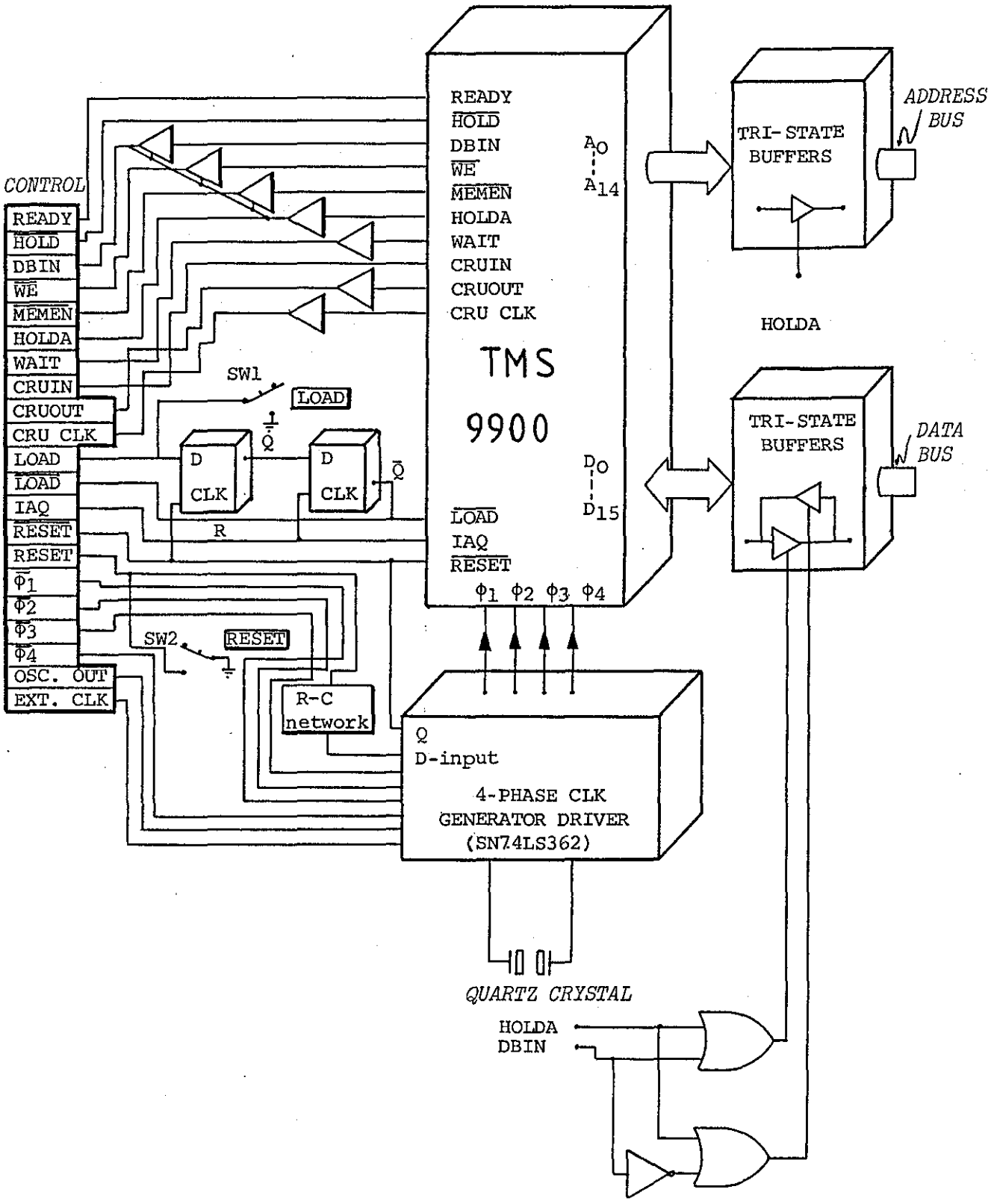


Fig. 5.11: Buffered TMS9900 CPU and Control

The $\overline{\text{LOAD}}$ signal should be active for one instruction period. Since there is no standard TMS9900 instruction period, IAQ is used to determine instruction boundaries. If the switch SWI is in the LOAD position during the time that the $\overline{\text{RESET}}$ signal is released, the CPU will perform the $\overline{\text{LOAD}}$ function immediately after the $\overline{\text{RESET}}$ function is completed, by fetching the $\overline{\text{LOAD}}$ vector from addresses $FFFC_{16}$ and $FFFE_{16}$ (subscript 16 denotes hexadecimal notation). If the switch is not in the LOAD position when the $\overline{\text{RESET}}$ signal is released the CPU uses the normal restart vector at 0000-0002.

The system software (the monitor and the assembler) is stored in EPROMs placed at the low end of the memory map ($F800_{16}$ through $FFFE_{16}$) and the last two locations $FFFC_{16}$, $FFFE_{16}$ contain the WP and the starting address of the monitor. As a consequence the $\overline{\text{LOAD}}$ function transfers control to the monitor.

All TMS9900 output control lines are buffered. HOLDA when active (high), puts the tri-state buffers of DBIN, $\overline{\text{WE}}$, $\overline{\text{MEMEN}}$ and the address bus in the high-impedance state to allow I/O devices to use the memory bus. HOLDA and DBIN signals control the data bus tri-state buffers. It should be noted that the TMS9900 buses and control lines are already provided with internally controlled tri-state outputs. The additional buffers are inserted in order to increase the drive capability of the buses.

5.4.2 Memory and address decoders

Two 1K x 8 bit EPROM's (Intel 2708) are used to provide 1K words of program storage and as mentioned earlier they are resident between memory locations $F800_{16}$-$FFFE_{16}$. In addition sixteen 256 x 4 bit static RAM's (TMS4042) are wired in parallel to provide 1K words of read/

write memory. This bank of RAM's is situated between locations 0000_{16}-$07FE_{16}$. The first 16 words (0000_{16}-$001E_{16}$) store interrupt trap vectors in order of priority, the non-maskable reset vector occupying the first two locations. The next contiguous block of 48 memory words (0020_{16}-$007E_{16}$) is used by the monitor and assembler as workspace register files and temporary data storage.

A block diagram of the memory and address decoders is illustrated in Fig. 5.12. Fig. 5.13 shows a schematic diagram of the address decoders. The decoders detect any reference to the top 2K words and bottom 2K words. The top 2K words (0000_{16}-$0FFE_{16}$) are allocated to the RAM's, and the bottom 2K ($F000_{16}$-$FFFE_{16}$) to the EPROM's. The assembler is located at $F800_{16}$ through $FBFE_{16}$ and the monitor at $FC00_{16}$ through $FFFA_{16}$. The extra space bounded by $F000_{16}$ through $F7FE_{16}$ and 0800_{16} through $0FFE_{16}$ is reserved in anticipation of the availability of additional EPROM's and RAM's respectively.

The two signals top 2K and bottom 2K are used in conjunction with DBIN and $\overline{\text{MEMEN}}$ (Fig. 5.12) to control the tri-state buffers connecting the data bus to the memory. The decoders also generate memory chip select signals (CS). Provisions are made for expanding the RAM by plugging in 4K x 16 memory boards designed and built in the Department of Electronic and Electrical Engineering, University of Loughborough. At present 2 boards are connected and this results in a total of 9K words of RAM.

The memory device used in the memory boards is INTEL 2114L, with access time of 450 nsec, which falls within the minimum access time determined by the system clock (about 500 nsec). The TMS4042's, with their access time of 1000 nsec., may not properly respond within the minimum access time. For this reason two WAIT STATES are inserted

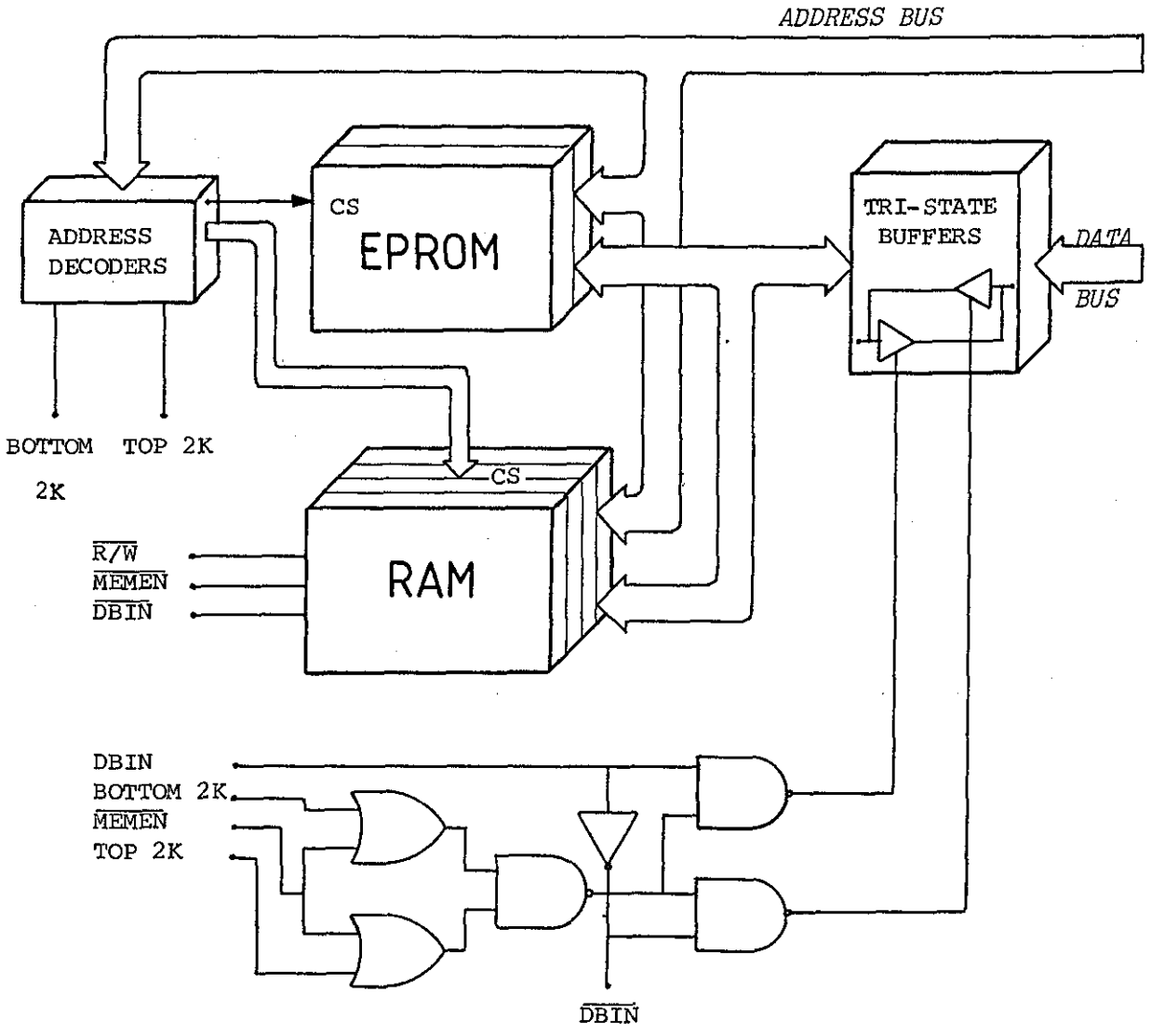


Fig. 5.12: Memory and Decoders

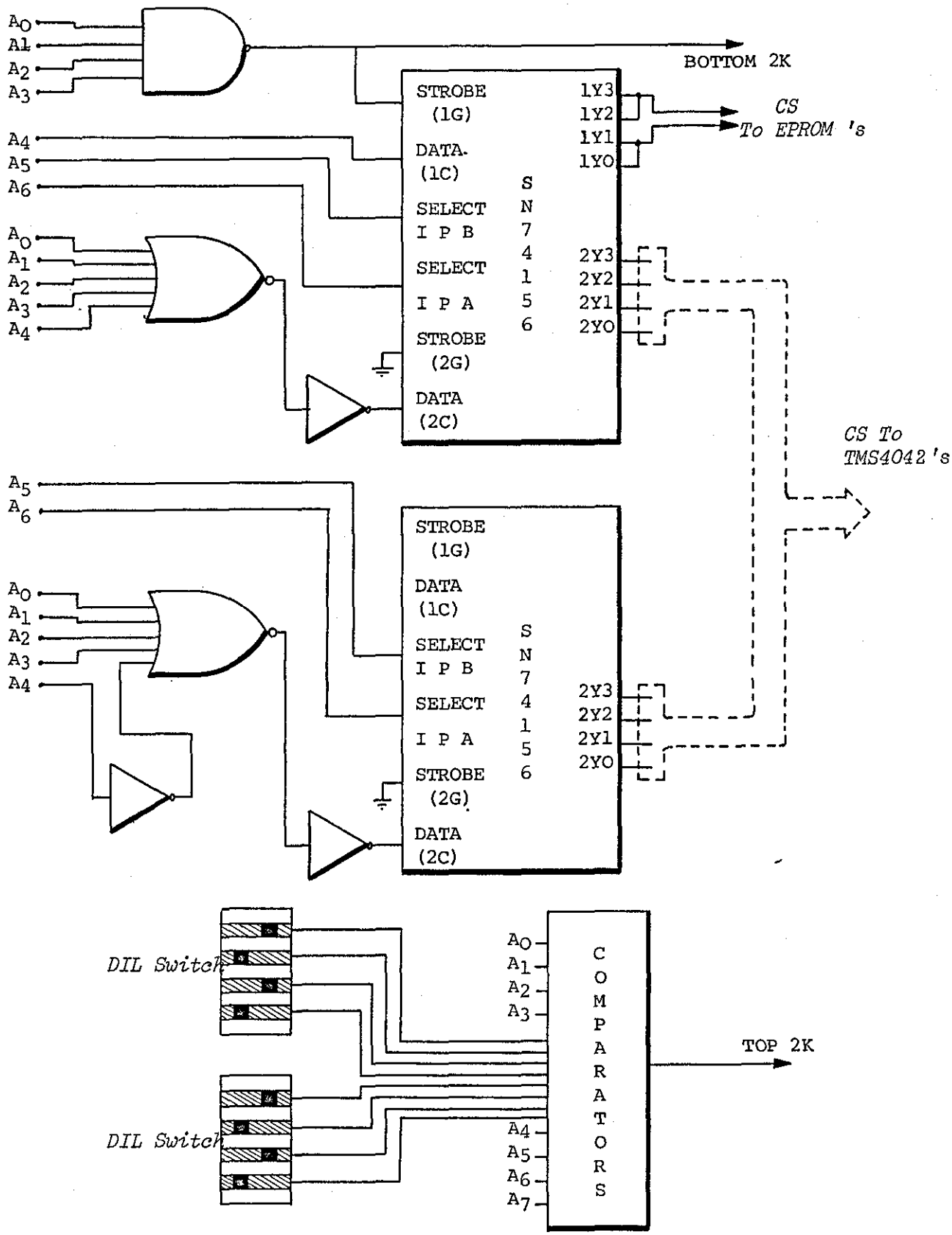


Fig. 5.13: Address Decoders

into memory access cycles whenever the CPU is addressing a TMS4042. Fig. 5.14 shows the circuit used to generate the two WAIT states by delaying propagation of the WAIT output to the READY input one clock cycle with a D-type flip-flop. For this reason programs stored in the 4042's have longer execution time and hence the memory boards

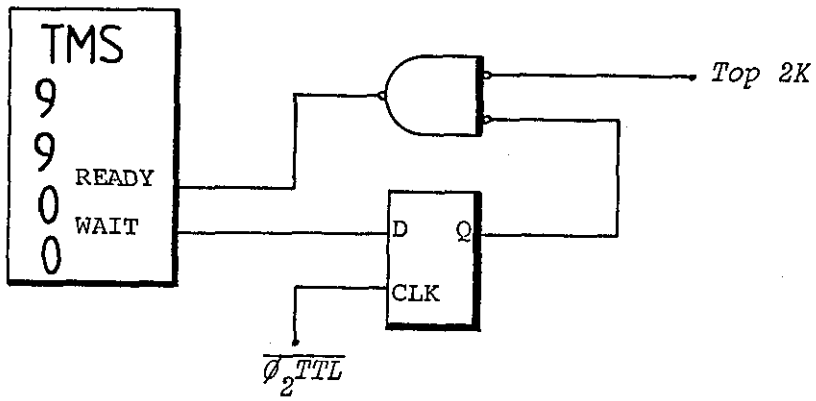


Fig. 5.14: Generation of Double WAIT STATES

are assigned to real time programs where execution time is of utmost importance.

5.4.3 CRU interface

A block diagram of CRU interface with the TMS9900 is shown in Fig. 5.15. Two octal multiplexers (SN74LS251) and two addressable latches (SN74LS259) are used for the CRU-based I/O. The interface consists of A_0-A_{14}, CRUIN, CRUOUT and CRUCLK. A_0-A_2 indicate whether data is to be transferred and A_3-A_{14} contain the address of the selected bit for data transfer.

The TMS9900 develops the CRU address from the CRU-base address contained in workspace register 12 and a signed displacement count

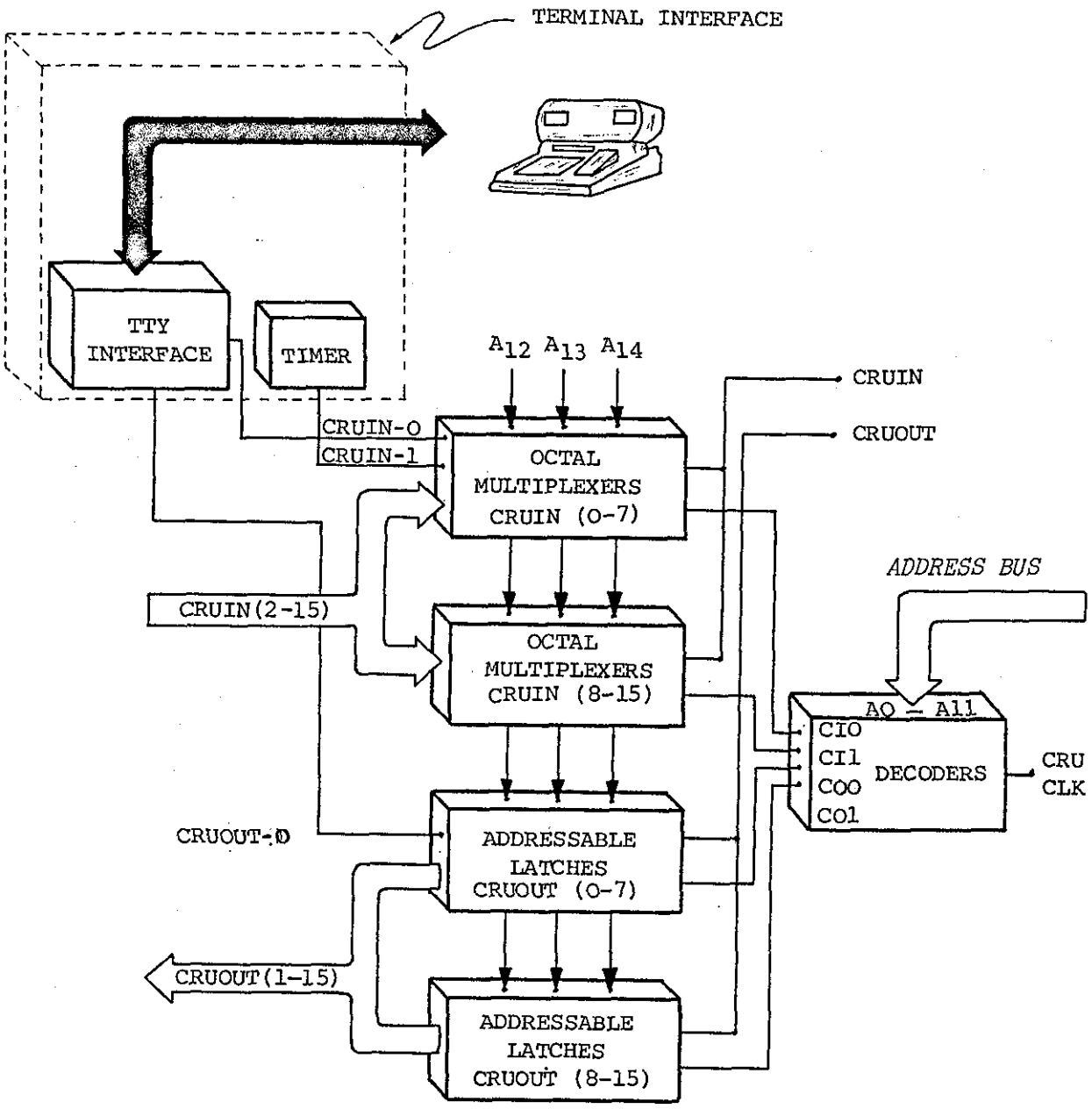


Fig. 5.15: Block Diagram of CRU Interface

contained in the instruction. The decoders utilise the upper address bits A_0-A_{11} to generate CIO CI1, COO and CO1. The octal multiplexers are used for CRU inputs and are enabled by CIO and CI1. The octal addressable latches are used for CRU outputs. COO and CO1 are connected to the latches and the latch outputs are altered only when CRUCLK is pulsed. The I/O bits are addressed as bits 0-F. The least significant address bits (A_{12}-A_{14}) determine which of the eight inputs or outputs in the multiplexers and latches is addressed. The I/O bits may be expanded by adding appropriate decoding logic. Two of the input bits (0-1) and one output bit (0) are used for RS-232/TTY interface. A timer connected to CRUIN-1 controls the baud rate and can be manually adjusted through the range 110-1200 baud.

5.4.4 Memory mapped input/output

To achieve fast transfer of data from the coordinate generator to the TMS9900 and from the TMS9900 to display unit-2 interface; it is essential to send data in parallel along the data bus rather than using a multiple bit CRU operation which takes 58 clock cycles for its execution. One possibility was to use a DMA interface. DMA permits the accessing of memory without having the CPU involved. This was avoided for the following reasons;

a) For fast manipulation and easy retrieval of data it is desirable to preprocess the coordinates of the edge points before storing them for further analysis. This entails performing some logic and arithmetic operations and consequently raw data has to pass through the CPU first.

b) Data for display is readily available from the CPU and since there is no need to store it in bulk, a DMA interface is not convenient for outputting it to a display device.

c) DMA control circuitry is somewhat expensive and complex.

An alternative to a DMA interface is to implement a memory mapped I/O scheme. Here the address lines are decoded such that the input ports and output ports simply appear to be the contents of specific memory locations. The memory locations selected lie outside the memory of the system. The circuitry for decoding the address lines is shown in Fig. 5.16. The 4 top address lines (A_0-A_3) are used for generating two sets of signals (MAP1, MAP2 and MAP3-MAP5) which are routed to the coordinate generator and the display unit-2 interface respectively. The role of these signals will be explained later.

5.4.5 Interrupt interface

The TMS9900 has 15 user interrupt levels in addition to the $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ functions. These interrupts are maskable and prioritised. Their transfer vectors are similar to the $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ vectors. The presence of an interrupt is indicated by an external device driving $\overline{\text{INTREQ}}$ low and placing the priority code on IC0 through IC3. The interrupt interface implemented in the TMS9900 system has eight unique levels. The priority encoding logic for this interrupt service is shown in Fig. 5.17.

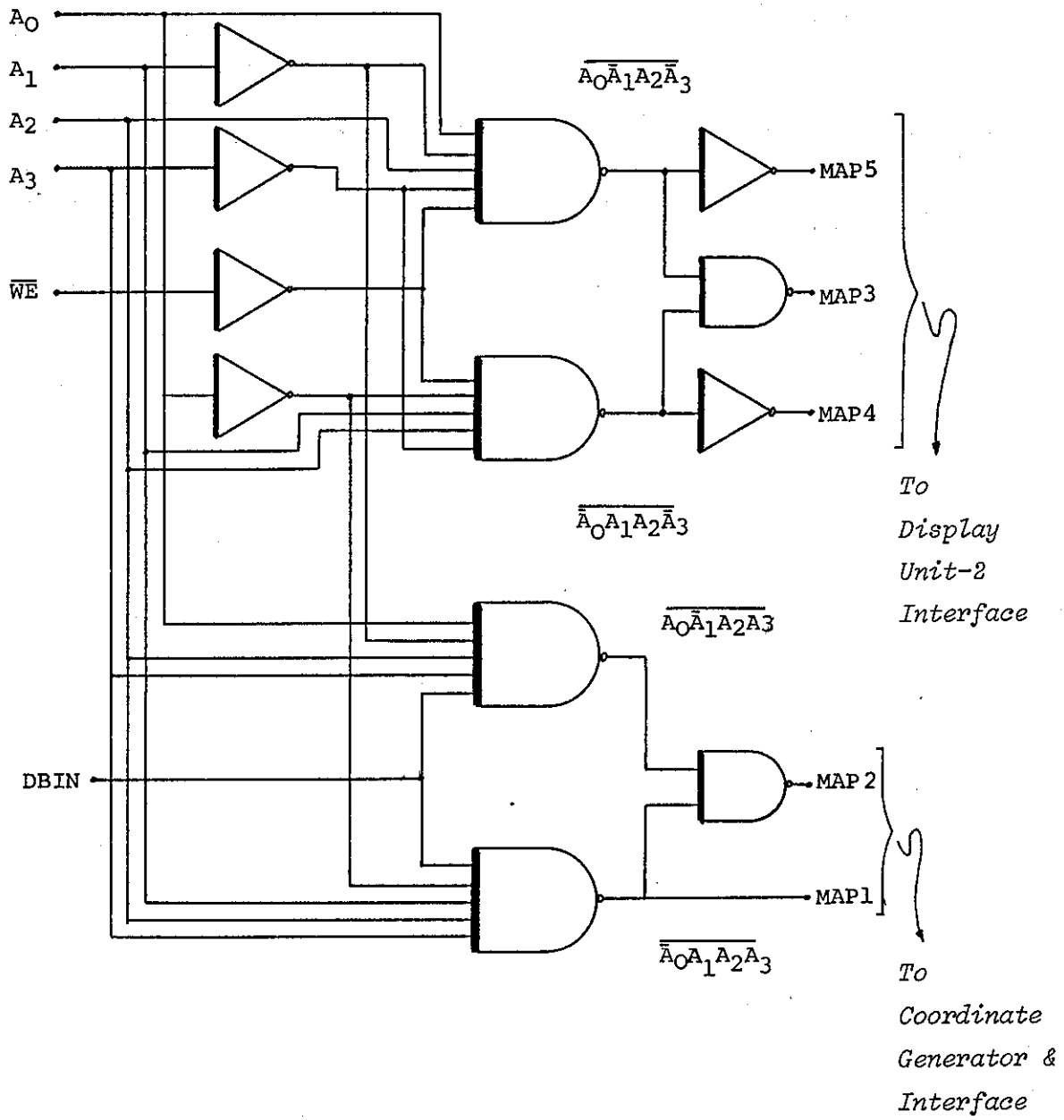


Fig. 5.16: Address Lines Decoders for Memory Mapped I/O

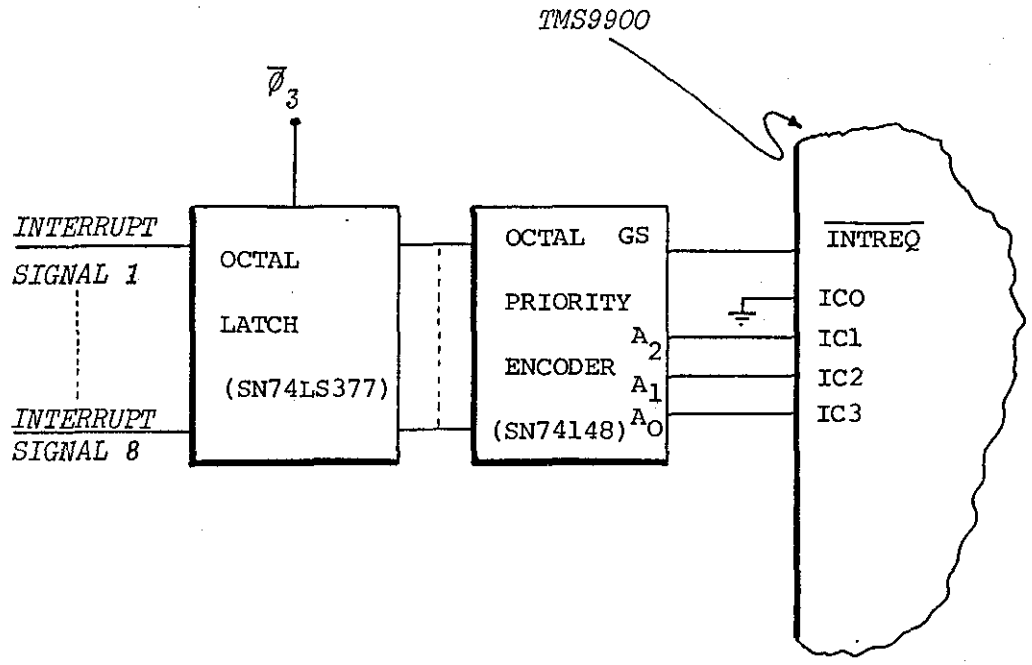


Fig. 5.17: Eight Input Interrupt System

5.4.6 System software

The 1K words of the EPROM is divided equally between the monitor and the assembler, each occupying $\frac{1}{2}$K words. The system software support is not the work of the Author.

a) The Monitor

The monitor is an interactive software program which provides full control of the system during program development. It is used for debugging the user software under the silent 700 console control.

The monitor prompts the user to enter a command by typing a question mark at the beginning of a new line. In response the user enters a single character command code specifying the feature of

the monitor he wishes to use, followed by the arguments in hexadecimal. The command is terminated by any non-hexadecimal digit including carriage return. The monitor commands perform the following functions:-

1. Modify contents of memory.
2. Copy contents of a block of memory into another area of memory.
3. Setting a breakpoint or trap in the user program. Whenever the processor encounters the trap, the state of the machine is saved and control is transferred back to the monitor for user action.
4. Listing contents of a block of memory on the printer or recording it on a cassette tape.
5. Loading a program file into memory from cassette tape.
6. Modify a CRU bit.
7. Transfer control to a specific location in memory for program execution.
8. Calculate hexadecimal sum and difference of two numbers.
9. Display a CRU bit.
10. Display user workspace registers.
11. Snap registers and blocks of memory when user program is running.

b) Assembler

A one pass, or instant input assembler provides the capability to enter patches or short programs using the standard TI9900 mnemonics and operands. It is started by branching to memory location $F800_{16}$ using a monitor command.

Besides the TI9900 mnemonics the assembler recognises three commands. Two of them are used for storing string constant (in hexadecimal) and numeric constant (in decimal or hexadecimal). The third command is used for changing the program counter location. Being small the assembler can detect only three types of errors, which are:-

1. Syntax error, when input contains a syntax error.
2. Displacement error, when target address of a jump exceeds the allowable range.
3. Range error, when input is out of range.

5.5 Camera Drive and Interface

The camera drive and interface subsystem consists of a clock generator, a TV sync. pulse generator, a two stage video amplifier, a lowpass filter and an analogue-to-digital convertor (ADC). It interfaces a TV camera to the edge detector and supplies synchronising pulses to other subsystems.

5.5.1 Clock generator and camera drive

A schematic diagram of the clock generator and camera drive is shown in Fig. 5.18. A buffered 10 MHz crystal oscillator provides clock pulses to the edge detector and latches in other subsystems of VAS. The TV sync generator (ZNA134) utilises a 2.5MHz clock to produce synchronising pulses necessary for raster generation in a 625 line TV system with 2:1 interlaced scanning.Outputs of the sync generator are buffered and routed to destinations outlined in the diagram.

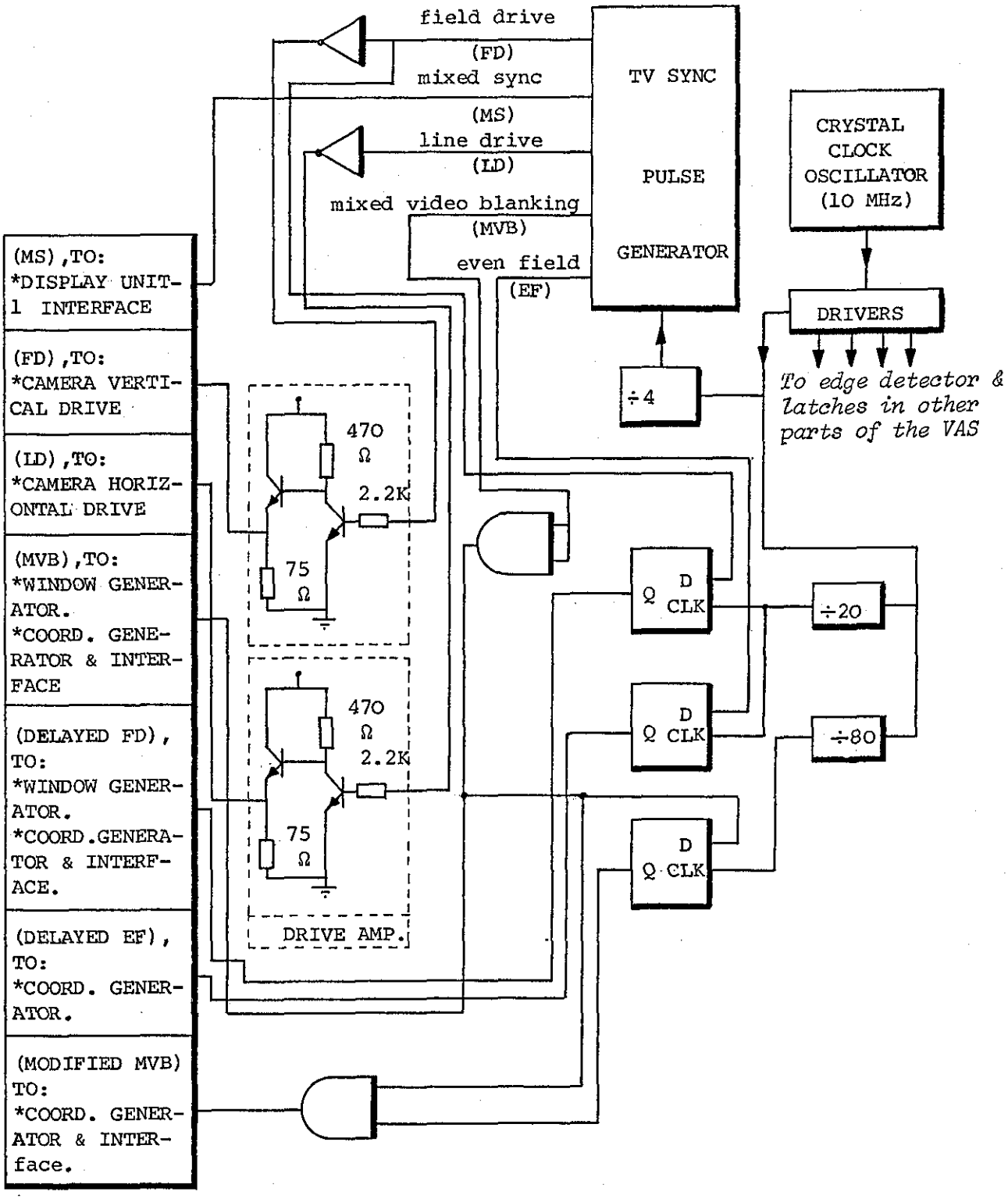


Fig. 5.18: Clock Generator and Camera Drive

Even field (EF) and field drive (FD) pulses are delayed to provide proper timing to generate the least significant bit of the coordinates of edges as will be shown later. Mixed video blanking (MVB) pulses are modified and used to eliminate edges generated by the edge detector at sync level. The modified MVB has wider field blank and line blank periods, produced by delaying the MVB using a D-type flip flop, and ANDing the delayed signal with the original one (Fig. 5.19).

5.5.2 Video amplifier

Output video signal of the TV camera (super lynx) is 1.4V peak to peak. This output is amplified by a two-stage video amplifier, with a bandwidth of 4.7 MHz, voltage gain of 5.6 and 75-Ω output impedance. The amplifiers are constructed on P.C. boards, implementing L115 operational amplifiers feeding push-pull stages. The circuit diagram of the first stage is shown in Fig. 5.20. The second stage is similar to the first. In the first stage a provision is made for changing the D.C. level of the video signal through a 47K potentiometer, which has its output summed with the video input by the L115. The output of each stage is shown in Fig. 5.21.

The video amplifiers serve two purposes. First, by amplifying the video signal, greater values of its dynamic range are available for analogue-to-digital conversion, rendering it more immune to noise. Second, by changing the D.C. level of the video signal the whole or any part of the dynamic range of the signal can be relayed to the edge detector.

5.5.3 Lowpass filter

An essential feature of the lowpass filter is that the characteristics of the video signal through it should be preserved

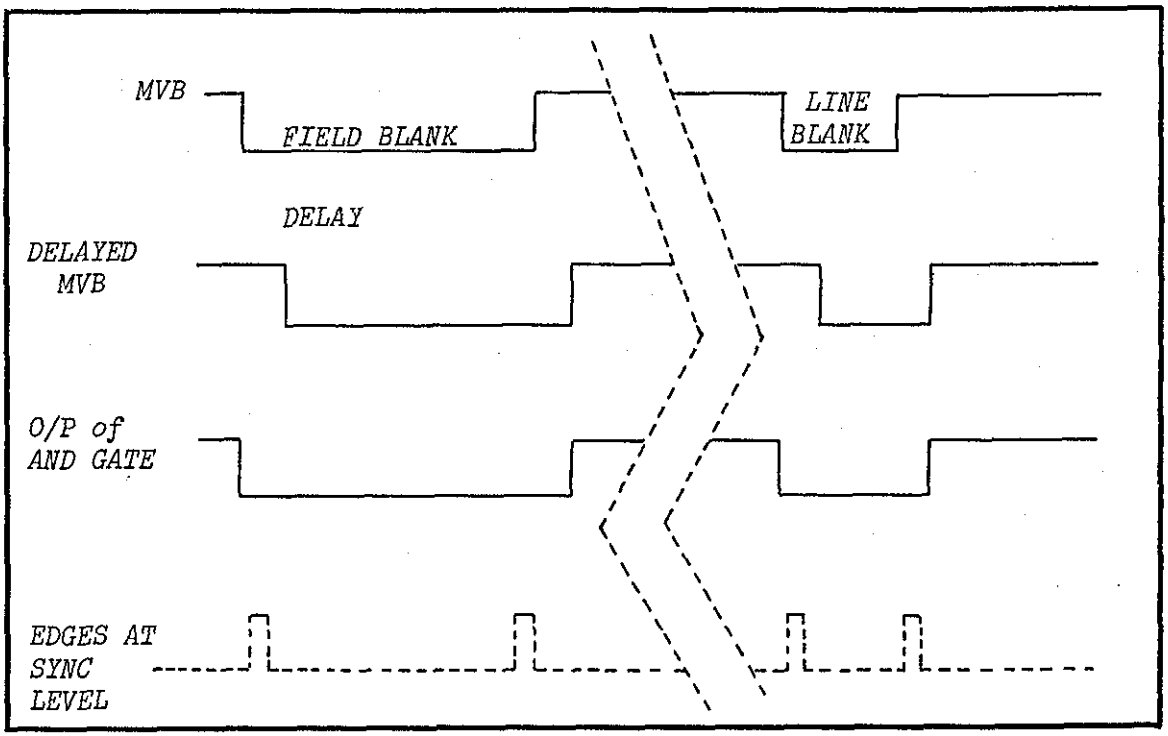


Fig. 5.19: Modified MVB to get rid of Edges at Sync Level

without significant distortion. Consequently the phase response of the filter in its passband should be linear or equivalently, both phase- and group-delay must be constant in the passband. One way of achieving this is to adopt the "brute force" approach of designing the filter from its loss specifications and then phase-equalising it by cascading all pass network. This approach was avoided because of its complexity.

Following an extensive literature survey a technique suggested by Temes and Gyi has been adopted[64,65]. This technique is based on designing a filter with Gaussian passband and Chebyshev stopband. Here the flat delay is approximated using a Gaussian function. A disadvantage of Gaussian filters is that they cannot provide appreciable loss selectivity even when the order is high. The remedy is to

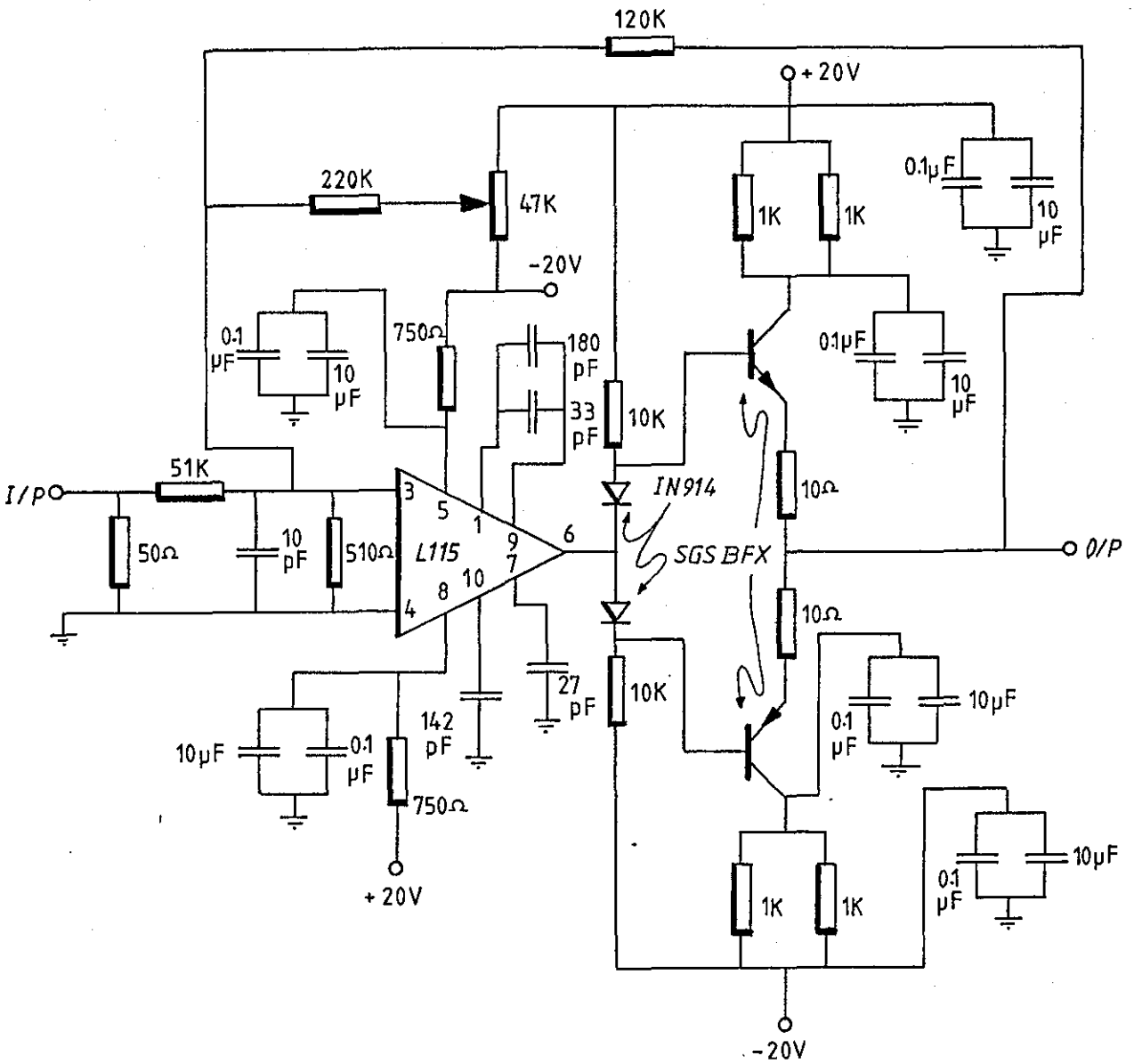
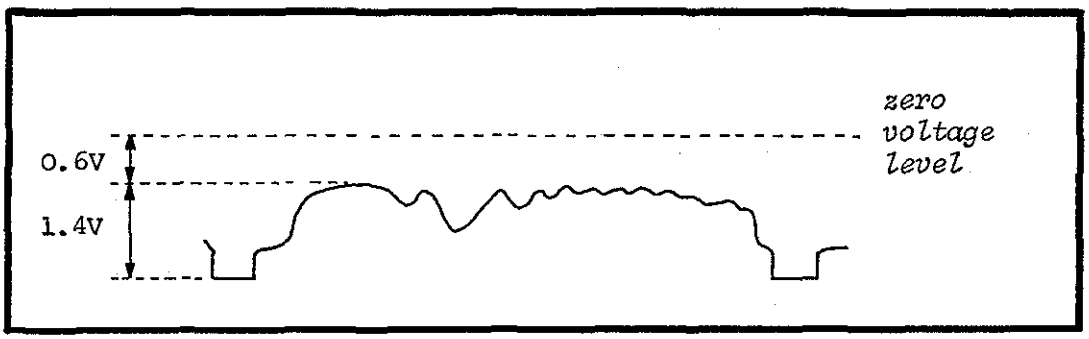
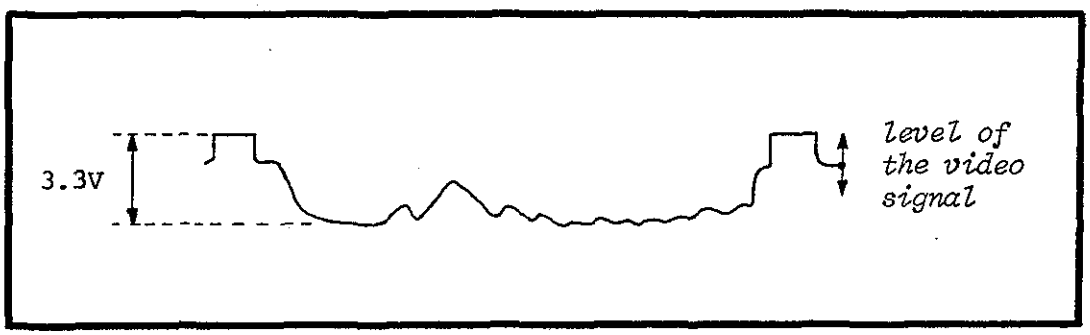


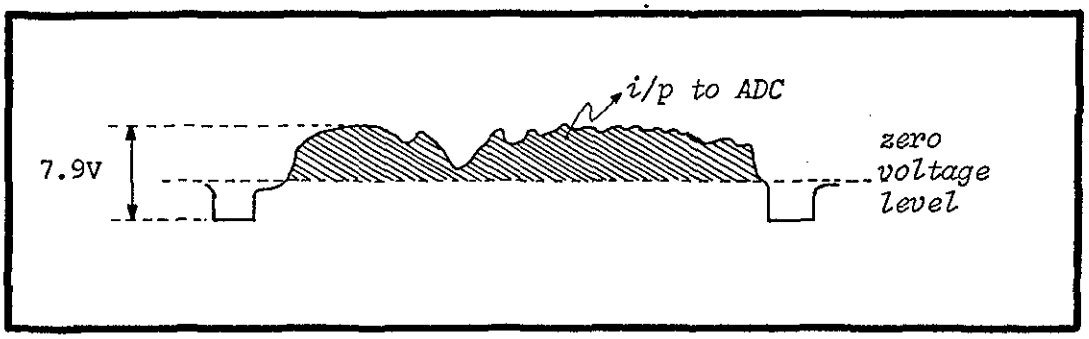
Fig. 5.20: Video Amplifier - 1st. Stage



(a) Input to 1st. stage.



(b) Output of 1st. stage.



(c) Output of 2nd. stage.

Fig. 5.21: Amplified Video Signal.

incorporate a Chebyshev stopband to improve the selectivity of the linear-phase passband.

The filter designed is of order 5, with stopband attenuation of 40 dB at 5.6 MHz, and is based on parameters (natural frequencies and loss-poles) derived by Temes and Gyi[65]. The design process consists of a mathematical procedure to construct the transfer function of the filter, followed by tedious numerical analysis in order to synthesise the filter through ladder development[66]. Fig. 5.22 illustrates the configuration and element values of the filter.

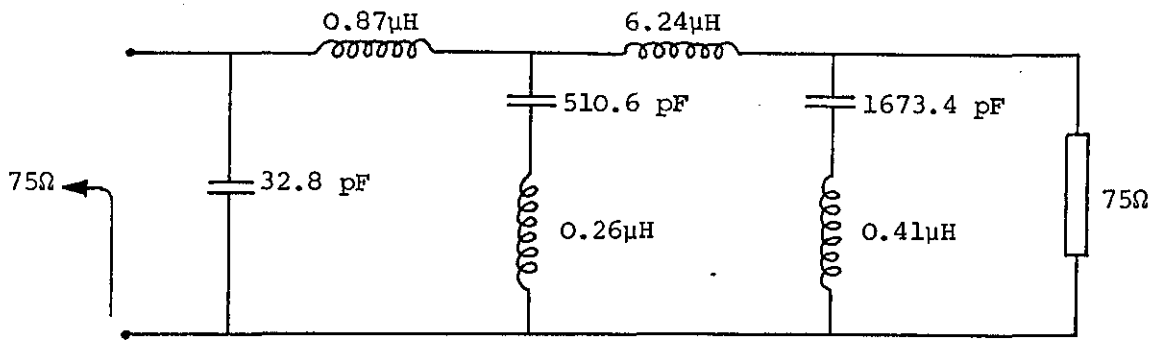


Fig. 5.22: Lowpass Filter with Gaussian Passband and Chebyshev Stopband.

5.5.4 The ADC[67]

The requirement to obtain bits at a rate of 10 MHz makes it essential to design a fast ADC. Parallel, or, as they are sometimes called simultaneous ADC, fulfil this requirement. They use one analogue comparator, with a fixed reference voltage at one of its inputs, for every quantisation level in the encoded digital word. Input analogue voltage is connected to the other input of each comparator so that an analogue comparison can be made with all the reference voltage levels representing all the quantisation levels. The outputs of the comparators drive an encoding logic to generate the equivalent digital word. Since conversion is performed in one step, rates of 50 megabits per second can be achieved using SN72710 differential comparators and ordinary TTL logic.

The main disadvantage of the parallel ADC is that, for each additional binary bit in the digital word, the amount of required circuitry is practically doubled; for example, for an 8-bit conversion it would be necessary to have 255 comparators, to generate 255 reference voltages and to have a proportionate number of gates in the encoding logic. To avoid excessive hardware it has been decided to build a 4-bit ADC. Fig. 5.23 shows a single-polarity four-bit parallel ADC. Since a four-bit word has fifteen possible quantisation levels other than zero, fifteen comparators are used. The outputs of the comparators are latched before sending them to the encoding logic. The circuit used to provide the reference voltage is shown in Fig. 5.24. The reference voltage is adjustable and can be varied to cover the whole or a smaller part of the dynamic range of the video signal.

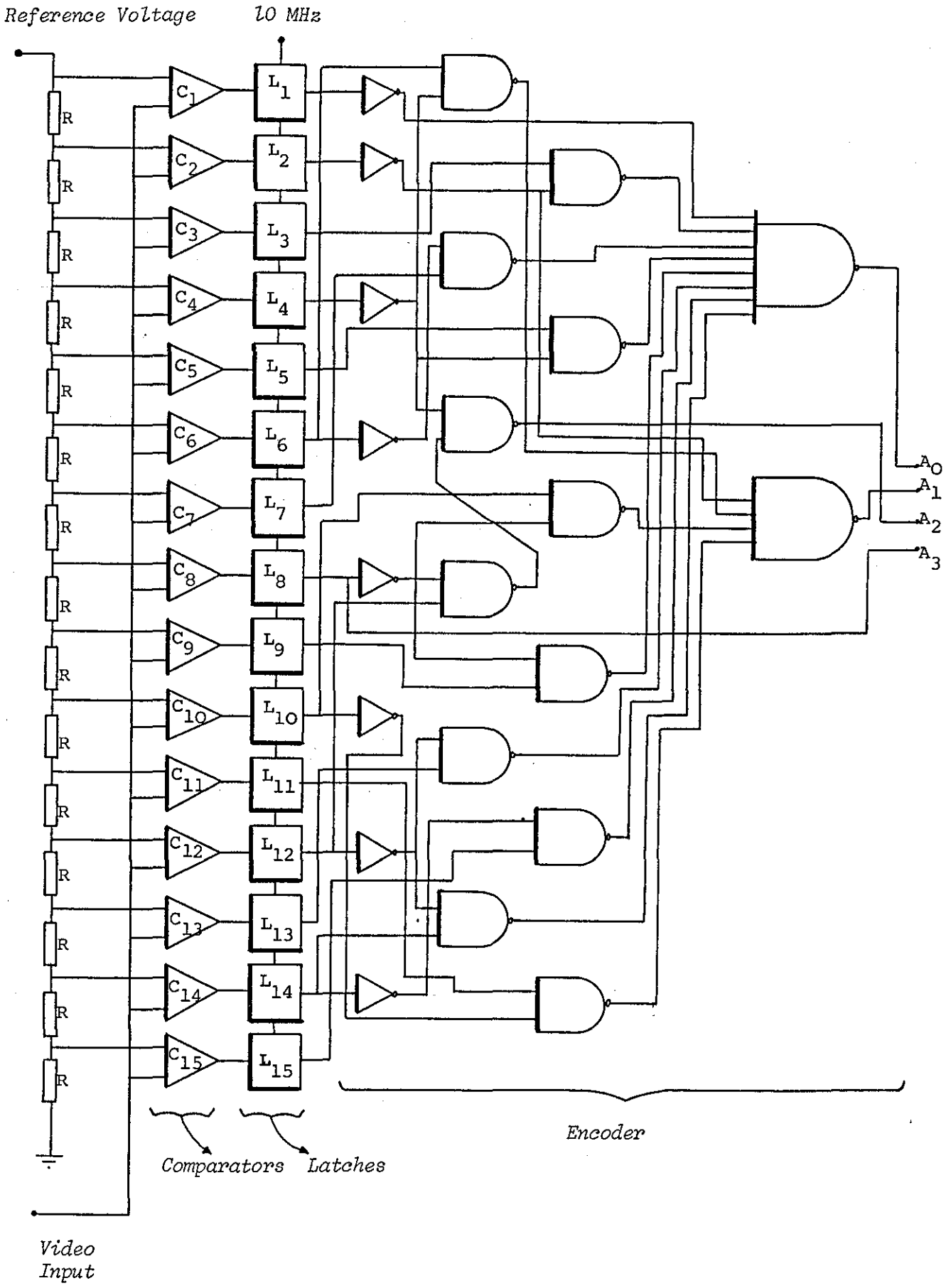


Fig. 5.23: A 4-Bit Parallel ADC Converter

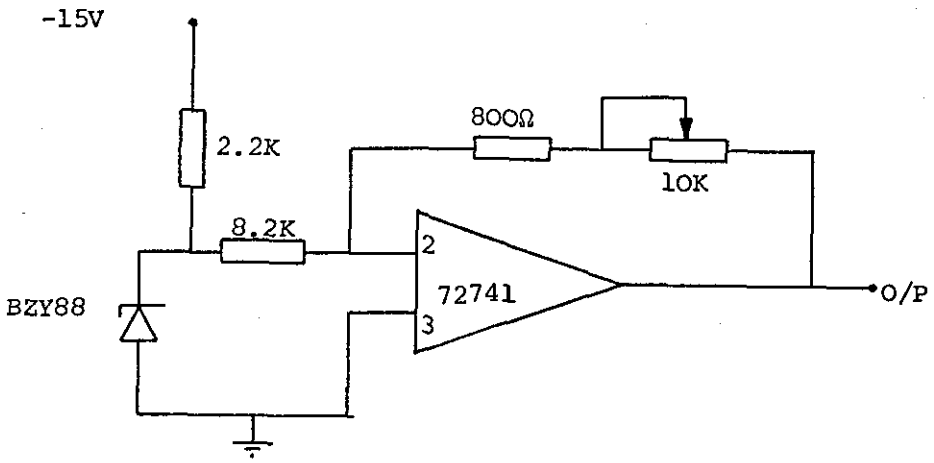


Fig 5.24: Generation of Reference Voltage for the ADC Converter

5.6 Coordinate Generator and its Interface

This subsystem consists of the three functional blocks (Fig. 5.25):-

1 - Counters to generate the x- and y-coordinates of edge points.

2 - FIFO buffers to ensure reliable transfer of coordinates from the counters to the microprocessor system.

3 - Input port of the microprocessor system.

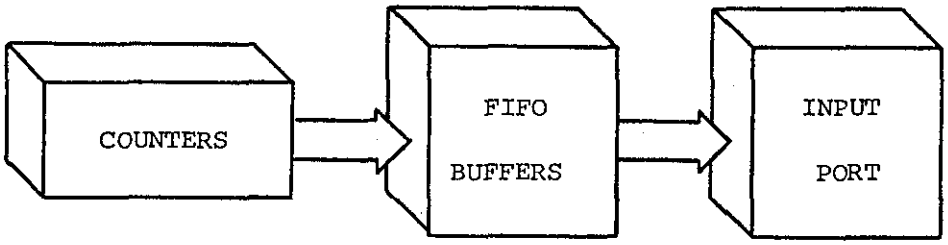


Fig. 5.25: Functional Diagram of Coordinate Generator and its Interface

5.6.1 Counters

To identify the location of edge points a Cartesian coordinate system is established on the raster, with its origin in the upper left hand corner of the screen. The x-coordinate increases from left to right and the y-coordinate from top to bottom i.e. in the same direction as the TV raster scan (Fig. 5.26).

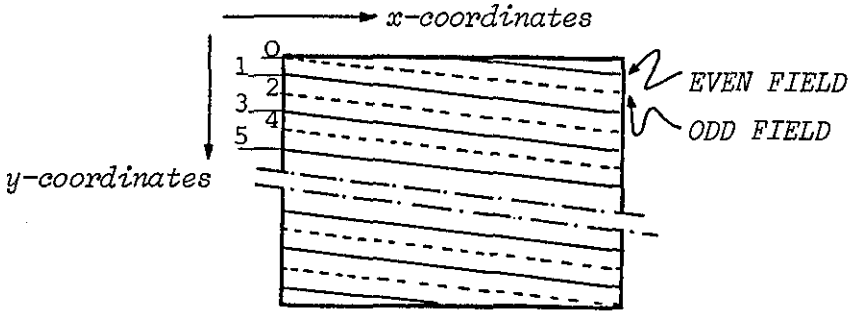


Fig. 5.26: TV Raster Scan

To create such a coordinate system two sets of 10-bit counters are employed, one for the x-coordinate (element or point counter) and the other for the y-coordinate (line counter), as shown in Fig. 5.27. Both counters are implemented using high speed counters with look ahead carry. In the x-coordinate counter the count-up rate is determined by the 10 MHz clock and the counter is reset by the MVB. For the y-coordinate counter, the count rate is determined by the MVB signal and the counter is cleared by the FD signal. The x-coordinate range is (0-533) and the y-coordinate range is (0-575). Thus any point within the TV raster can be approximated using a 533 x 575 point grid.

The fact that the scanning is interlaced was taken into consideration when designing the coordinate generator. Referring to Fig. 5.26, the y-coordinates of lines in the even field assume odd values while those in the odd field assume even values. To account for this, a least significant bit (C_0) is added to the y-coordinate values and the count sequence becomes as shown in Table 5.1.

Counters					C_0	
B_9	---	B_2	B_1	B_0		
0	---	0	0	0	0	0
0	---	0	0	1	0	2
0	---	0	1	0	0	4
0	---	0	1	1	0	6
0	---	1	0	0	0	8
					0	
					0	

a) Odd Field

Counters					C_0	
B_9	---	B_2	B_1	B_0		
0	---	0	0	0	1	1
0	---	0	0	1	1	3
0	---	0	1	0	1	5
0	---	0	1	1	1	7
0	---	1	0	0	1	9
0		1	0	1	1	11
0		1	1	0	1	
		1	1	1	1	

b) Even Field

Table 5.1: Count Sequence for Lines in Odd and Even Fields

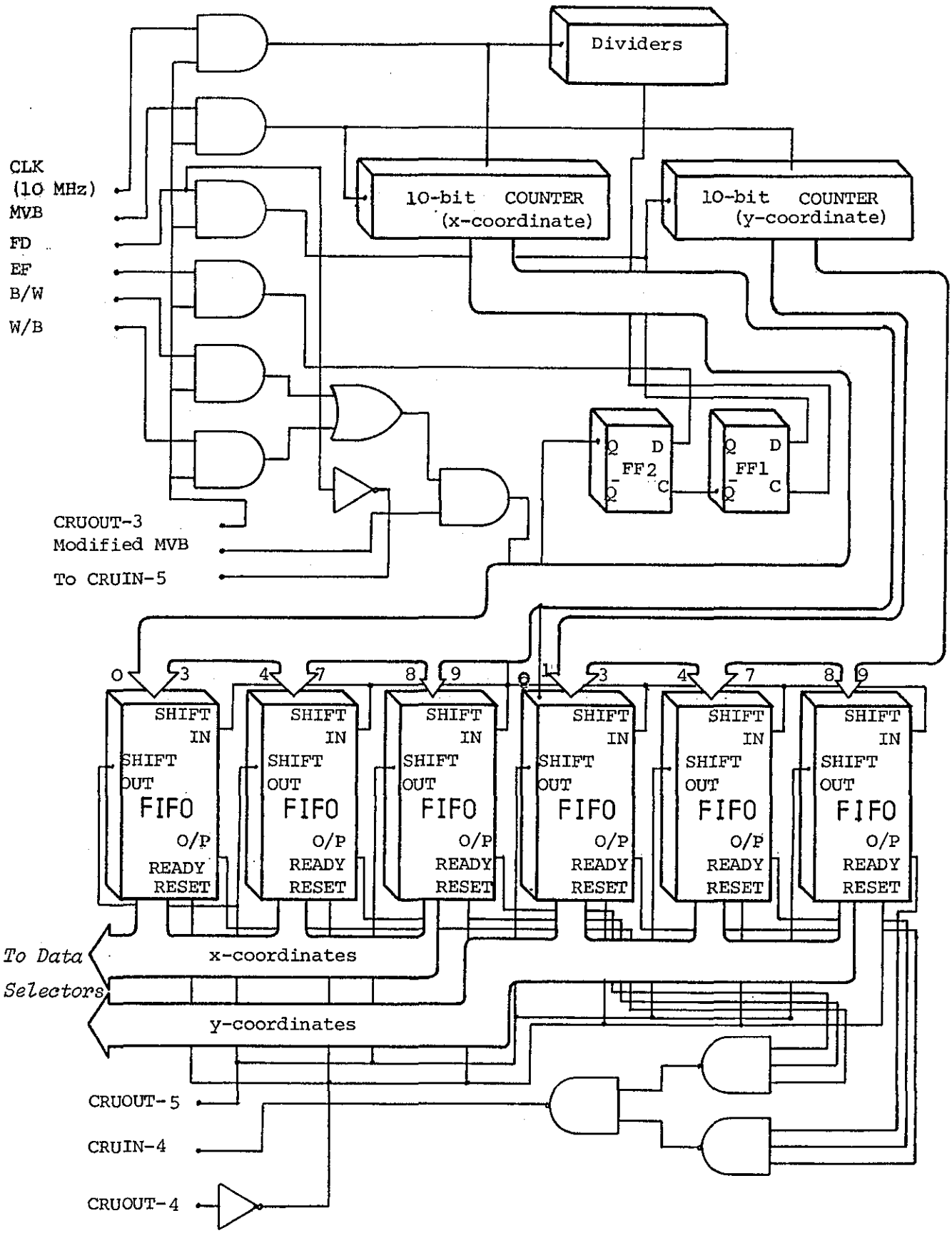


Fig. 5.27: Block Diagram of Counters and FIFO Buffers

The two flip-flops (FF1, FF2) of Fig. 5.27 are used to generate the least significant bit for the y-coordinates. At first the 10 MHz clock is reshaped by a dividing circuit. The output of this circuit delays the FD signal using FF1. The $\overline{Q}$ output of FF1 clocks FF2 which has its D-input connected to the EF pulses. The Q output of FF2 represents the least significant bit of the y-coordinate. The timing diagram for this operation is shown in Fig. 5.28.

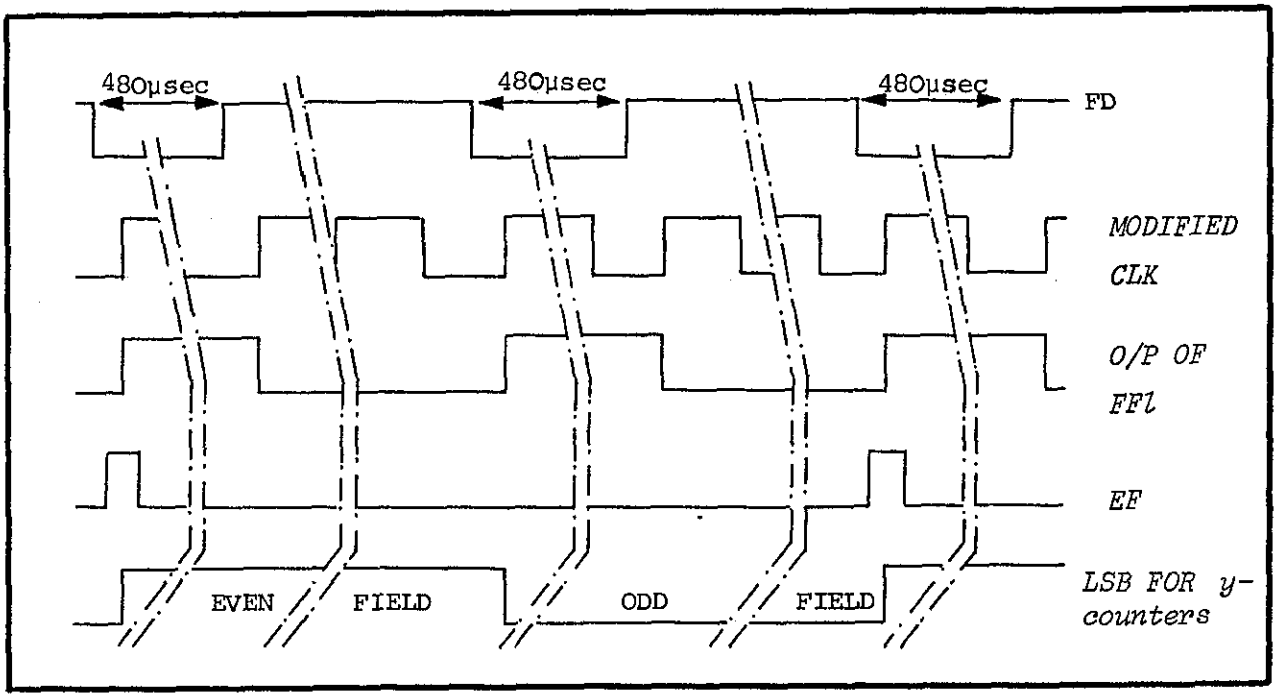


Fig. 5.28: Generation of LSB for y-coordinates.

5.6.2 FIFO Buffers

There is a serious timing constraint when reliable transfer of coordinates from counters to the microprocessor system is considered. This is due to the following factors:-

1. Edge points come at a very high rate (a maximum of 10×10^6 samples/second for adjacent edge points on the same horizontal line). The microprocessor, with its 3 MHz clock, is too slow to collect such data.

2. There are two sets of coordinates (the x- and y-coordinates) and both are expressed in 10-bit format. Hence the microprocessor has to store them one after another i.e. two storage sequences for each edge element.

3. Since the coordinate generator and the microprocessor are asynchronous, provision has to be made for handshaking to achieve reliable data transfer which adds to the timing constraint.

The use of a FIFO, or first-in first-out, memory provides an easy-to-apply, relatively inexpensive alleviation of this timing constraint. They have the ability of accepting data at one rate and emitting it at another, and provide flags to indicate the presence of data and the availability of empty storage locations.

The FIFO's used for buffering the x- and y-coordinates (Fig. 5.27) are Monolithic Memories Incorporated 67401, 64 x 4 FIFOs. They are Schottky Bipolar devices and are capable of 20 MHz shift in, shift out rates. 3 FIFO's are connected to each counter to provide parallel transfer of data. In effect, coordinates of up to 64 edge points can be stored in them, and made available when the microprocessor is ready. Coordinates are loaded into FIFO whenever there is an edge (B/W or W/B). Handshaking between the FIFO input and the counters is not necessary as the transfer of data between them is synchronised and it is assumed that the number of edge points per field does not

exceed 64. A flag between the output of FIFO and the microprocessor is provided. The microprocessor first tests whether data is available or not through its CRUIN-4. If data is available it stores it in succession, first the y-coordinate followed by the x-coordinate. Then it sends a pulse through its CRUOUT-5 which is connected to the SHIFT OUT terminal of the FIFO and the cycle is repeated. CRUOUT-4 is connected to the RESET terminal, and is used in resetting all words in FIFO memory to the empty state at the beginning of each operation involving FIFO. The whole process is controlled by CRUOUT-3 which enables the gates connected to the edge pulses and the control lines feeding the counters. Before the gates are enabled the microprocessor tests CRUIN-5 which is connected to the FD signal. After detecting the beginning of a field, the microprocessor opens the gates to allow storage of data in FIFO.

5.6.3 Input port

A block diagram of the input port is shown in Fig. 5.29. MAP1 and MAP2 are generated by the memory mapping I/O circuitry. When a microprocessor instruction addresses memory location $7FFF_{16}$ (MAP1 = 0), the data selectors select an x-coordinate, otherwise a y-coordinate is selected. When either location $7FFF_{16}$ or $BFFF_{16}$ is addressed (MAP2 = 1) the tristate input buffers change from the high-impedance state and data propagates through to the data bus.

5.7 Display Unit-2 Interface

Fig. 5.30 is a block diagram of display unit-2 interface. Data transfer from the CPU to the display unit, which is a storage oscilloscope, is controlled by three signals, MAP3, MAP4 and MAP5; generated by the memory mapping decoding logic.

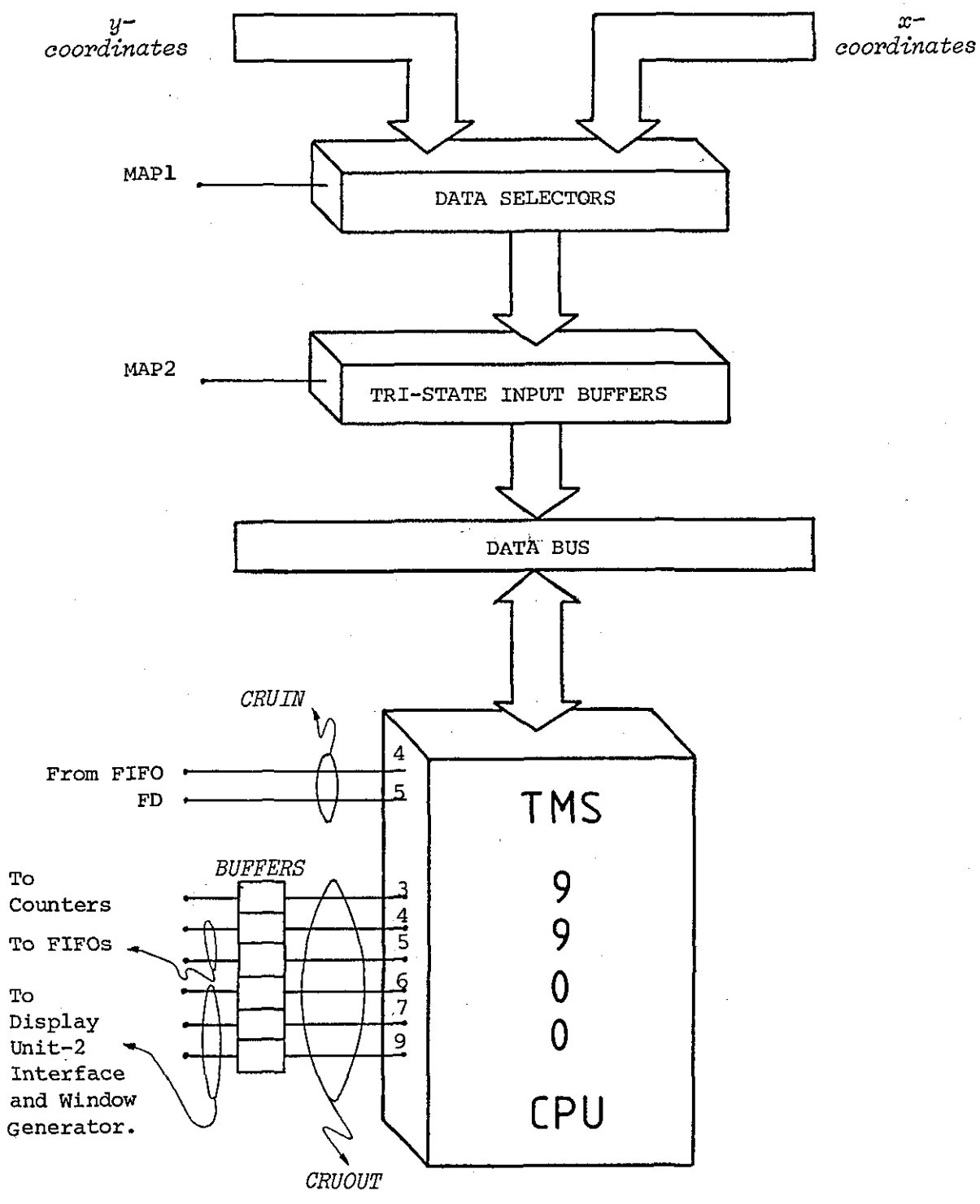


Fig. 5.29: Input Port

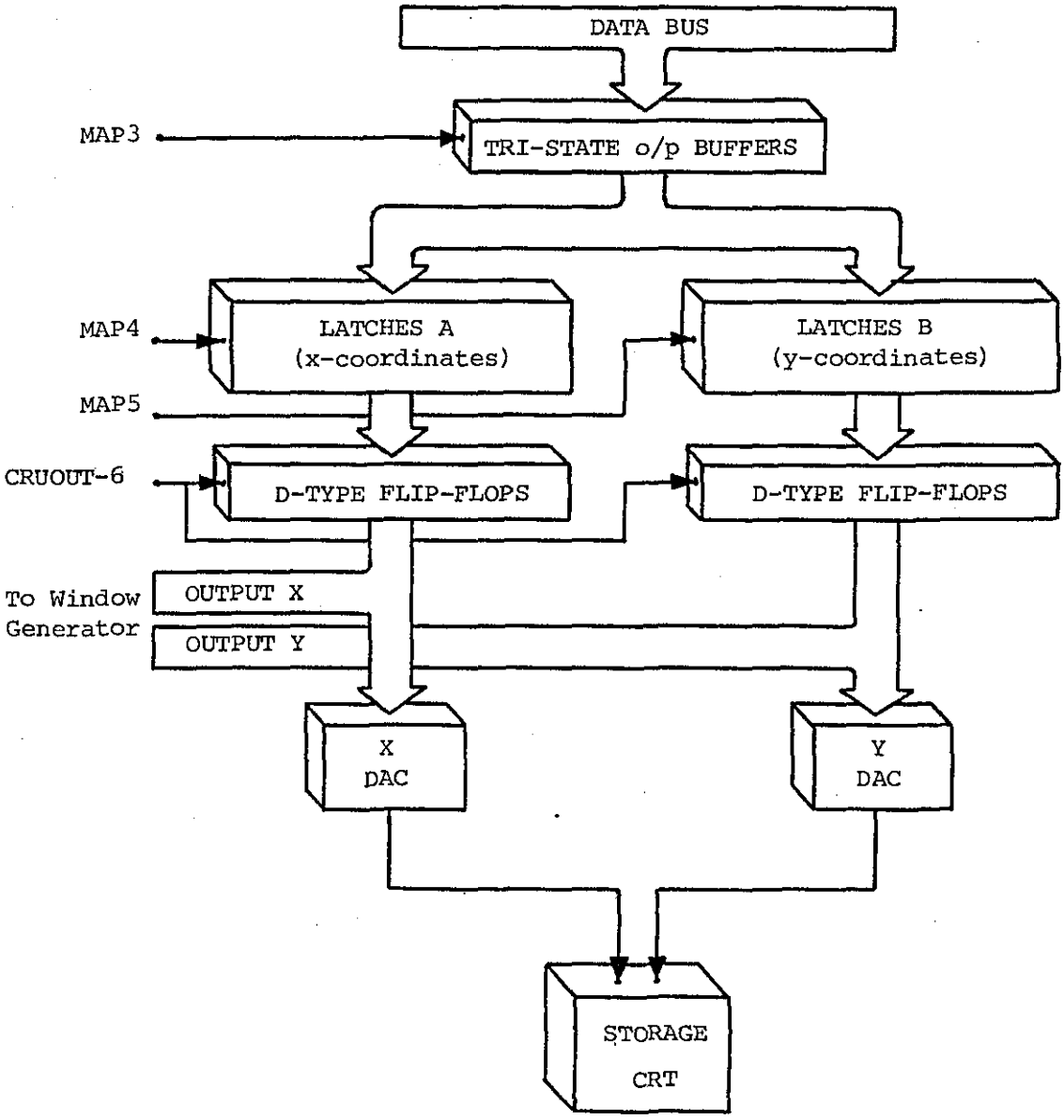


Fig. 5.30: Display Unit-2 Interface

When either memory location $AFFF_{16}$ or $6FFF_{16}$ is addressed (MAP3 = 1), the tristate buffers change from the high-impedance state and allow data transfer from the data bus to two sets of latches. When the address is $6FFF_{16}$ (MAP4 = 1), latches A are enabled and data appears at their output. Similarly, when the address is $AFFF_{16}$ (MAP5 = 1), latches B are enabled and data flows to their output. Thus data is separated into two different channels, one leading to the x-input and the other to the y-input of the storage oscilloscope. When data is available at the inputs of both D-type flip-flops, the microprocessor sends a pulse through CRUOUT-6 causing data to be routed simultaneously to two DAC's (digital to analogue converters) and from there to the two inputs of the storage oscilloscope. The CPU also uses the display unit-2 interface to send values of x- and y-coordinates to the window generator.

5.8 Window Generator

This subsystem is used to generate a window to blank the whole raster except for a rectangular area. The window is generated for the following reasons:-

a) Sometimes spurious edge points due to contouring effects (chapter 4) appear on the periphery of the raster. The window is used to eliminate these spurious points.

b) Because of the limited capacity of FIFO the microprocessor can store coordinates of only 64 edge points per field. The window provides a means by which the CPU can store a stationary pattern represented by any number of edge points, the limiting factor here being the capacity of the memory. This is achieved by shifting a small window successively from the

top to the bottom of the raster and each time storing edges lying within the window.

c) The window can also be used to display processed coordinates on the raster at a rate of one edge/field.

d) The window can be used to accommodate a moving object and then lock to it.

A block diagram of the window generator is shown in Fig. 5.31. Values of x_{min}, x_{max}, y_{min} and y_{max}, defining the borders of the rectangle, are transferred from the CPU, via the display unit-2 interface and are stored in the D-type flip-flops. The process is synchronised by pulses sent through the CRUOUT-bit 7 and CRUOUT-bit 9 lines. Outputs from the flip-flops are fed to 4 comparators, the other inputs of comparators being connected to counters generating x- and y-coordinates of the raster scan. Outputs $x>x_{min}$ and $x>x_{max}$ are EXCLUSIVELY-ORED together to generate the width (W_d) of the window (Fig. 5.32). Similarly, $y>y_{min}$ and $y>y_{max}$ are EXCLUSIVELY-ORED to obtain the height (W_h) of the window. Referring to Fig. 5.31, W_d and W_h are ANDed together and the resulting signal W_w can be ANDed with the edge pulses B/W and W/B to exclude undesirable edge points. W_w is sent to display unit-1 to blank corresponding areas in the digitised video picture.

To reproduce an edge point the CPU sends the stored x- and y-coordinates of the point along the same path as x_{min} and y_{min}. Resulting $x>x_{min}$ and $y>y_{min}$ are routed to monostables to generate pulses (P_x, P_y) with a duration of 100 nsec which is the same as that of elements of the digitised video signal. These pulses do not occur at the same time as shown in the timing diagram. For this reason P_y is fed

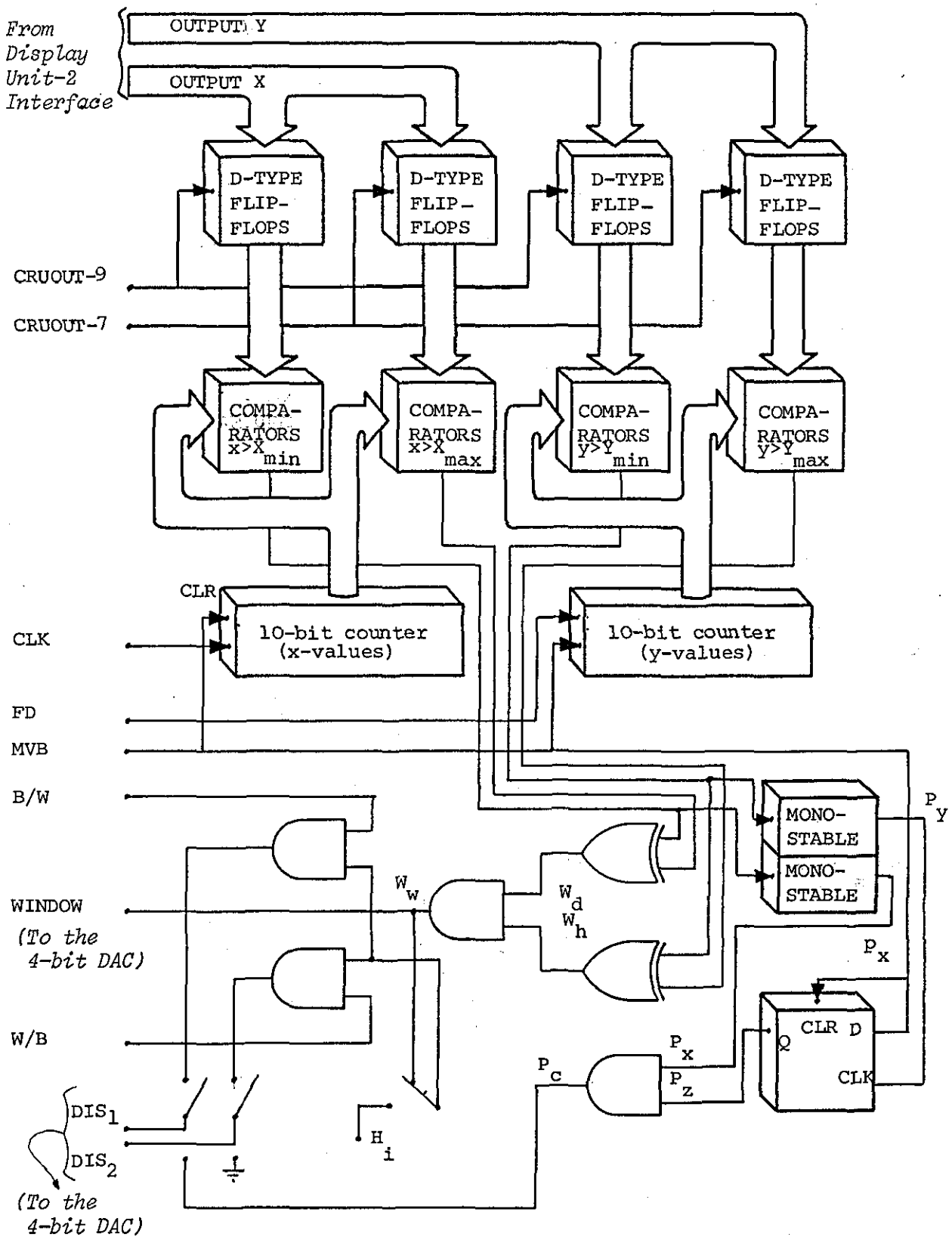


Fig. 5.31: Window Generator

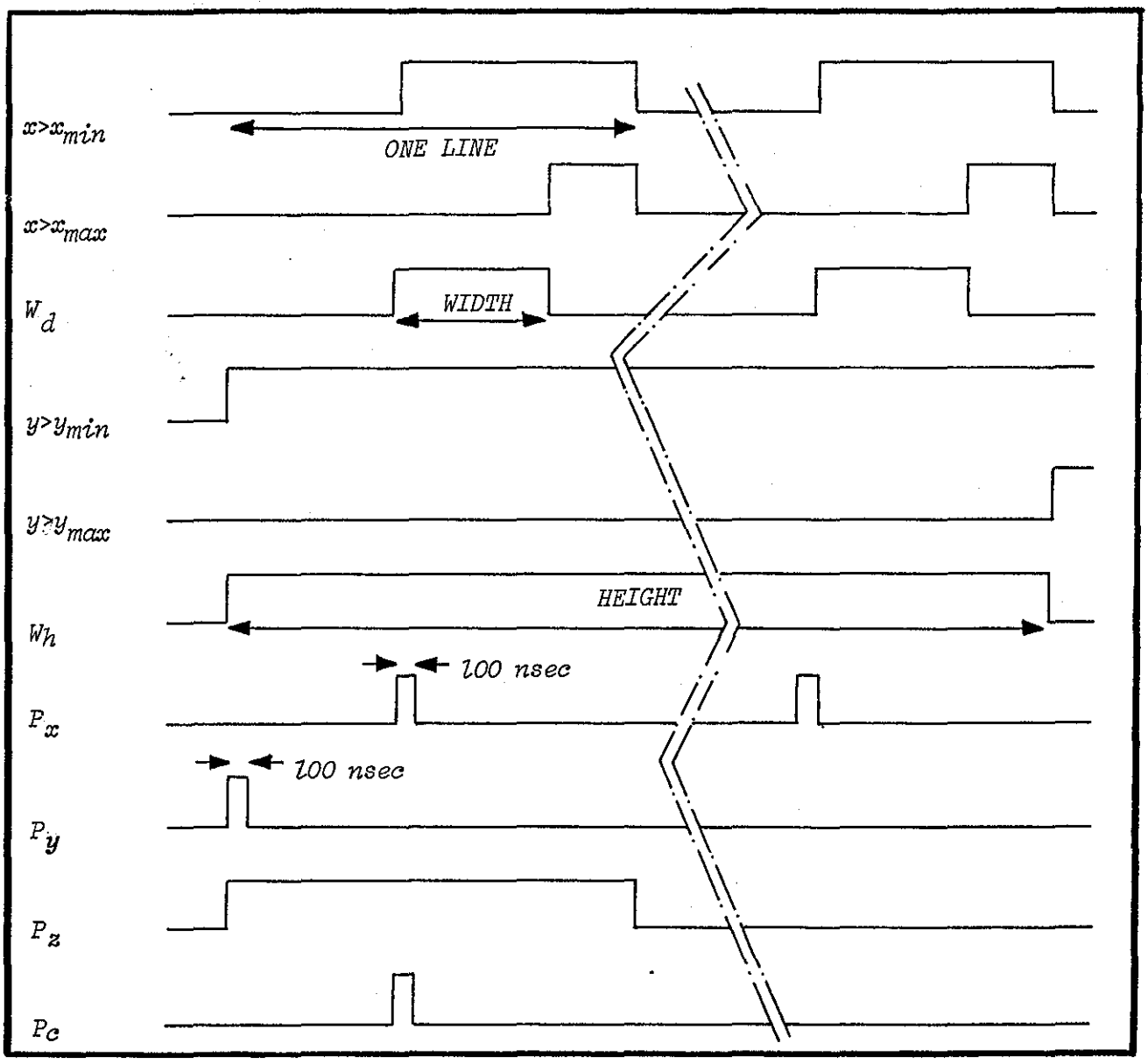


Fig. 5.32: Timing Diagram for Window Generation

to a D-type flip-flop which has its D-input and CLR connected to the MVB signal, to provide a pulse P_z representing the line on which P_x lies. This pulse is ANDed with P_x to obtain P_c which indicates the exact position of the edge point on the raster. A toggle switch is provided and it can be set to connect either edge pulses (W/B, B/W) or P_c to display unit-1 (the 4-bit digital-to-analogue converter).

5.9 Display Unit-1 Interface

A schematic diagram of display unit-1 is shown in Fig. 5.33. This unit functions as a 4-bit DAC, a mixer to reinstate the video sync information and an amplifier driver stage with a 75-Ω output impedance matching a 75-Ω cable connecting the unit to a TV monitor. The operator can select the appropriate video signals to be displayed by setting 8 toggle switches on the front panel of VAS. Six of these switches are shown in Fig. 5.33, and the rest are in Fig. 5.31. The display covers the whole raster or part of it depending on whether a window is imposed on the video signals or not.

The information displayed can be either of the following:-

1. Digitised video picture. Provision is made to display any one or a combination of the 4-bits representing the digitised video signal.
2. Edges of video images.
3. Reproduced edge points.

5.10 Power Supply Unit

The power supply unit generates from the mains supply all the D.C. power required by VAS. It is built using transformers, semiconductor rectifiers and regulators. The outputs of the regulators are filtered to provide +20V, -20V, +15V, -15V, +12V, -6V and -5V. The unit also incorporates a Gould MGT5-10 power supply to provide the high current required for the +5V output.

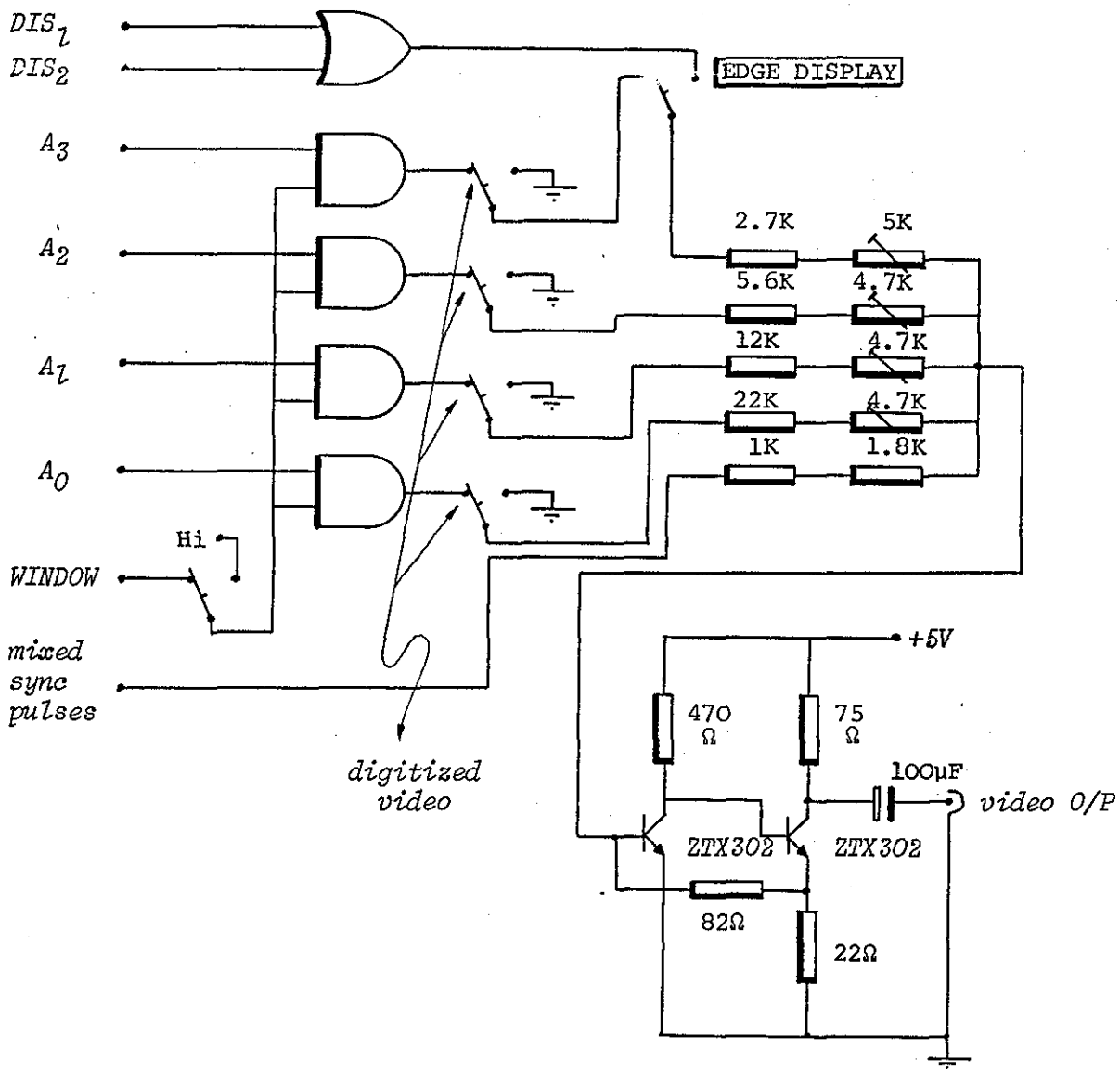


Fig. 5.33: Display Unit-1 Interface.

CHAPTER 6

PROGRAMS FOR STORAGE AND DISPLAY OF EDGE POINTS

6.1 Introduction

This chapter discusses general application programs for storage and display of edge patterns extracted from the video image. Excluding the program for storage of edge points, these programs are not executed in real time. Real time programs for video image processing, presented in other chapters, are goal-guided towards the solution of particular problems and their design is based on a priori knowledge of the shapes of input patterns. Requirements for such programs are covered in detail at the end of this chapter.

The general application programs described here perform the following functions:-

1) Generation of a window to blank the whole raster except a rectangular area. Edge points within this area can be collected and stored by the microprocessor.

2) Display of raw or processed edge points on a storage oscilloscope and a TV monitor.

3) Generation of a hardcopy display of raw or processed data as a print-out on the Silent 700.

4) Loading of edges belonging to even or odd fields into memory.

All these programs are interactive and data transfer between the peripheral and the microprocessor is accomplished by accessing relevant subroutines in the monitor. Other programs have been designed to test the reliability of the software-hardware interaction of the system and to perform simple data manipulation operations. They are not covered in this thesis due to space restriction.

In general, programs when described briefly, here or elsewhere in this thesis, are deceptively simple, but when expressed in terms of an assembly language they become lengthy and rather complicated. The main advantage of an assembly language over a high-level language, is the efficiency in the number of memory words required to hold the software. Software coded in assembly language also minimises processing times and if the input/output timing is critical, the software needed to handle this timing can be controlled more directly[68]. High-level languages typically require 2-10 times as much memory as optimised assembly languages and run 2-100 times slower. But on the other hand development time can be cut from a half to a tenth of the time required for assembly language programming[63].

6.2 Window Generation

A flowchart of a subroutine for the generation of a window, labelled WINGEN, is given in Fig. 6.1. WINGEN starts by sending queries to the operator about the size of the window. The operator responds by printing the maximum and minimum values of the x- and y- coordinates of the outline of the window, thus specifying the size and location of the window on the raster (Fig. 6.2). The window is used

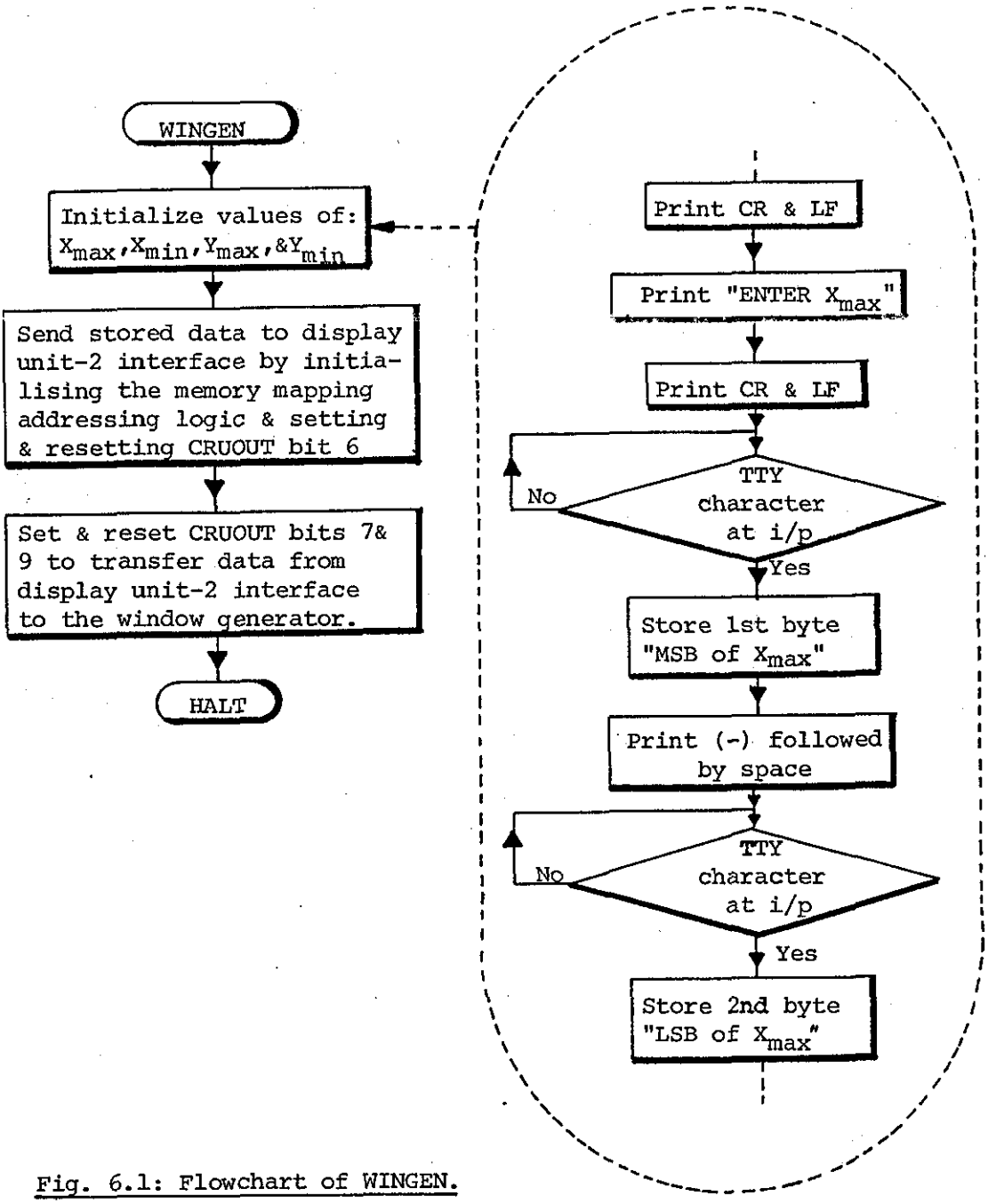


Fig. 6.1: Flowchart of WINGEN.

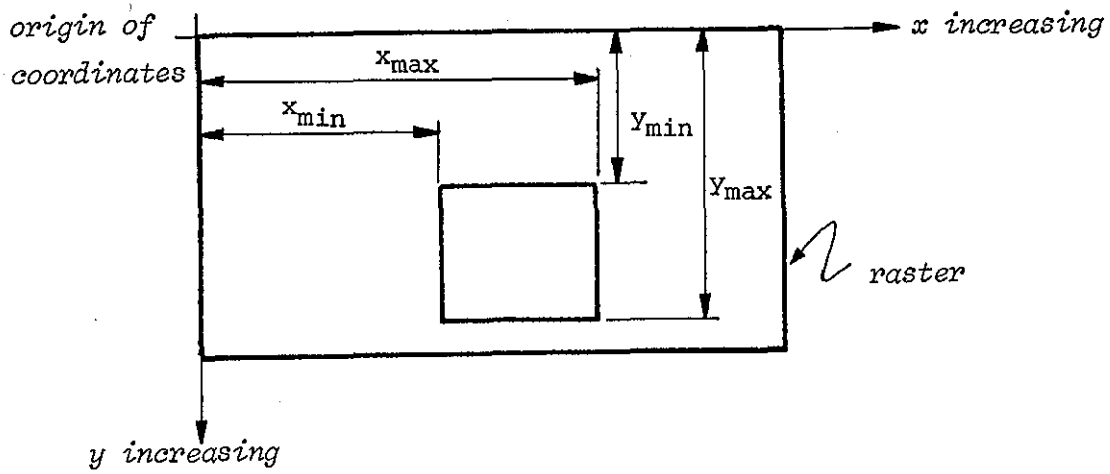


Fig. 6.2.

to blacken parts of the raster which contain irrelevant edge points. The software-hardware interaction involved in generating the window is explained in Section 5.8.

WINGEN can be modified and used in conjunction with a data input subroutine to store stationary patterns represented by more than 64 edge points, the maximum number that can be accommodated in the FIFO buffer store. (Section 5.6.2). An interactive program to accomplish this has been developed and tested. The idea is to generate a window of a suitable size, such that when it is overlaid on any part of the pattern, the total number of edge points it contains is less than 64. The program places the window at the top of the raster and stores edges within the window. Then it shifts the window progressively downwards until the whole raster is covered and after each shift it stores corresponding edge points.

6.3 Display of Edge Patterns

Fig. 6.3 shows a flowchart of a subroutine, labelled VIDUD, used to display edge points simultaneously on a storage oscilloscope and/or a TV monitor. Before edges are displayed, the program prompts the operator to answer queries about the initial and final addresses of the file where data is stored.

The storage oscilloscope receives the coordinates of edge points from the microprocessor via the display unit-2 interface circuitry (Section 5.7). The same set of coordinates are processed by the hardware responsible for window generation and then relayed to a TV monitor. To display edges on the monitor is not a straightforward

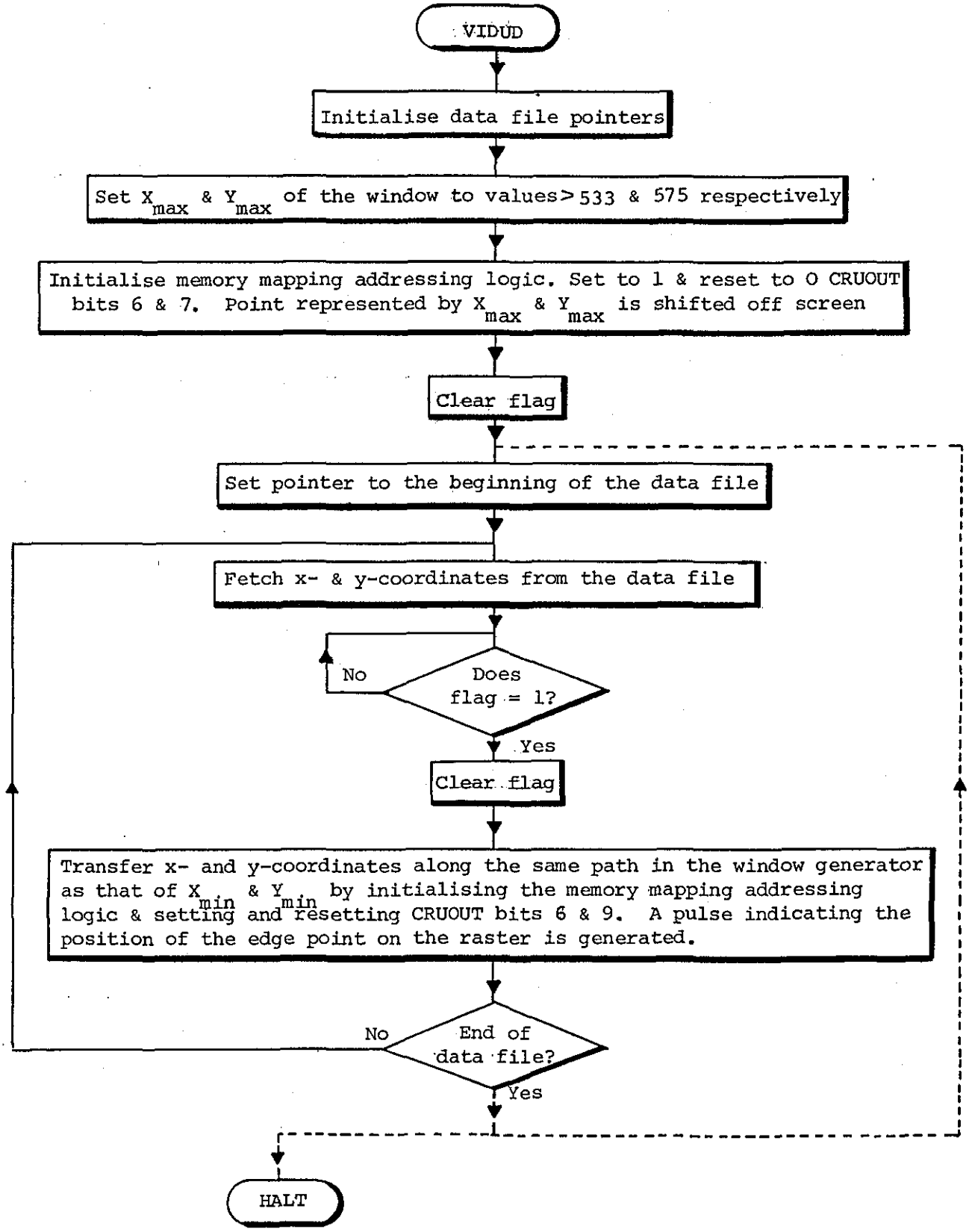


Fig. 6.3: Flowchart of VIDUD

procedure. VIDUD; at first, sets X_{max} and Y_{max} to values greater than 533 and 575 respectively, so that the point at which they intersect is shifted outside the screen (Fig. 6.2). Then the x- and y-coordinates of an edge point are fetched from the data file and transferred along the same path, in the window generator hardware, as that of X_{min} and Y_{min}. The point of intersection of these two coordinates represents the location of the displayed edge on the raster. A white pixel is generated at this location to represent the edge point, as explained in Section 5.8. The CRUOUT bits and the memory mapping addressing locations involved are the same as those used in WINGEN.

Edge points are displayed at the rate of one per field because the microprocessor is too slow to send two edge points separated by a few microseconds without loss of synchronisation with the TV raster scan. A flag flipflop is used to achieve synchronisation. The microprocessor, under program control, tests the output of the flipflop and initiates the display operation only when the output of the flipflop is set to logic 1.

Fig. 6.4 shows the flag circuit. When the flipflop is clocked by the field drive the flag is set to logic 1. The flag is looked at through CRUIN bit 6. The microprocessor tests this bit and if it senses a high level, it clears the flipflop by resetting CRUOUT bit 8 to zero and then enables it by setting the same bit to 1. Following this it displays an edge point on the storage oscilloscope and the monitor and then waits for the flag to be set to 1 by the next field drive pulse.

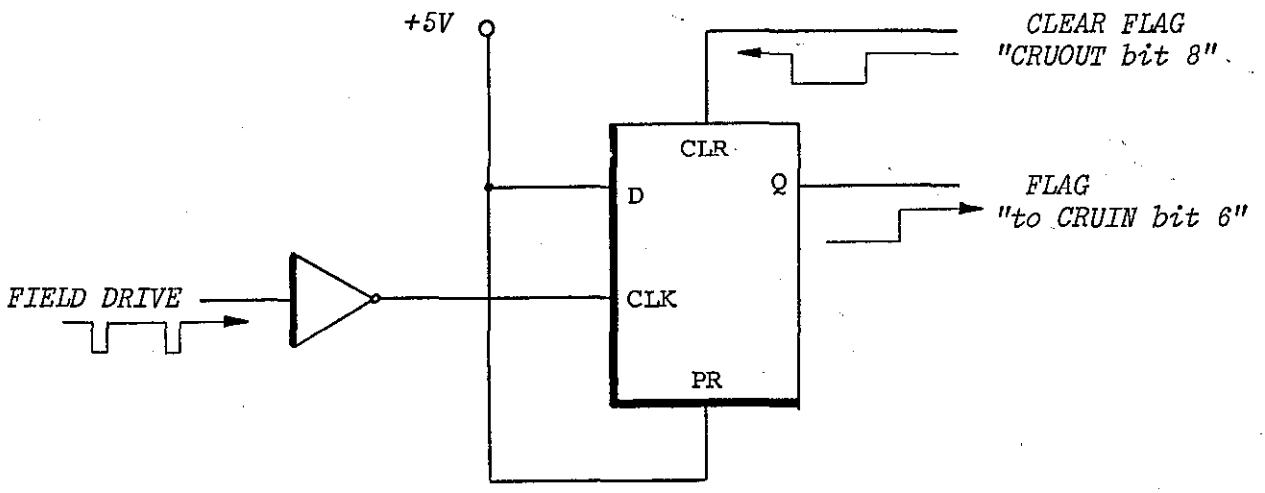


Fig. 6.4

VIDUD can display an edge pattern once or continuously depending on whether the last instruction is a HALT or a BRANCH instruction. Edge points appear permanently on the storage oscilloscope, whereas on the monitor they appear intermittently due to the rate at which they are displayed.

6.4 Hardcopy Display of Edge Patterns

A subroutine, labelled HARDED, is used to obtain a hardcopy display of edge points as a print-out on the Silent 700. A simplified flowchart of HARDED is shown in Figures 6.5 and 6.6. Upon receiving information about the location of the data file, HARDED examines the x-coordinates of the edge points representing the stored pattern to

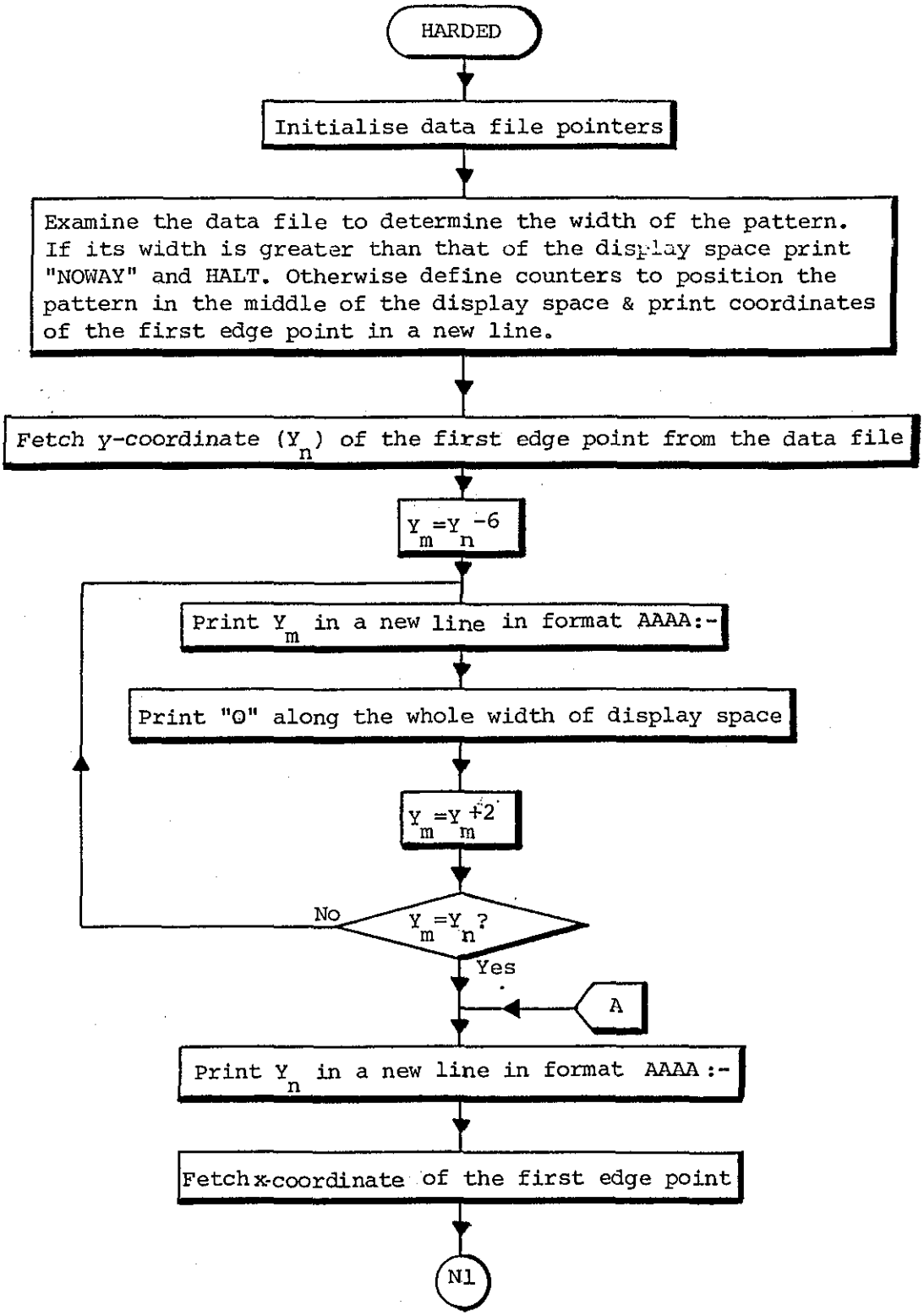


Fig. 6.5: Flowchart of HARDED

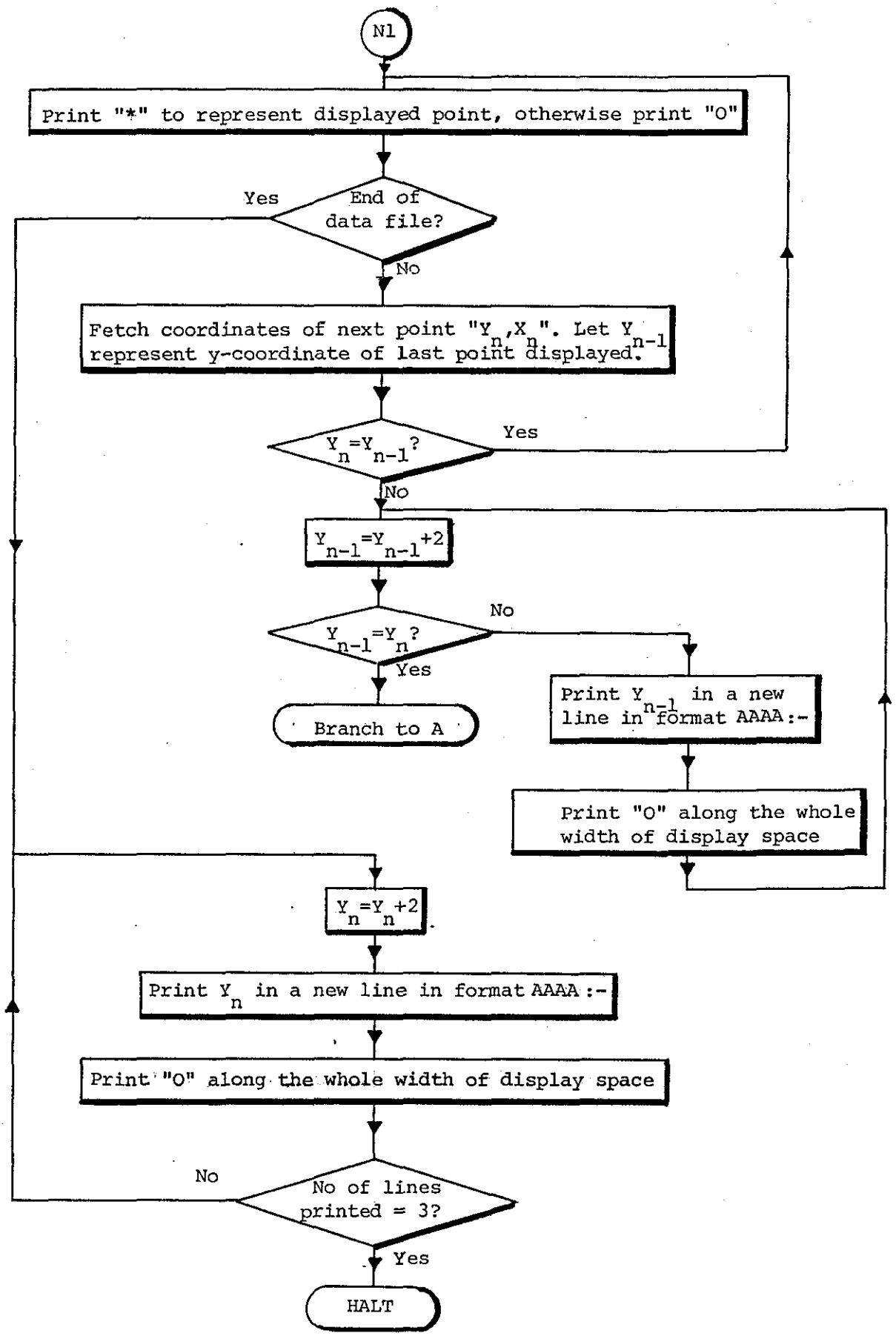


Fig. 6.6: Flowchart of HARDED (continued)

determine their maximum and minimum values (X_{max}, X_{min}). It subtracts X_{min} from X_{max} and if the result is greater than the width of the display space(W_s) a message is sent to the operator. The operator can decrease the size of the pattern by dividing the coordinates of its edges by a constant (2^n). This division is performed by running an interactive subroutine called SCACO. If the result of the subtraction is less than W_s, HARDED sets two counters to define the minimum and maximum x-coordinates of the display space such that the displayed pattern occupies the central part of this space. The minimum and maximum coordinates, S_d and F_d respectively, are given by:-

$$S_d = \frac{X_{min} + X_{max} - W_s}{2}$$

$$F_d = \frac{X_{min} + X_{max} + W_s}{2}$$

The program starts the display by printing the y- and x-coordinates of the first edge point displayed to act as a reference. y-coordinates of each horizontal line are printed on the left hand side of the display space in the format:

AAAA:-

All numerals in a hardcopy display are expressed in hexadecimal notation. As an example consider the hardcopy display of the edges of a circle shown in Fig. 6.7. The two numbers (0105_{16}, 0094_{16}) on top of the display space are the coordinates of the first edge point.

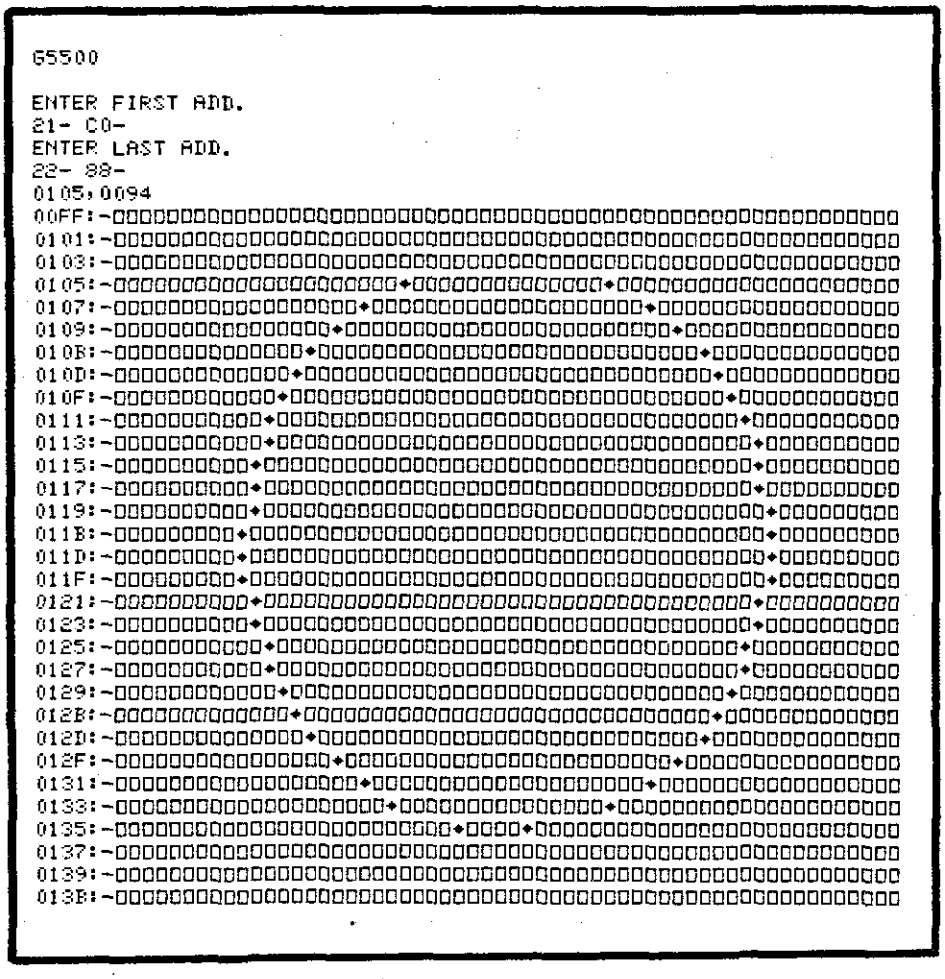


Fig. 6.7: Edges of a Circular Disc

The x-coordinate (the second number) can be used to determine the x-coordinates of other edge points. As the y-coordinates are odd numbers it follows that the pattern belongs to an even field. 3 lines are printed on top and bottom of the displayed pattern to make the display look better. The spacing between characters is greater in the vertical direction than in the horizontal one and as a consequence the pattern is slightly elongated in the vertical direction.

Concerning the width of the display space, the maximum number of characters which can be printed by the Silent 700 in a horizontal line is 80. Six of these characters are used to indicate the y-coordinates and hence the width allocated for the display space is equivalent to 74 characters. This width can be varied by the operator by changing the content of a counter in HARDED.

6.5 Loading of Edges Belonging to Even or Odd Fields into Memory

The subroutines discussed so far involve outputting data from the microprocessor system. The subroutine explained in this section stores data into the system and entails interaction with the FIFO buffer store circuitry and the input port of the microprocessor shown in Fig. 5.29. Such interaction is typical of all subroutines, discussed in subsequent chapters, for storing edge points.

The subroutine presented here is labelled FIDAT and is used for storing edge patterns belonging to even or odd fields into the system memory.

A flowchart of FIDAT is shown in Fig. 6.8. If the operator wants the microprocessor to store data coming from an odd field he has to print 0 in response to the subroutine query. For an even field the number entered is 1. These two numbers are employed by the microprocessor to decide whether incoming data belongs to an even or an odd field by examining the least significant bit of the y-coordinate of the first edge point. After defining the field the operator specifies the initial address of the file where he wants to store data.

The sequence of operations outlined in the flowchart and their functions can be understood by referring to Section 5.6. FIDAT can

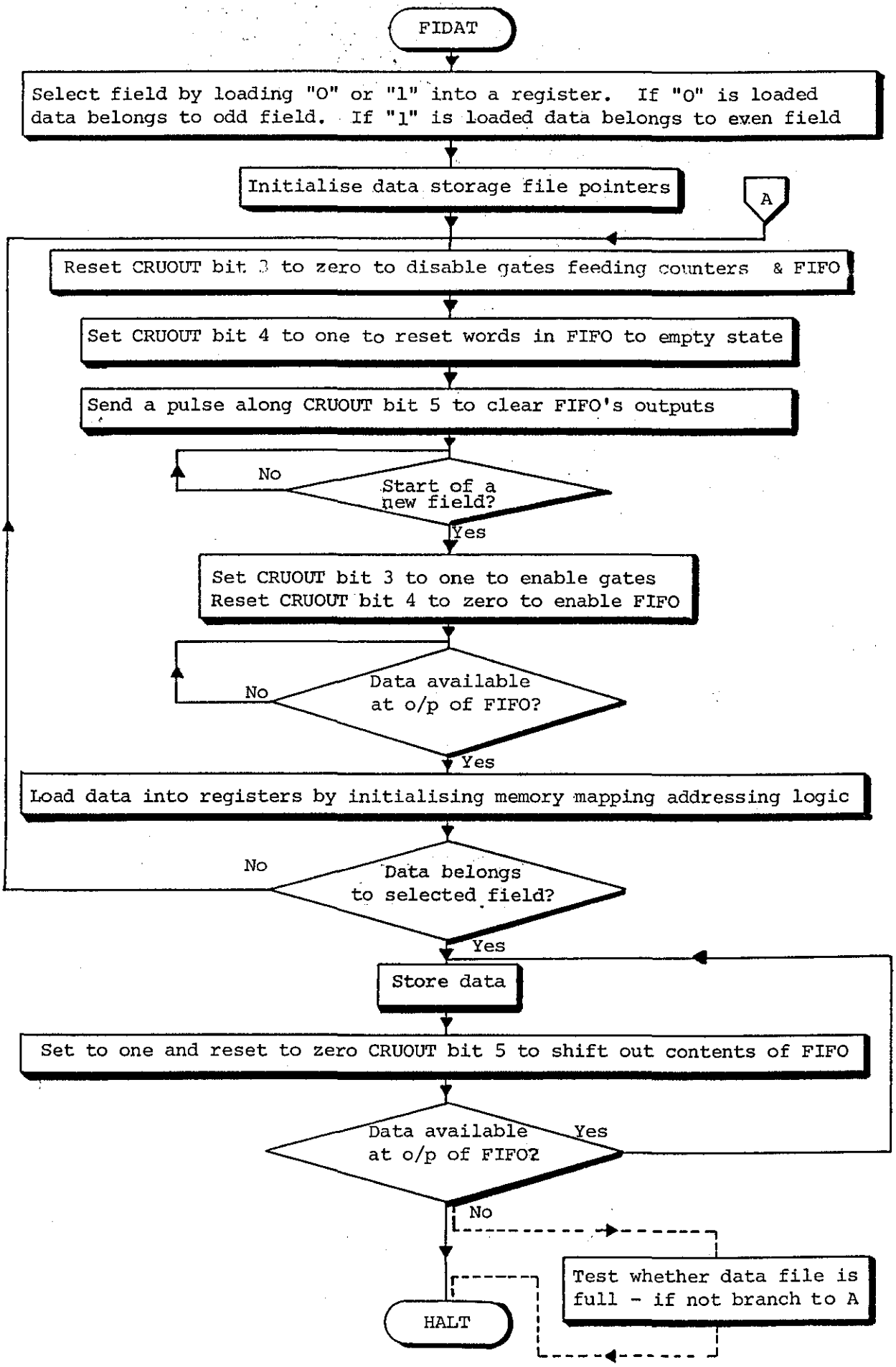


Fig. 6.8: Flowchart of FIDAT

be used to store one field or more by changing a few instructions at the end of the subroutine. The instruction to test whether data belongs to an even or an odd field can be skipped to store data from both fields in succession.

6.6 Requirements for Real Time Programming

A real time video analysis system receives data belonging to a field or a frame, processes it and returns results before data belonging to the next field or frame arrives. Because of the time constraint involved a real time program must be carefully designed and optimised to accomplish its task within the allowable time. The testing of the programs and the overall system is much more difficult for a real time than a batch system. The greater difficulties are due to factors such as the random nature of events in real time systems, the high input/output activity and fluctuations in the input rate. In batch processing systems, if an error occurs during operational running it is possible to stop the run and look for the cause of the error. This is not possible in a real time system, and because of the changing time relationships more subtle errors will exist which show up infrequently. In general, errors are realised after the operation during which they occur is over and this makes it very difficult to pin-point them. In view of these problems, real time systems impose extra reliability requirements compared to batch systems.

Fig. 6.9 illustrates an overall flowchart of a real time program for video image analysis. The most significant consideration concerns the times T_{min} and T_{max}. T_{min} is the time required for data acquisition and storage. T_{max} is the maximum time allowed for processing and

display of results. Both times have some degree of uncertainty due to the unpredictability and fluctuations of the input data. In real time video, if processing is performed on fields, then T_{max} is equivalent to the field duration minus T_{min}. Concerning VAS the maximum number of edges processed per field is 64. Assuming that the input edge pattern is symmetrical about the vertical axis, i.e. it covers 32 horizontal scan lines, then the maximum time interval between the first edge point representing the pattern and the last one is in the order of 2 msec. If the pattern is stationary then the time allowed for the execution of a real time program is about 18 msec (Fig. 6.10). If the pattern is moving, this figure will fluctuate considerably. In the extreme case if the pattern continues to change position rapidly from the top of the raster to the bottom and vice versa, the time interval between two edge patterns might fall to a few microseconds. In VAS changes in the time interval are not realised by the microprocessor due to the buffering of the input data with FIFO's.

The speed of a processor is an important factor for determining the time required to perform specific tasks in a real time system. TMS9900 is not a very fast processor. For instance a simple add takes from 4.7 to 10 μsec and multiplication or division takes from 18 to 42 μsec, depending on the addressing mode. For this reason greater care should be taken when designing a real time video analysis program. Whenever possible, instructions which take a long time to be executed are avoided and each program is dedicated to the solution of a particular problem.

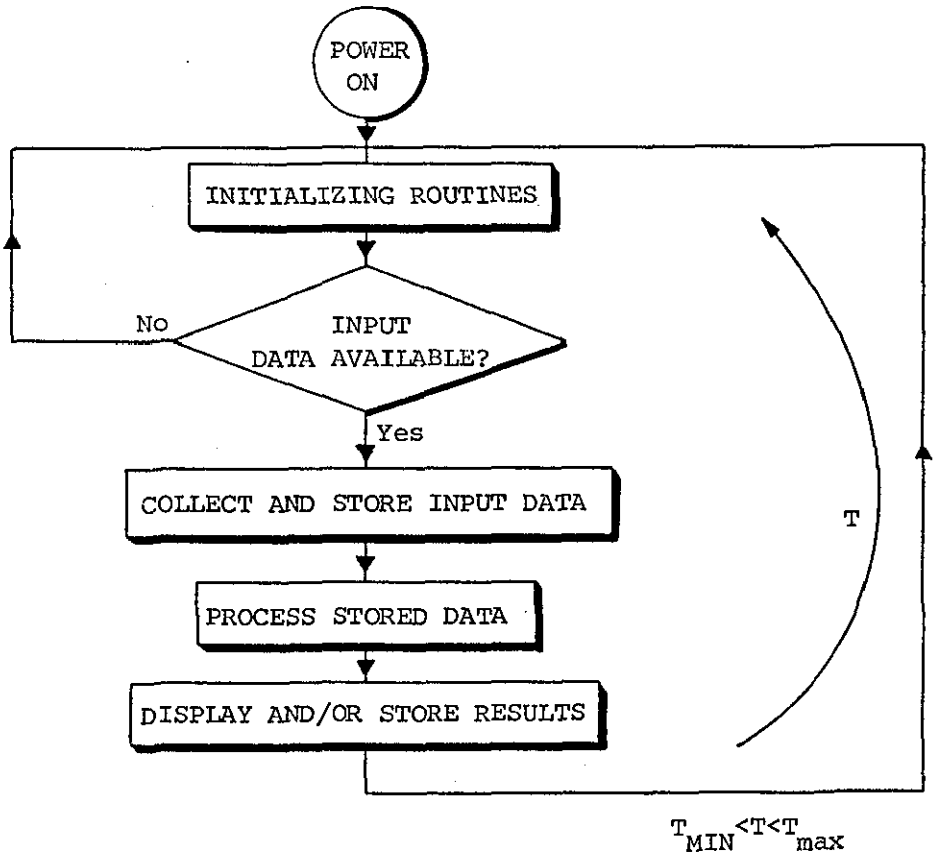


Fig. 6.9: Overall Flowchart of a Real time Program for Video Analysis

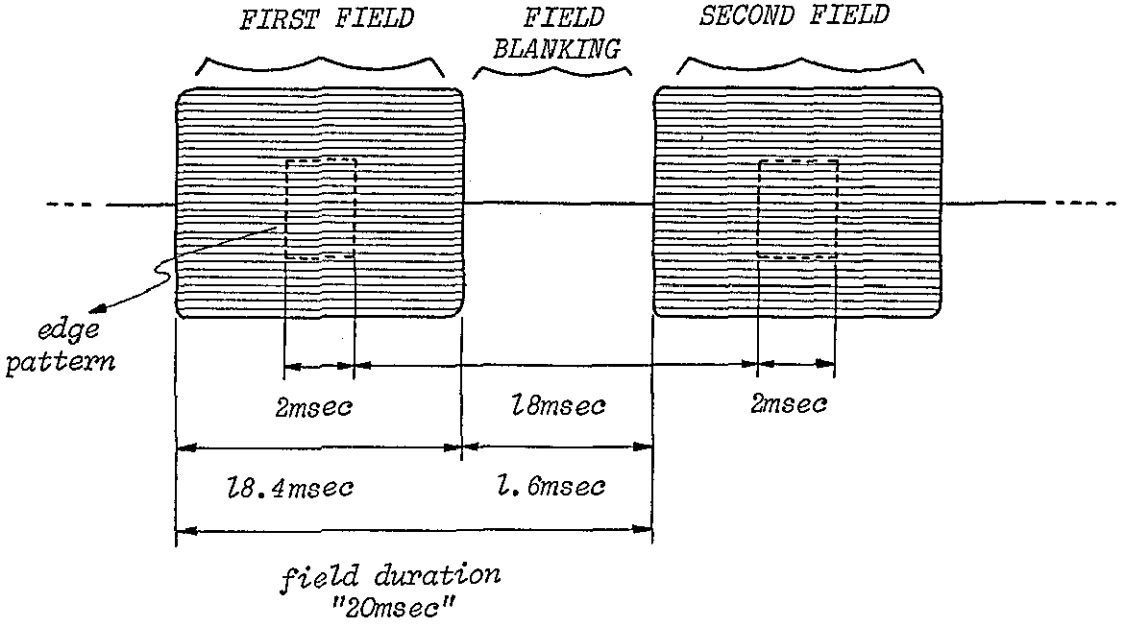


Fig. 6.10: Timing Diagram for Real time Processing of Video Signals

To increase the throughput of the system, the microprocessor performs some processing on the incoming data, guided by a priori knowledge of the shape of the input pattern. For example, if the pattern is a circle, the microprocessor stores the sequential input data as an array in which each two consecutive edge points have the same y-coordinate (Fig. 6.11). An edge point on one half of the circumference which has no counterpart on the other half is rejected. Such singular points do not occur very often and their omission does not affect the accuracy of results. Sorting out input data before storing it saves considerable amount of data manipulation required at later stages of processing. At the same time a processing subroutine knows exactly from where to fetch relevant information. For instance, consider a subroutine for determining the y-coordinate of the centre of the circle by averaging the y-coordinates of the mid-points of vertical chords constructed on the left half of the circumference (Fig. 6.11). This subroutine knows the memory locations in the data file which contain coordinates of edge points representative of the left half of the circle. By rejecting singular edge points and storing data in a structural form, laborious and time consuming procedure to isolate relevant edge points is avoided.

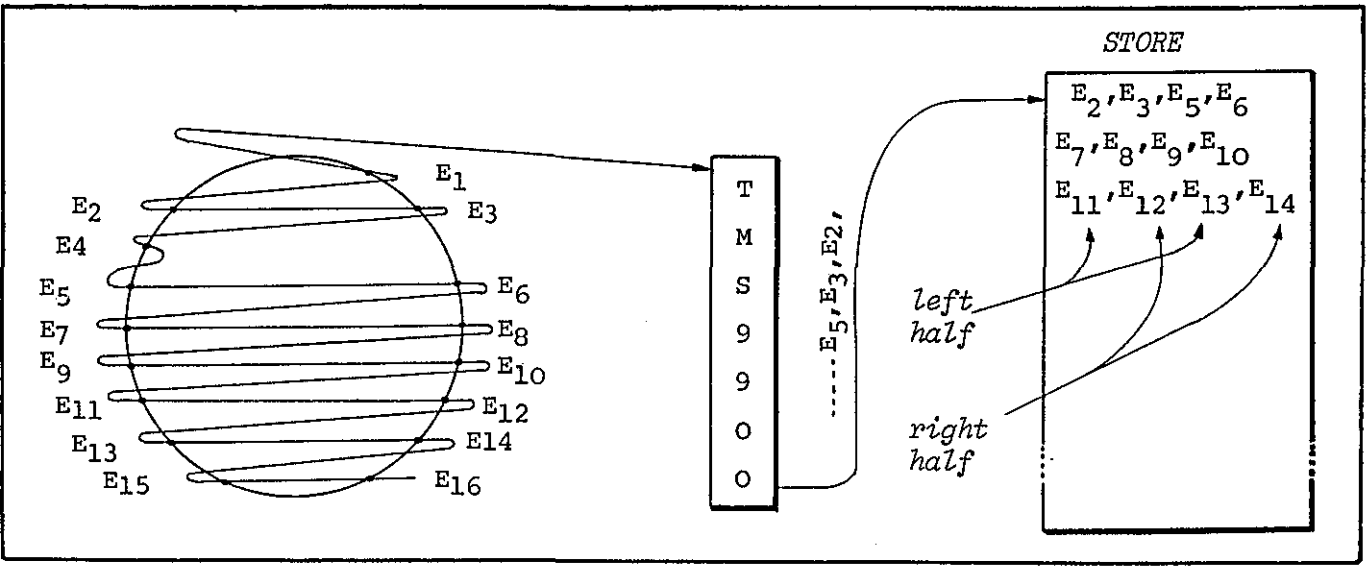


Fig. 6.11

CHAPTER 7

TRACKING OF MOVING OBJECTS

7.1 Introduction

In considering the problem of tracking moving objects by computer, three factors determine the complexity of the tracking algorithms:

1) the structure of objects in the image;

2) the number of objects; and

3) the type of motion.

In the absence of simplifying assumptions, these factors act individually or concurrently to create many difficulties that render the formulation of a reasonably general computational solution to the tracking problem a formidable task.

Finding objects in a scene and discovering which objects in previous scenes correspond to them are very hard problems, augmented by changes in appearance through complex and nonstationary background/foreground situations. In time-varying images, an object might move and rotate in a three-dimensional world such that its shape, size and structure change from frame to frame, and as a consequence the task of locating and identifying objects in the image sequences becomes difficult to accomplish in a manner that is both efficient and general. It is appropriate to note that, even when shape, size and structure are identical in consecutive images, minor variations in the lighting might induce undue complications as,for instance,shadows cast by moving objects. When a scene contains more than one object

occlusion and topological changes become a major burden[22,69] for which an effective solution is not yet available.

The added complexities due to time-varying images depend to a great extent on the type of motion under study. It is apparent that requirement for three-dimensional motion is more laborious than that for two-dimensional motion because of the difficulties inherent in describing the former analytically[70]. The same argument applies to rotational versus translational motions.

Since the task of designing systems to offer general solutions to different facets of dynamic scene analysis has proved to be difficult, researchers were forced to restrict problems under investigation in various ways to make them tractable[69]. Most of the work to date in computer analysis of time-varying images utilises a priori knowledge and models of the constituents of the image. The available techniques are designed to accept limited spatio-temporal patterns as inputs; and in most cases the dynamic image is assumed to satisfy one or both of the strong restrictions that:[71] 1) the constituents show only two-dimensional movements; and 2) their shapes are rigid. In addition the depth of understanding of present systems is mainly concerned with segmentation and tracking, and so they respond to a limited set of questions[69]. Some aspects of motion as a cue to segmentation have been covered by Potter[72] under the assumption that "things" that move with the same velocity are part of the same object. He has presented a program which segments a scene into regions on the basis of displacements of discontinuities "edges" in two dimensions. Aggarwal and Duda[22] have employed rigid polygons as an idealisation of the problem of determing cloud motion. The type of motion studied

is two-dimensional and does not involve rotation. In the two aforementioned techniques and in others formulated by Roach and Aggarwal[73], Badler[74] and Wallace and Mitchel[75], there is gross oversimplification of the problems tackled. The objects used are very simple and highly stylised and the techniques have not yet been demonstrated on complex, realistic scenes. Other techniques have been applied using real data (e.g. Thompson[76], Gilbert et al[21]). It is noteworthy to mention that no practical implementation of any of the systems proposed by the above authors has come into existence. However, the main obstacle facing dynamic scene analysis, (and static analysis); is the unavailability of general interpretation procedures, or, in the event of them being available for particular problems, the computational requirement and cost incurred[14,77].

The problem domain as defined in this research is concerned with real time analysis, and this constraint in addition to the limited processing power of the microprocessor, eliminates the feasibility of implementing techniques even with a modest degree of computation. The problems are addressed within the context of a low level of understanding by the system. Techniques employed are tailored to the capabilities of the microprocessor, which at present can provide real time information on the position of simple moving objects.

The problems chosen for investigation are tracking of a single object, tracking of two objects and identification of a moving object in a scene which, in addition, includes a stationary object. Objects tracked are rigid circular discs whose motion is planar, i.e. they have constant surface areas. The restriction in the shape of objects and the type of motion is motivated by computational convenience, and

at the same time to lay the ground for the next stage of the research, which is concerned with eye motion analysis where part of the image formation tracked is a disc (Chapter 9). Before tracking algorithms are described a brief note about image motion degradation is presented. The software-hardware interaction involved when the tracking algorithms are executed is explained in Chapter 5. The number of operations contained in a tracking program can reach 2000, or even more, and hence great care should be taken when implementing such programs in real time using the TMS9900 as it is a slow processor.

7.2 Image Motion Degradation

Most practical cameras degrade a moving picture imaged on the target in at least two ways[13]. First, a point on the target integrates all light falling on it from the time the point is scanned in one frame to the time it is scanned in the next frame. Second, the target may not be completely discharged after it is scanned and some residual signal may remain. The first source of degradation is inherent in the operation of most cameras, while the second is quite negligible in high quality cameras.

The TV camera used in conjunction with VAS to perform tracking experiments (PYE Super Lynx), is not of high quality and suffers from both sources of degradation cited above. To minimise the effects of degradation, targets are moved slowly in the field of view of the camera. Objects tracked are discs drawn on a screen fixed to the shaft of a motor. As the motor rotates patterns move in a circular path in a plane perpendicular to the axis of the shaft.

7.3 Tracking of a Single Disc

The image of the disc tracked is represented by about 100 edge points, ie 50 points per field. A flowchart of the tracking program, labelled TRACON, is shown in Fig. 7.1

Collected edge points are stored in an array to facilitate data retrieval and manipulation, as outlined in Section 6.6. The microprocessor evaluates the coordinates of the location of the centroid of the disc and displays it on a storage oscilloscope in real time. The last part of the program can be modified to store displayed data to give a motion history of the target.

Location of the centroid of the disc can be evaluated by applying the principle that perpendicular bisectors of any two non-parallel chords of a circle intersect at the centre. With raster scan devices an obvious simplification is the choice of horizontal and vertical chords as shown in Fig. 7.2(a). Here n values of coordinates of mid-points of two sets of mutually perpendicular chords are computed and then averaged to obtain the coordinates of the centre of the circle.

Another method for determining the centre of the circle is to construct horizontal and vertical tangents as shown in Fig. 7.2(b). The x- and y-coordinates of the centre are found by averaging the two horizontal and two vertical tangents respectively, i.e.

$$\text{y-coordinate} = \frac{h_1+h_2}{2}$$

$$\text{x-coordinate} = \frac{v_1+v_2}{2}$$

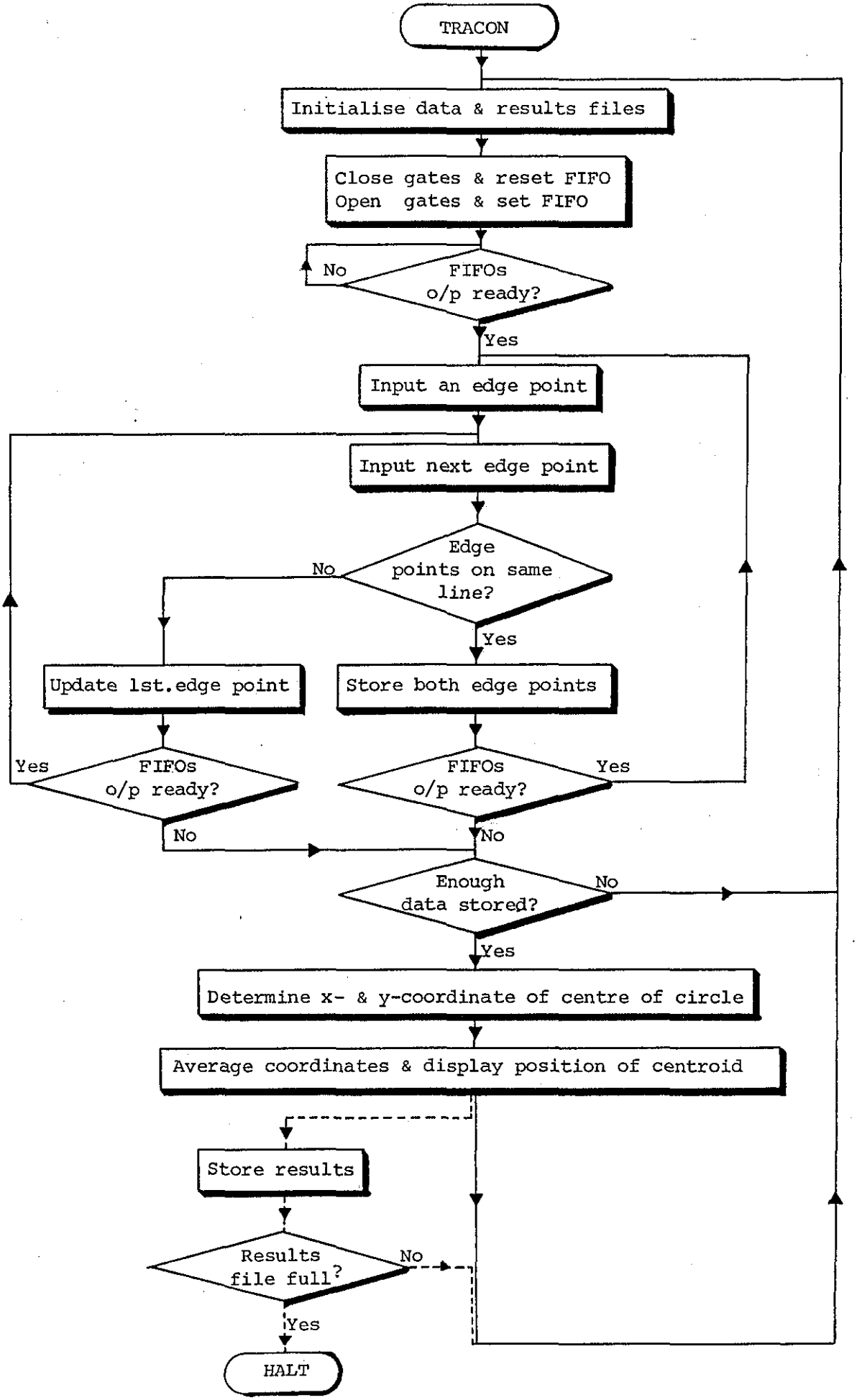


Fig. 7.1: Flowchart of TRACON

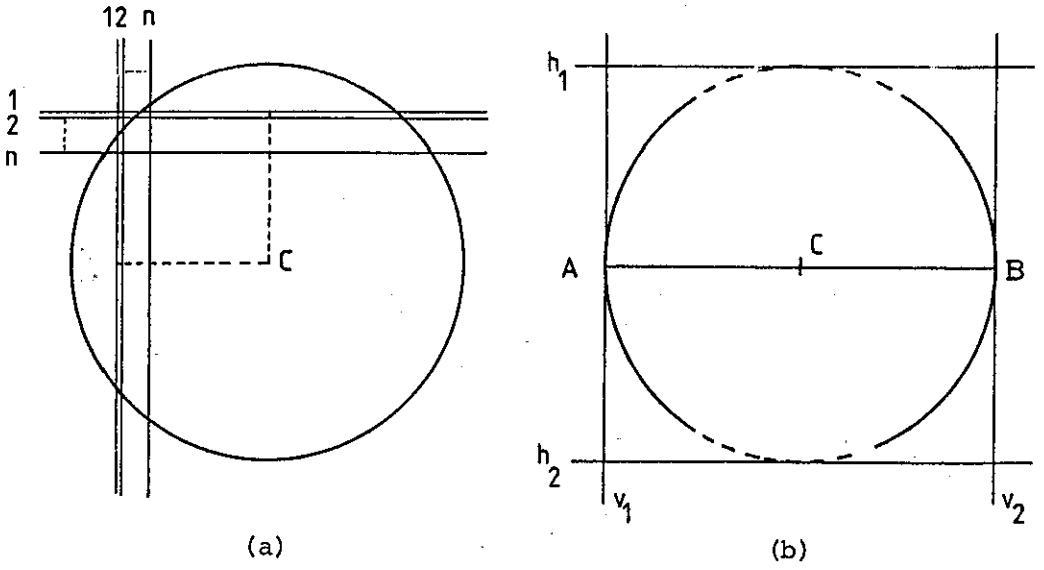


Fig. 7.2: Determination of the Centre of the Disc

VAS cannot determine h_1 and h_2 directly as edges at top and bottom of the circle, shown dotted in Fig. 7.2(b), are usually missing. This is because in extracting edges differentials are evaluated only in a horizontal direction. Consequently, before implementing this method an interpolation algorithm has to be executed to insert the missing edges, a procedure avoided because of the constraint of real time analysis. Alternatively, the coordinates of the centre can be determined by locating the mid-point of the diameter AB. Then the x-coordinate of the centre is equal to the average of the x-coordinates of the end points A, B; i.e. the average of the maximum and minimum values of the x-coordinates of the edge points of the circle, and the corresponding y-coordinate is equivalent to that of point A or B.

Both this method and the one based on mutually perpendicular chords were tested and found compatible with real time video analysis.

Concerning the method based on mutually perpendicular chords, the evaluation of x-coordinates of the mid-points of horizontal chords is a straightforward procedure in which edge points are read from the store, two at a time, and their x-coordinates are averaged to obtain the x-coordinate of the mid-point of the chord joining the two edge points. It should be recalled that in the store points with the same y-coordinate are stored in adjacency.

In relation to the y-coordinates of the mid-point of vertical chords, pointers in the program responsible for determining the centroid are adjusted such that a minimum number of locations in the data file are addressed during the process of matching points with equal x-coordinates representing the end points of such chords. Referring to Fig. 7.3 the program picks points on the upper left hand arc of the circle and searches for edge points with identical x-coordinates in the lower arc on the same side of the circle. The first point picked is A and the part of the circle examined to find the corresponding point is arc VZ. The arc where a point corresponding to B is expected to exist is YZ and for C it is XZ. The lengths of arcs examined by the program are adjustable and the choice of a suitable length is dependent on the time allocated for determining the coordinates of the centre within the constraint of real time tracking. A flowchart for that part of the program responsible for the evaluation of the y-coordinate of the centre from vertical chords is shown in Fig.7.4.

Concerning the method for determining the centre based on identifying the maximum and minimum values of the x-coordinates, a subroutine simply searches the data array for these values and averages them to

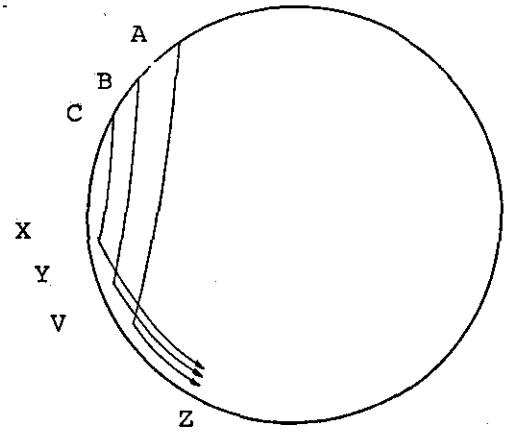


Fig. 7.3:

obtain the x-coordinate. For the y-coordinate, the procedure is not straightforward, because in that part of the circle where the minimum and maximum values of x are located, some edge points are aligned along vertical straight lines (Fig. 6.7), a consequence of taking differentials only on the horizontal direction when edges are extracted from the video signal. Thus to determine the y-coordinate of the centre of the circle, the program identifies the mid-point of these edges falling on the same vertical line. The y-coordinate of the mid-point corresponds to that of the centre of the circle.

The path of the circle displayed in real time by TRACON is shown in Fig. 7.5. Plate (a) exhibits part of the circular path and Plate (b) the whole path. Fig. 7.6 is a vector field representing

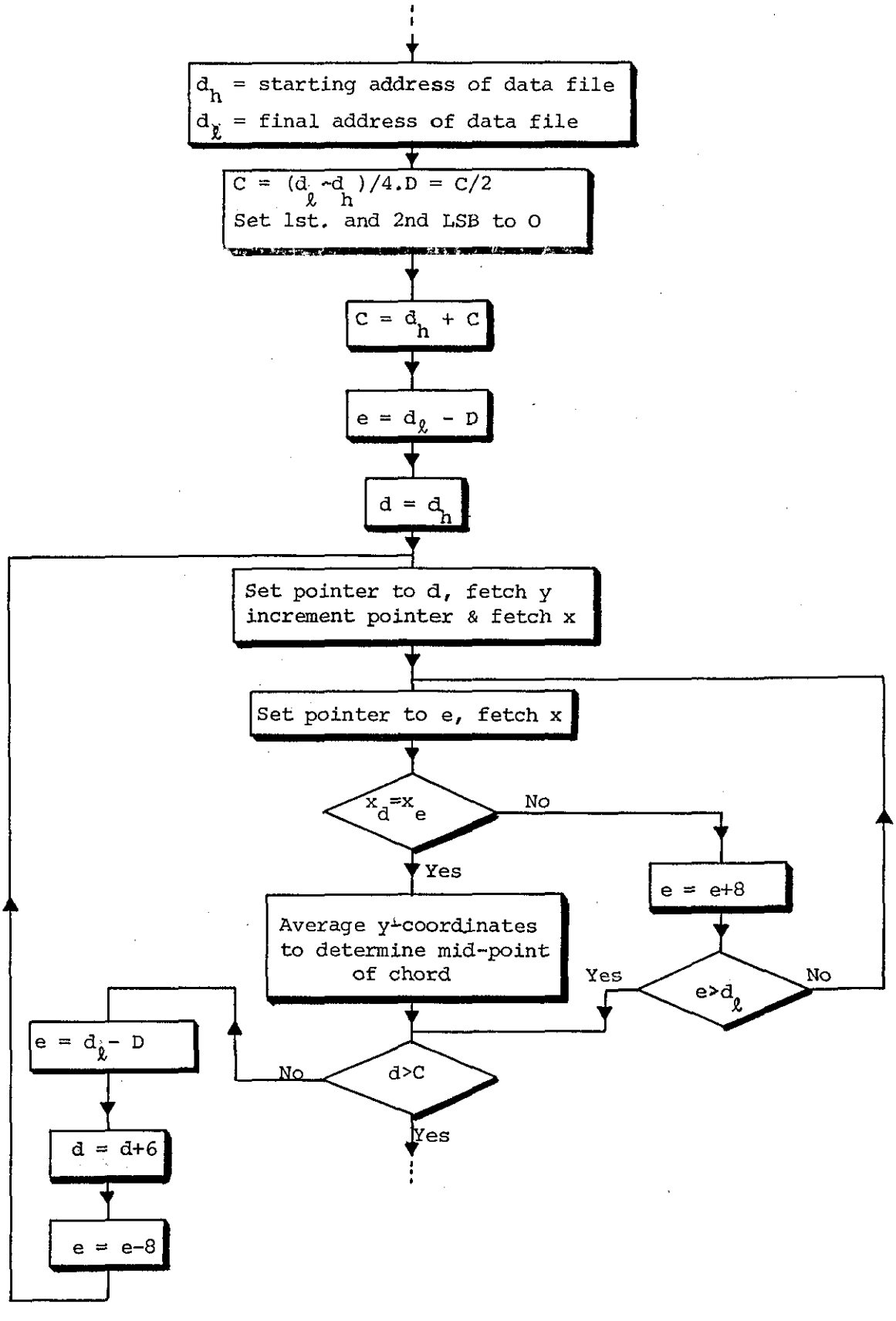
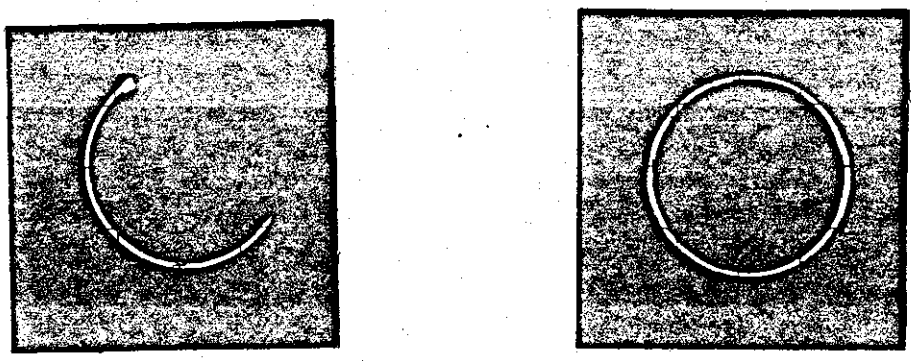

d_h = starting address of data file
d_ℓ = final address of data file
C = (d_ℓ - d_h)/4.D = C/2
Set 1st. and 2nd LSB to 0
C = d_h + C
e = d_ℓ - D
d = d_h
Set pointer to d, fetch y
increment pointer & fetch x
Set pointer to e, fetch x
x_d=x_e
No
Yes
e = e+8
Average y-coordinates
to determine mid-point
of chord
Yes
e>d_ℓ
No
e = d_ℓ - D
No
d>C
Yes
d = d+6
e = e-8

Fig. 7.4



(a) (b)

Fig. 7.5: Track of a Single Disc

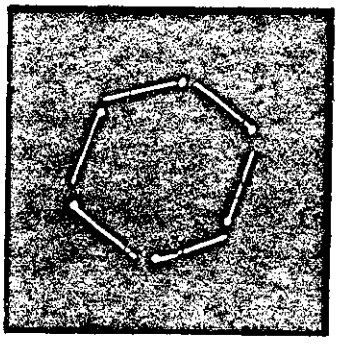


Fig. 7.6: Vector Field of the Circular Motion

displacements of the tracked object over a time interval of about 9.3 sec which is equivalent to the duration of 465 fields. The vector field was plotted using stored motion history data. The velocity of the object (displacement per fixed duration) is represented by a straight line and the bright spot at the end of the line indicates the direction of motion. The straight lines are drawn using a subroutine, labelled SUBLINE, which interpolates points representing a straight line defined by its end points. SUBLINE is described in detail in Section 9.9.2. Inspection of data representing displacements indicates that the object was moving at about 0.24 picture elements per frame (pefs).

7.4 Tracking of Two Discs

A flowchart of a program for tracking simultaneously two discs, labelled TRACTO, is shown in Fig. 7.7. The program inspects the incoming data to store it in two separate files, each for a different circle. This is accomplished by dividing the raster into two halves by a vertical line passing through its mid-point at $x = 266$ The program assumes that the two discs do not move in the same half at the same time.

After all edge points have been collected from the FIFO's, the microprocessor works on the data stored in the two files to determine the location of the centroids of the two moving targets, and displays the result on the storage oscilloscope.

Fig. 7.8, Plate (a) shows part of the targets' path. Plate (b) is a record of the complete circular path. This record shows spurious

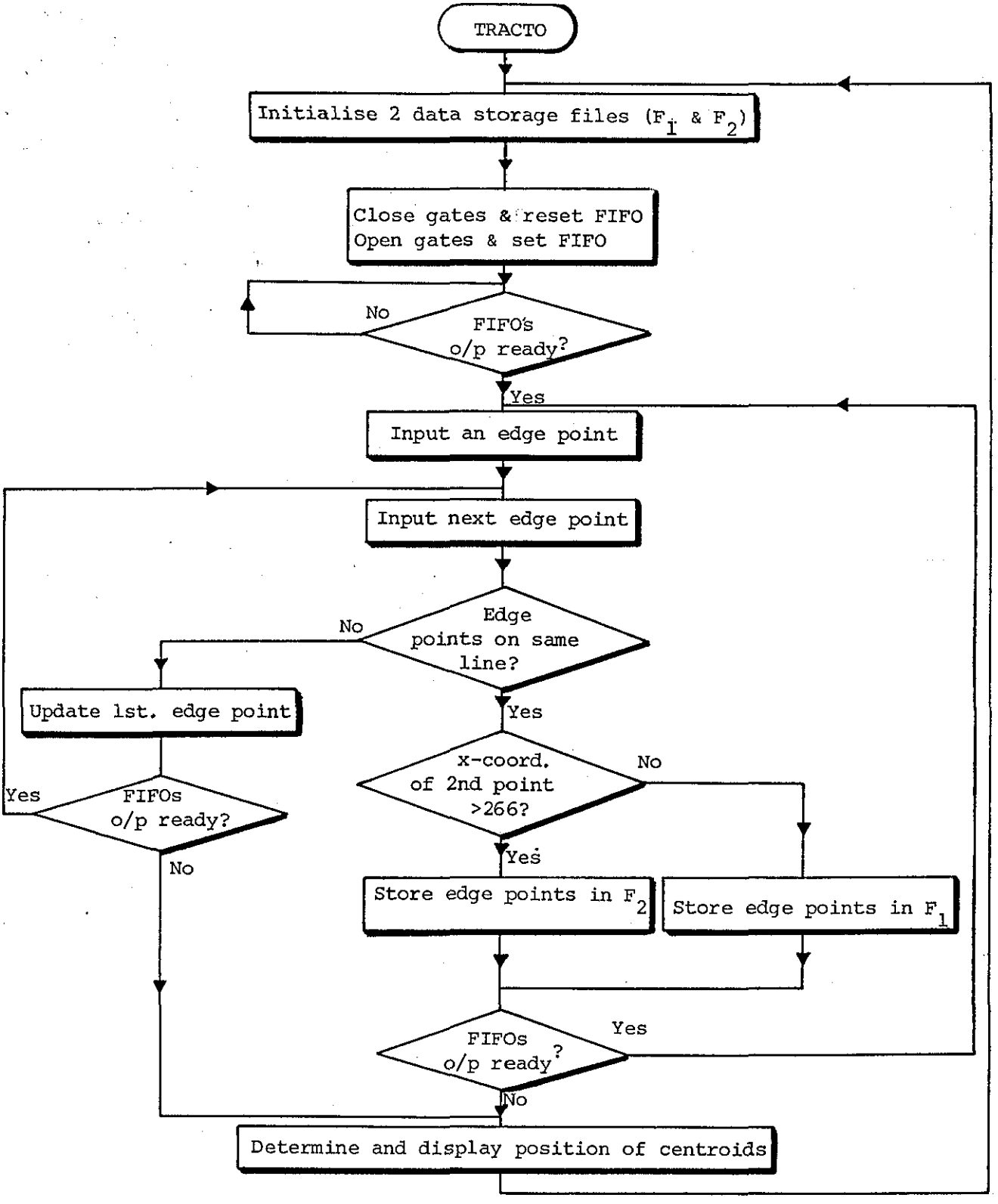
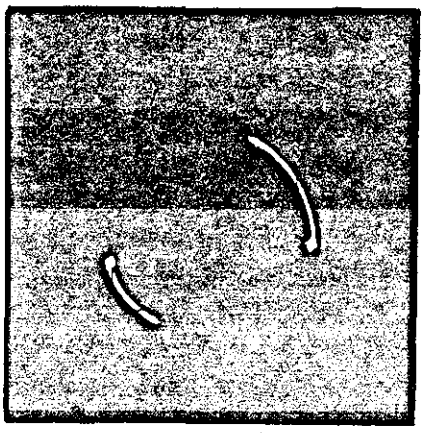


Fig. 7.7: Flowchart of TRACTO

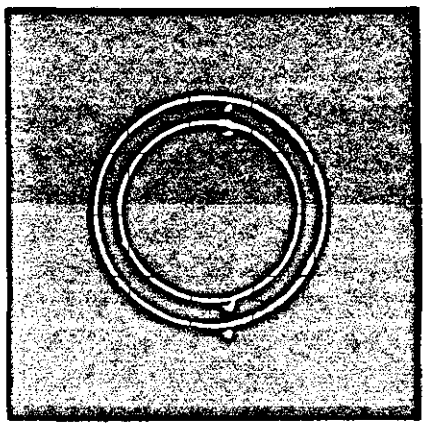
points at the top and the bottom which is a penalty for the assumption that only one target exists in either one of the two halves of the screen at the same time. This assumption gives satisfactory results for all positions of the targets except when they are spatially close to the line dividing the screen in the two halves. Dividing the screen into more than two regions through the inclusion of more straight lines does not improve the performance of the tracking program as ambiguities concerning the sorting of incoming data into two separate sets will still arise whenever one of the targets or both of them start to cross the dividing boundary between two adjacent regions.

Plates (c) and (d) in Fig. 7.8 show the path of the targets using data collected every 20 and 40 fields respectively. This is accomplished by branching to a subroutine which utilises CRUIN bit 6 and CRUOUT bit 8 to achieve delay for specific number of fields (Section 6.3); a procedure adopted when motion data pertaining to consecutive fields is too large to be accommodated in the available store.

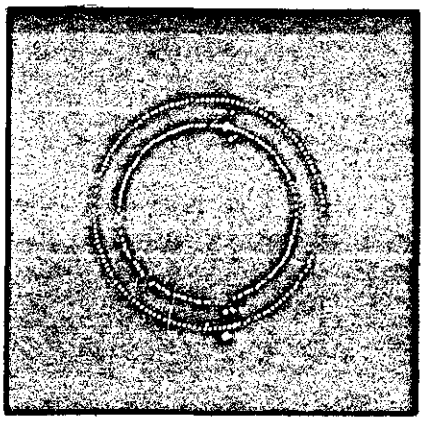
Fig. 7.9(a) shows two vectors representing the velocities of the two targets. These vectors are a measure of displacements over 280 fields. The upper vector represents a speed of 0.64 pefs and the lower one represents 0.46 pefs. The speed of the upper target is greater than that of the lower due to the fact that the first target is moving in a circular path which is further from the centre of motion than the second target. Plate (b) shows the two vectors superimposed on the track of Fig. 7.8(d).



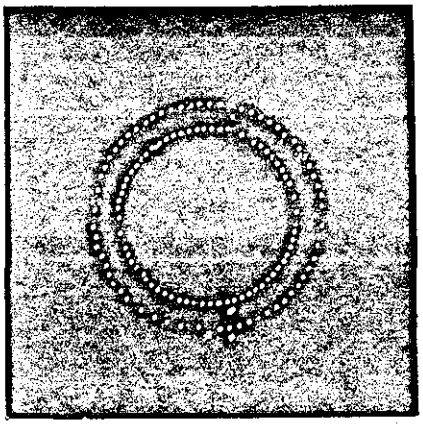
(a) Partial Path of Two Discs



(b) Complete Path (every field)

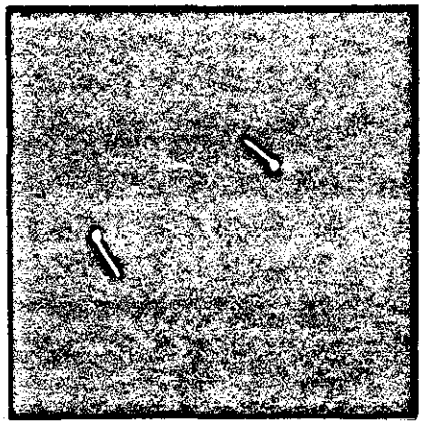


(c) Complete Path (every 20 fields)

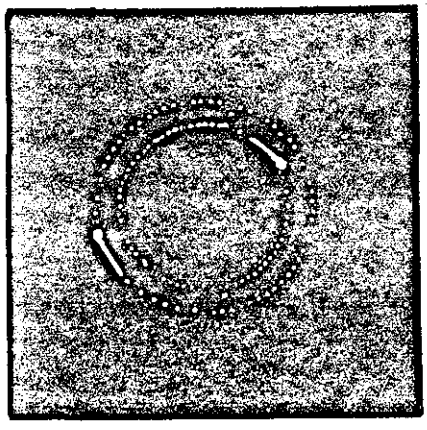


(d) Complete Path (every 40 fields)

Fig. 7.8: Track of Two Discs



(a)



(b)

Fig. 7.9: (a) Vectors representing motion of two discs at a particular instant

(b) The two vectors superimposed on the track of Fig. 7.8(d)

7.5 Tracking of a Moving Disc in a Scene Containing a Stationary One

Fig. 7.10 shows a flowchart of a program, labelled TRASIG, which identifies a moving disc in a scene including in addition a stationary disc. The moving disc is drawn on a screen rotated by the motor, while the stationary disc is drawn on a screen which forms the background. The edges of the targets are shown in Fig. 7.11(a).

To accomplish target acquisition a window is used to search the whole raster and then locks to the moving target. The program consists mainly of the three subroutines:-

a) WINGEN-M

This subroutine is responsible for generating a window. It is a modified version of WINGEN (Section 6.2)

b) STODA

This subroutine stores edge points of patterns contained within the window

c) CALCE

A subroutine for determining the location of the target

TRASIG moves a window of suitable size and overlays it in an orderly fashion across the whole raster, starting at the upper left hand corner. Each time the window is overlaid, TRASIG determines whether data has been stored or not. When data is available, the program inspects it to determine whether it is sufficient. If it is, the location of the centroid of the disc enclosed by the window is evaluated. Because the target motion is slow, the program loops for a particular period determined by a counter (n). At the end of this period the location of the centroid is examined to determine whether it is moving or not.

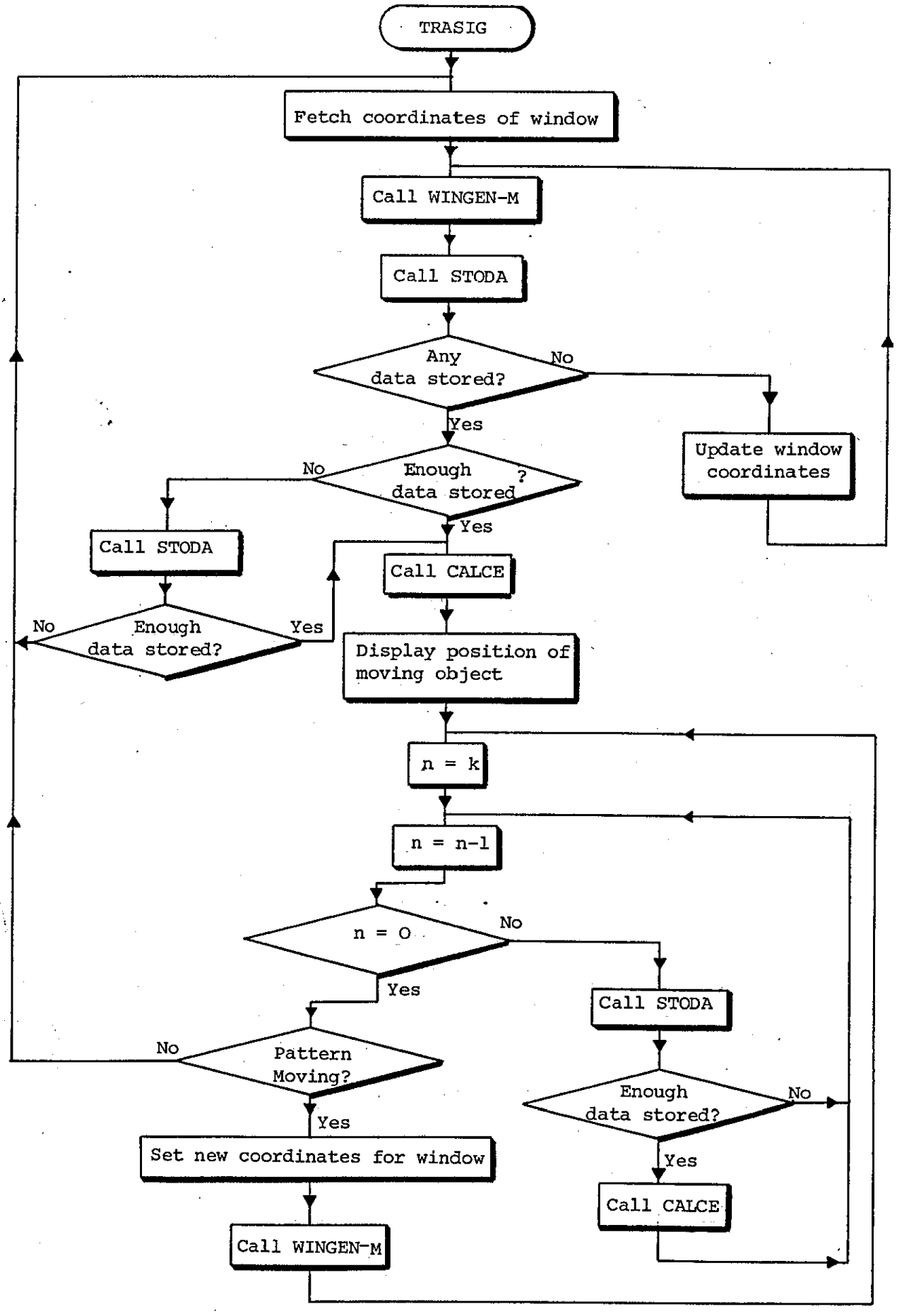
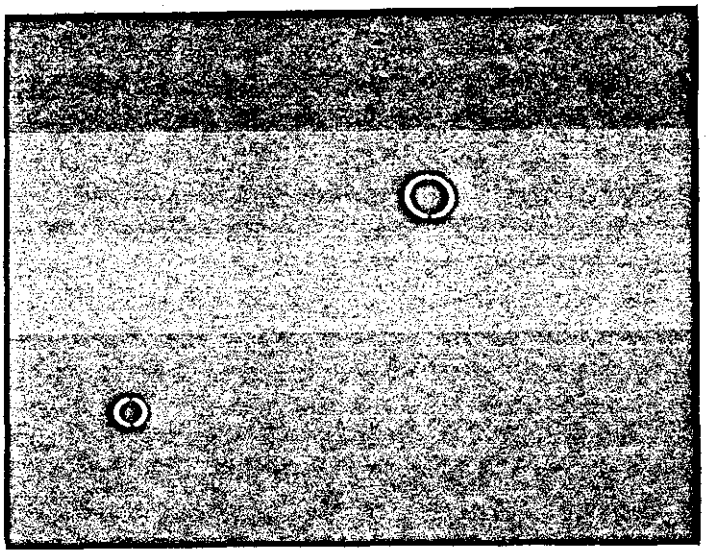


Fig. 7.10: Flowchart of TRASIG

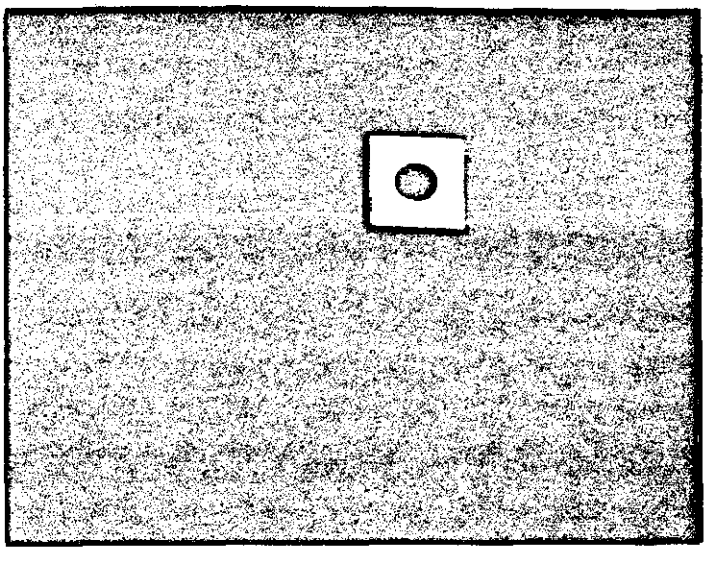
If it is moving the size of the window is diminished and is locked to the moving circle (Fig 7.11(b)). If it is not moving the program starts again searching for the moving target.

7.6 Remarks

The programs tested successfully kept track of the moving targets and provided information about their position in real time. The problems investigated deal with the tracking of a single target, two targets and identifying a moving target in the presence of a background object. The design of the tracking algorithms was subject to the two major constraints: the limited computing power of the microprocessor, and the real time environment. To endow a microprocessor-based system with a degree of understanding comparable to that of other processors, and within the context of real time application is an ambitious goal considering the present state of the art. However, with the emergence of more powerful microprocessors, such a goal might become a reality, but even when this is fulfilled, the problems associated with real time video tracking, such as identification of the desired targets from the background, and relevant target acquisition in a multiple target situation, remain to be resolved.



(a)



(b)

Fig. 7.11(a) Edges of the Two Targets
The Moving Target is at the Top

(b) A Window Overlaid on The Moving Target

CHAPTER 8

EYE MOVEMENT MEASUREMENT TECHNIQUES: AN OVERVIEW

8.1 Introduction

The human visual system is the most fascinating apparatus for the processing of spatial data. Recently, investigators representing different theoretical disciplines have directed their efforts towards elucidating the mystery engulfing it. One part which has been a subject of intensive study is the peripheral (the eye), where physiological and psychological studies have yielded much data on its structure, movements and response.

The eye exercises different kinds of movements so as to place on the area of maximum resolution (the fovea) that part of the total image about which the observer seeks more detailed representation[78]. The human observer does not assimilate all the information in an image immediately, but follows a sequential adaptive scanning procedure by selectively linking small areas and details of the image in an attempt to interpret the whole. Without understanding the role of eye movements it is impossible to decipher the mechanisms of vision and understand the thought process at work[78].

When an investigator is interested in the problem of perception of complex objects, records of eye movements are valuable. It will be easy to determine the order in which an object is examined, what elements are fixated by the subject, how often and for how long. Such records of eye movements illustrate the course of the process of perception.

Virtually any field that relies on or includes human processing of visual data can benefit from structures of eye movements. For example, training of a novice in the interpretation of photographs can greatly be improved if data is available about a search pattern adopted by an experienced observer. Knowledge of the properties of perception of images may be used in[79-83]: motion pictures and television, apparatus construction, arrangement of instruments on panels, evaluation of the possibilities of perception in complex conditions, target tracking, traffic studies, visual representation of data and advertising. Clinical and psychological applications include: measurement of pursuit and saccadic eye movements, and nystagmus ; measurement of vergence and muscular imbalance ; reading studies; testing effects of training, stress and fatigue, workload, etc. Also when understood, the role of eye movements and the principles governing them will help in diagnosis of disturbances of the central nervous system as well as visual disorders. Instruments which record and measure eye movements (oculometers) have many potential applications. For instance they can be used to aim a weapon or to control the pointing and the magnification of an image using eye control instead of the slow conventional manual control.

This chapter starts with a literature survey of elementary facts concerning the structure of the human eye and types of eye movements, followed by a review of eye movement measurement techniques.

8.2 The Structure of the Human Eye[78,84-86]

Functionally the human eye resembles a photographic camera (Fig. 8.1). It has a lens which can be focussed for different

distances; a diaphragm (the iris) which regulates the size of the light opening (the pupil) and a light-sensitive retina located at the rear of the eye, corresponding to the film of the camera. Next to the retina is a sheet of cells filled with black pigment which absorbs extra light and prevents internally reflected light from blurring the image. This sheet, called the choroid coat also contains blood vessels and is the nutritive layer of the eye. The outer layer is formed of a tough membrane, the sclera, consisting of firm connective tissue continuous in its anterior part with a transparent membrane, the cornea. The sclera enables the eye to maintain a constant shape and protects its contents. The same function is served by the cornea.

The dioptric apparatus of the eye, which takes part in the formation of an image on the retina consists of the cornea, the biconvex transparent lens, the transparent aqueous humour, and the transparent vitreous, filling the eye. The lens bends the light rays coming in, bringing them to a focus on the retina. It is aided by the curved surface of the cornea and the refractive properties of the fluids inside the eyeball.

The eye accommodates, or changes focus for near or far vision, by changing the curvature of the lens. This is made possible by the stretching and relaxing of the lens by the ciliary muscle fibres, which attach the lens to the ciliary body. Accommodation permits the image to be sharply focussed, resulting in an inverted real image of objects in front of the eye.

The only part of the human eye which is light-sensitive is the retina, a hemisphere made up of an abundance of photosensitive

cells (receptors), called according to their shape, rods and cones (Fig. 8.2). The cones are responsible for bright light vision, and are sensitive to colours, while the rods function in twilight or dim light and are insensitive to colours. In addition, the retina contains many sensory and connector neurons and their axons. There is no logical basis for the fact that the sensitive cells are at the back of the retina and to reach them, light must pass through several layers of neurons. It seems that as the eye develops as an outgrowth of the brain, it folds in such a way that the sensitive cells eventually lie on the furthermost side of the retina. At a point in the back of the eye the individual axons of the sensory nerves unite to form the optic nerve and pass through the eyeball. Here there are no rods or cones. This area is called the "blind spot", since images falling on it cannot be perceived.

The greatest concentration of cone cells is found in the fovea. a tiny depression in the centre of the retina. Only cones are present in this area; rods are absent. Also absent are overlying neurons and the cones are exposed to light directly. By virtue of its dense accumulation of cones the fovea permits the most acute vision. The concentration of cone cells decreases with increasing distance from the fovea, and at the retinal periphery cones do not occur at all. Rods on the other hand are particularly abundant there.

An external object is "pictured" on the retina as a series of points, each point corresponds to a rod or a cone. Impulses from these points are transmitted into the brain such that all fibres from the left sides of both eyes lead into the left half of the brain, and all fibres from the right sides of both eyes lead into the right half of the brain (Fig. 8.3). In each brain hemisphere the fibres from the eyes

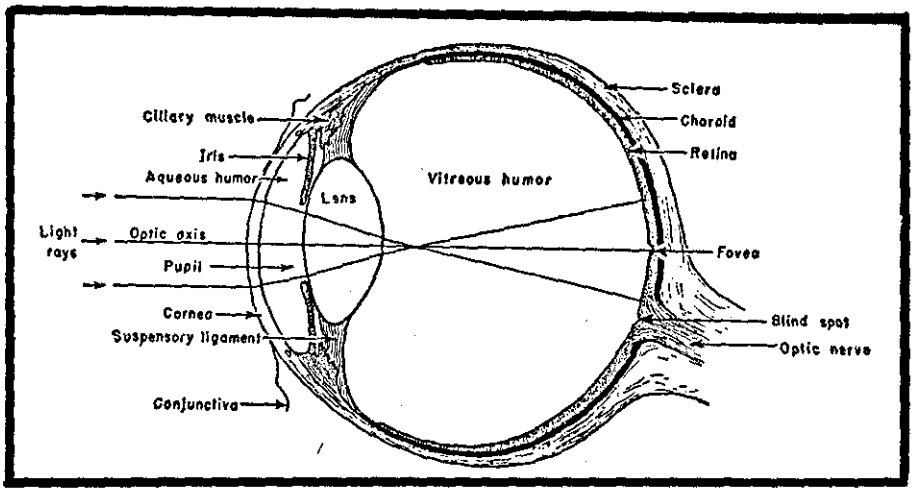


Fig. 8.1: Horizontal Section of the Human Eye (84)

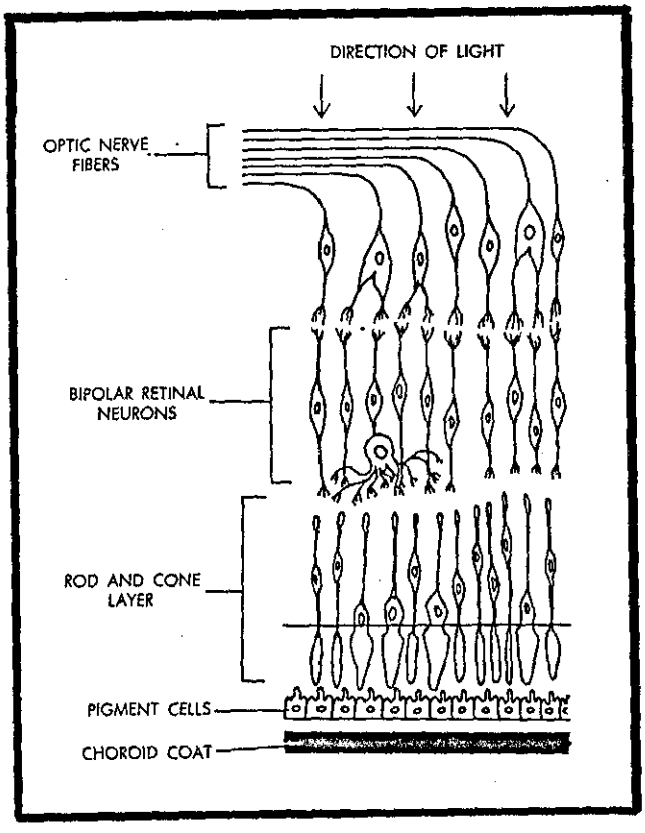


Fig. 8.2: The Structure of the Retina (85)

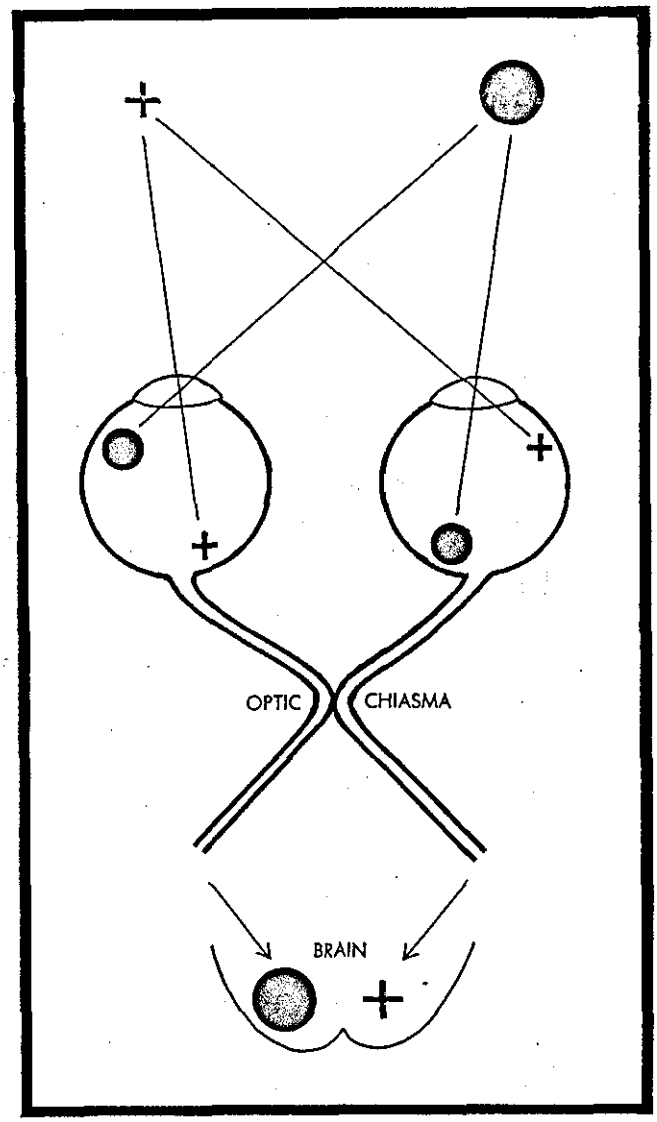


Fig. 8.3: Nerve fibre tracts from eye to brain. An object in the left field of vision registers on the right halves of both retinas, and impulses are transmitted into the right half of the brain[85]

lead to an optic lobe which contains the visual centres. It can be shown that for each group of rods and cones there exists a corresponding group of interpreter neurons in the visual centres[87-89].

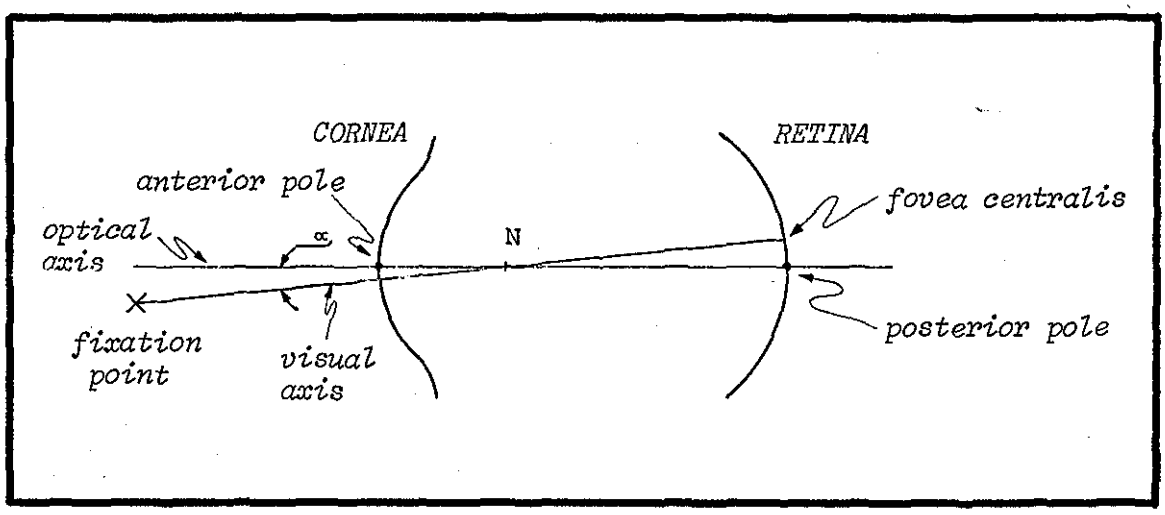
8.3 Angles and Axes of the Eye[90]

Part of the complexity of the ocular optical system results because an object and its image are generally not located on the optical axis and the limiting aperture of this system, the pupil, is displaced towards the nasal side. A number of angles and axes are a consequence of these unusual anatomical characteristics. Some of the axes and their angular relations may be defined as follows:-

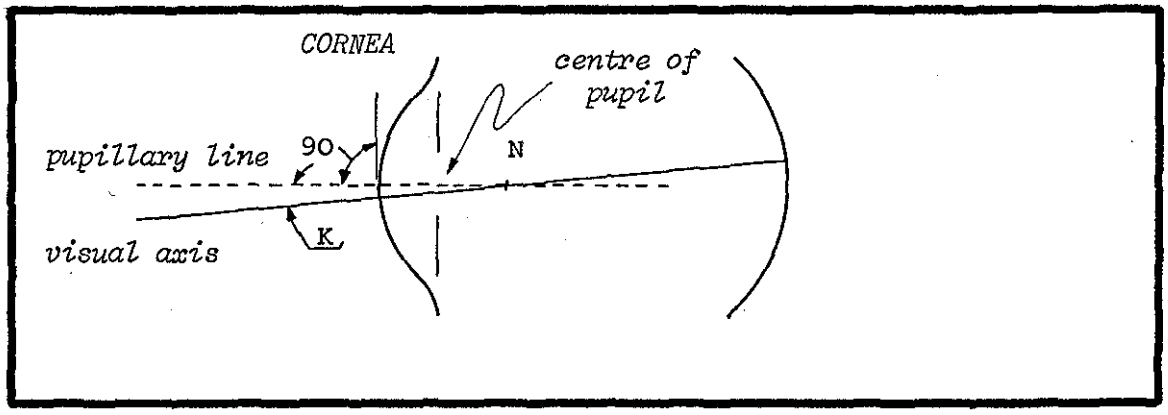
1. The visual axis (Fig. 8.4(a)) is represented by a line extending from an object in the visual field, through the cornea, a little to the nasal side of its centre of curvature, to the fovea centralis. When the sight is directed to a far view, the axes of the two eyes are parallel. But during accommodation for near vision, when the eyes converge, the axes meet at a point in the object at which one looks. This is called the fixation point.

2. The optical axis (Fig. 8.4(a)) passes through the anterior pole, the posterior pole, and the nodal point (N) of the eye. As in any optical system, it is the line along which the centres of the radii of curvature of all optical elements of the eye should be located. In the eye it cannot be defined with precision because uncertainty exists about exact location of the optical centres of each element.

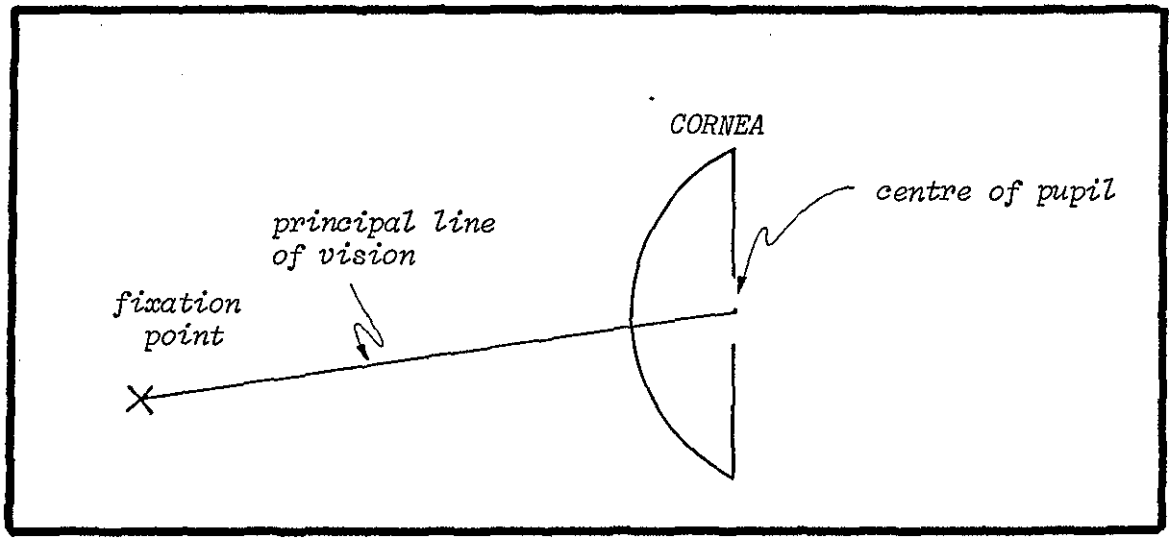
3. Angle α (Fig. 8.4(a)) is formed between the optical axis and the visual axis at the nodal point N of the eye. It usually measures about 5^{o}.



(a)



(b)



(c)

Fig. 8.4: The Angles and Axes of the Eye

4. The pupillary line (Fig. 8.4(b)) is a line drawn through the apparent centre of the pupil perpendicular to the surface of the cornea. The term "apparent centre" is used because it refers to the image of the pupil as seen through the cornea. This line is not symmetrically located with regard to the remainder of the optical system as the pupil is displaced slightly nasalward.

5. Angle K (Fig. 8.4(b)) is the angle formed by the visual axis and the pupillary axis at the nodal point.

6. The principal line of vision (Fig. 8.4(c)) is the axis from the apparent centre of the pupil to the fixation point.

8.4 Movements of the Eye[91]

All movements of the eye amount to its rotation about a certain centre lying inside the eye somewhere on the optical axis. This centre is not fixed with respect to the head, but the discrepancies are not large and are of no practical importance for most measurement applications.

Rotation of the eye is performed by small muscles, attached, on one hand, to the walls of the orbital cavity, and on the other, to the sclerotic coat at a distance between (0.85-1.27) cm, behind the circumference of the cornea (Fig. 8.5). There are six muscles for each eye, the four recti muscles (internal, external, superior and inferior) and the two obliques (superior and inferior). The internal rectus muscle turns the eye inward, i.e. towards the nose, the external rectus turns it outward. The superior rectus turns the eye upward and inward; the inferior rectus downward and inward. The superior oblique rotates the eyeball downward and outward, the inferior oblique

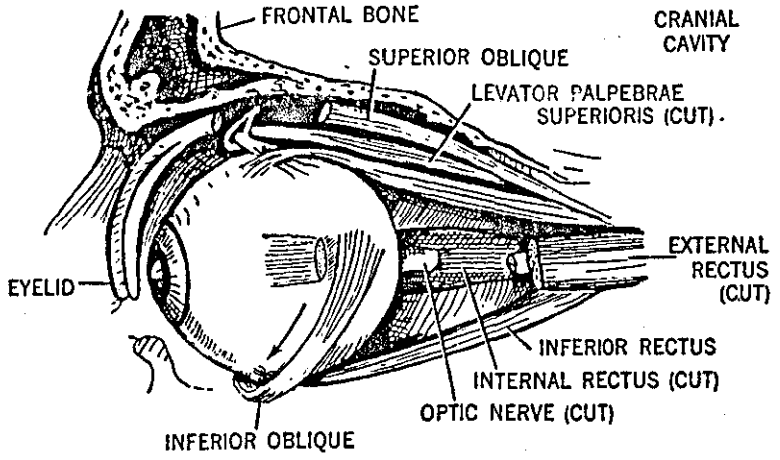


Fig. 8.5: Showing the eyeball in its bony socket (orbital cavity) and the ocular muscles.(91)

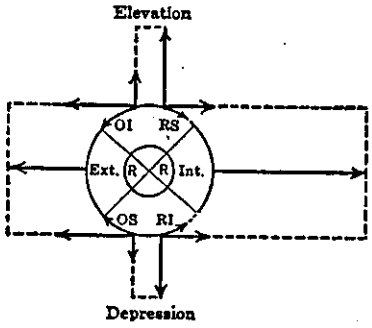


Fig. 8.6: Diagram of the actions of the muscles of one eye. (From Fuchs after Marquez.) OI, inferior oblique; RS, superior rectus; R.ext., external rectus; R.int., internal rectus; OS, superior oblique; RI, inferior rectus. The arrows indicate the direction in which each muscle tends to move the eyeball. It will be noted, however, that the actions of some muscles are antagonized by those of others. For example, the eye is turned outward by the combined actions of the external rectus, inferior oblique, and superior oblique. The upward and rotary movement caused by the inferior oblique is neutralized by the opposing action of the superior oblique.(91)

upward and outward. A diagram of the actions of these muscles is shown in Fig. 8.6. The eye muscles, especially the obliques, also cause a wheel-like rotatory movement.

The muscles of the two eyes act in unison. For example, the eyes are turned to one or other side - conjugate deviation - by the contraction of the external rectus of one eye and the internal rectus of the other. Convergence of the eyes - i.e. turning both eyes inward - is brought about by the contraction of both internal recti. In any movement of the eye, the muscles which bring about the opposite movement - i.e. the antagonistic muscles - are inhibited.

8.5 Types of Eye Movements[92-93]

When a subject is asked to fix his gaze at a particular point in a visual field, experiments have revealed that his eyes are not steady, but exhibit various types of movement. The major eye movements are summarised below.

1. Saccadic Movements

Saccadic eye movements are rapid conjugate movements by which a subject changes fixation from one point to another voluntarily. They include the "jump and rest" fixation movements observed in scanning a visual scene or in reading. The purpose of such movements appears to be fixation of the image of the target on the fovea (the high-acuity region of the retina), corresponding to 0.6-1.0 deg of visual angle. They are characterised by very high initial acceleration and final deceleration (up to 40,000 deg/sec^2) and a peak velocity during the motion which varies with the amplitude of the saccade and

may be as high as 400-600 deg/sec. The duration of a saccadic eye movement also varies with its magnitude and is generally about 30-120 msec. Saccadic eye movements have a range of 1-40 deg. Head motion is often involved when the target displacement exceeds 30 deg. In response to a visual stimulus, saccadic eye movements exhibit a latency of 100 to 300 msec.

2. Pursuit Movements

Pursuit, or slow-tracking, movements are conjugate eye movements used to track slowly moving visual targets with velocities in the range of 1 to 30 deg/sec. They are smoothly graded and appear to partially stabilise the image of the moving target or background on the retina, independent of the saccadic eye movement system. Smooth pursuit movements are not generally under voluntary control and usually require the existence of a moving visual field for their execution.

3. Compensatory Eye Movements

The compensatory eye movements are smooth movements which compensate for active or passive motion of the head or trunk. They tend to stabilise the retinal image of fixed objects during head motion and are attributable both to semicircular-canal stimulation sensing head motion and to neck proprioception associated with the turning of the head on the trunk.

4. Vergence Eye Movements

These are movements of the two eyes in opposite directions in order to fuse the image of near or far objects. They are considerably slower and smoother than conjugate eye movements,

reaching maximum velocities of the order of 10 deg/sec over a range of nearly 15 deg. They are stimulated by focussing error as well as binocular disparity.

5. Optokinetic Nystagmus

The optokinetic nystagmus is a characteristic sawtooth pattern of eye motion elicited by a moving visual field containing repeated patterns. It consists of a slow phase in which the eye fixates on a portion of the moving field and follows it with pursuit motion, and a fast phase or return saccadic jump in which the eye fixates on a new portion of the field. The minimum time between fast phases is approximately 0.2 secs resulting in a maximum frequency of approximately 5 Hz, with amplitude ranging from 1-10 deg.

6. Vestibular Nystagmus

The vestibular nystagmus is an oscillatory motion of the eye, similar in appearance to optokinetic nystagmus, containing a slow phase and a fast saccadic-like return. It is primarily attributable to stimulation of the semicircular canals during rotation of the head. A counter clockwise head rotation about a vertical axis leads to deflection of the cupulas of the horizontal semicircular canals, which induces image-stabilising slow-phase eye movement in a clockwise direction. As head motion continues the eyes jump back rapidly to pick up another position and repeat the sawtooth pattern.

7. Miniature Eye Movements

All classes of movement considered so far confer some obvious benefit on the visual system: they either serve to move the point of regard from one place to another, or alternatively they hold or help to hold the image of a visual object in place on the retina, despite any movement it may make relative to the head. But normal subjects make a number of other, less prominent types of eye movement called miniature eye movements, whose usefulness, if any, is not so obvious. These movements are generally less than 1 deg in amplitude and occur during attempted steady fixation on a target. They are classified into 3 categories, the drift, the microsaccades and the tremor. Drift is a slow random motion of the eye away from a fixation point at velocities of only a few minutes of arc per second. Microsaccades are small rapid eye movements which have been shown to be dynamically of the same nature as large voluntary saccades, of magnitudes as large as 1 deg. and occurring at intervals separated by as little as 30 msec. Microsaccades, like voluntary saccades, bring a visual target to the centre of the fovea. In addition, fixating on targets exhibits a high-frequency tremor in the range of 30 to 150 Hz with peak amplitudes of approximately 30 arc sec in the region of 70 Hz. Because of the presence of miniature eye movements, accuracy of 0.5 to 1 deg. is often sufficient in eye-monitoring tasks designed to show what part of the visual field is being fixated.

8.6 Characteristics Useful for Eye Movement Measurements[78,92]

There exists a variety of eye characteristics that have been utilised to measure movements of the eye. The following sections describe briefly these characteristics.

a) After-Images

The retina moves with the eye and hence makes possible subjective assessment of eye movement. One of the earliest quantitative techniques for determining the velocity of the eye during pursuit and saccadic eye movements was based on after images. Here a small light source is flashed periodically leaving a trace of after images on the retina. Density and spacing of these after images indicates fixation duration and the velocity of movements respectively. The main drawbacks of techniques based on after images is the subjective nature of the measurements and the fact that they can only be applied for a brief interval, after which a subject must report on number and placements of his after images.

b) Electrical Properties of the Eye

A potential difference of up to 1mV between cornea and retina normally exists and is utilised as the basis of electro-oculography, the most widely applied clinical eye movement measurement technique. Another useful electrical characteristic of the eye, is the impedance across it. The impedance measured between electrodes placed at the outer canthi of the two eyes varies with eye position.

c) The Cornea

The cornea has a smaller radius of curvature than the eye itself. Its bulge can be felt through the eyelid of the closed eye. Pressure transducers placed over the eyelid can detect changes in pressure and hence changes in eye position. Also the cornea can act as a mechanical post to centre tight fitting scleral contact lenses to which other measurement devices are attached.

The front surface of the cornea, although not a perfect optical surface, approximates a spherical section over its central 25 deg. Reflections of a bright object from this surface form a virtual image behind the surface which can be photographed and recorded. The position of the corneal reflection is a function of the eye position.

d) Purkinje Images

Light in passing through the eye generates four reflections. The sources of these reflections are the front and back surfaces of the cornea, and the front and rear surfaces of the lens. These four reflections are referred to as the Purkinje Images. The first Purkinje Image, reflected from the front of the cornea is the brightest. The next brightest one is the fourth, coming from the posterior surface of the lens. Measurements of the relative displacement between first and fourth images, representing points focussed from planes of different depths in the eye are the basis of a technique for measuring the orientation of the eye in space independent of its relation to head position.

e) The Limbus

The boundary between the iris and sclera, the limbus, is fairly sharp and hence can be used to gain an idea of position of the eye relative to the head. The ratio of the darker iris to the brighter sclera observed on both sides of the eye can be measured directly with photo-sensors or indirectly on an image of the eye.

f) The Pupil

The pupil is easily distinguished from the surrounding iris by its difference in reflectance. It can be made to appear much darker than the iris when the bulk of incident light does not come in directly along the axis of measurement and is not reflected out. On the other hand the pupil can be made to appear very bright when most of the light enters along the optical axis and is reflected back from the retina. The pupil normally varies between 2 and 8 mm in diameter in adult humans, and appears elliptical when viewed other than along the optical axis, with the minor axis shortening as the eye rotates.

g) Other Optical and Nonoptical Landmarks

In addition to the iris and pupil, other optical landmarks can be traced. Retinal blood vessels, scleral blood vessels or folds of the iris can be identified and tracked by optical techniques. Some artificial landmarks (a globule of mercury, chalk, egg membrane...etc) have been placed on the eye and detected optically. A small piece of a metal embedded in the sclera is the basis for magnetic tracking of the eye position.

8.7 Major Eye Movement Measurement Techniques[92]

The great variety of techniques employed for eye movement measurements illustrate the inadequacy of any one method for all applications. A number of new methods have been brought forth in recent years permitting improvements in convenience and accuracy of assessing fixation point. All these techniques employ one or more of the properties covered in the previous section .

8.7.1 Measurement of corneoretinal potential

Corneoretinal potential is measured by placing electrodes around the eye and reading the potential differences between them. In electro-oculography, D.C. recording methods are used to determine eye position, whereas in electronystagmography A.C. recording is implemented for measurement of eye movements including the fast and slow phases of nystagmus.

Electro-oculography has the largest range of any of the objective methods practical for human studies, since it does not require visualisation of the eye. The method is usable for eye movements up to $\pm$70 deg. Linearity deteriorates at excursions greater than 30 deg., especially in the vertical direction. The chief sources of error are muscle artefacts, eyelid interferences, basic nonlinearity in the technique and variations in the corneoretinal potential attributable to light adaptation,diurnal variations and the state of alertness. The presence of external electrical interference is troublesome and as a precaution the system must be electrically isolated from its surroundings.

8.7.2 Corneal reflection

Because the radius of curvature of the cornea is less than that of the eye, the corneal reflex moves in the direction of eye movement, relative to the head. If a light source is fixed with respect to a subject's head, then to relate eye position to the material being fixated requires either a fixed head system, a method for recording head position, or a technique for recording the field of view relative to the head at every sample. Recording of the field of view relative to the head can be obtained by using a head-mounted eye monitor camera (Fig. 8.7). Systems based on corneal reflection have an accuracy of 0.5-1 deg.

The uncorrected linear range of all corneal reflex systems which employ a single light source for the reflex is limited to eye excursions of $\pm$15 deg. vertical or horizontal. Larger excursions place the reflex in the non-spherical and rougher peripheral portion of the cornea and requires a complex calibration and linearisation technique. The reflex range is ultimately limited by the size of the cornea and its partial disappearance behind the eyelids. Factors which limit the accuracy of corneal reflection are variations in cornea shape, thickness of tear fluid and corneal astigmatism.

8.7.3 Limbus, pupil and eyelid tracking

The limbus is an easily identifiable edge which can be detected optically. When only horizontal eye movements are of concern, then the left and right extremes of the iris can be tracked; either by measuring the gross difference in reflected illumination from fixed areas of the eye on one side of the central gaze position; or by tracing the limbus with a video scan system. When vertical

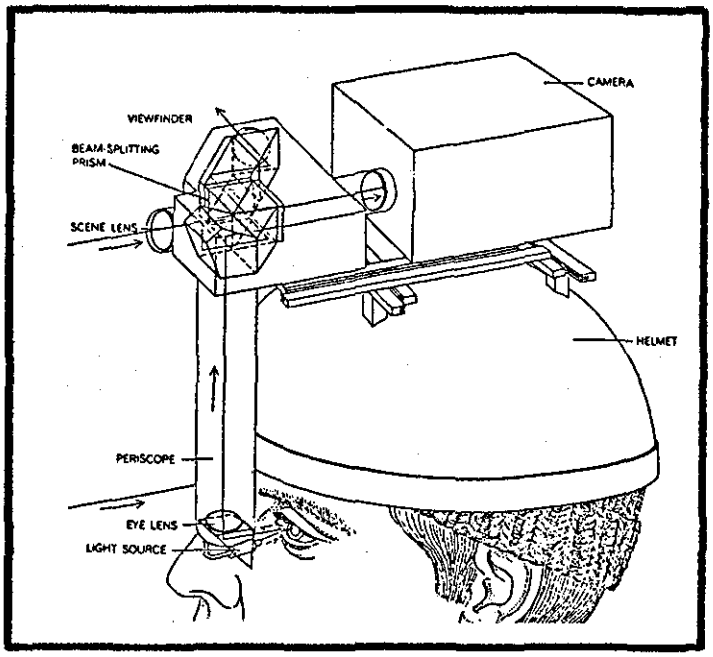


Fig. 8.7: A Head-Mounted Eye-Monitor Camera (92)

measurement is also required one may track either the eyelid level or the vertical motion of the visible part of the limbus. Nearly all limbus tracking systems use invisible infra-red illumination. They all measure the position of the limbus relative to photodetectors. For head-fixed photodetectors and illuminators, free head movement is possible and the measurement is of edge relative to the head.

The pupil offers a number of distinct advantages over the limbus; being smaller and hence unobscured by the eyelid for a much greater range of eye motion. It presents a greater portion of the round or slightly elliptical shape, the centre of which virtually coincides with the foveal optical axis of the eye. The edge of the pupil is usually crisper and sharper than the limbus, offering a higher

resolution. On the other hand, the pupil, when viewed under normal illumination, appears black and consequently presents a lower contrast with its surrounding iris, than iris with respect to sclera. Improvement in the contrast is achieved by using collimated illumination, where light is reflected from the interior of the eye along the illumination axis, causing the pupil to appear bright.

Some techniques used for the tracking of iris or pupil involve scanning the edge with a normal TV camera, with sufficient sensitivity in the near infra-red region (700-900 millimicrons) and provided with infra-red lighting. Unless infra-red or other non-visible light is used, the illumination can be distracting to the subject. On the other hand the use of infra-red illumination demands special sensors and optical design. Image dissectors are also used to trace the circumference of the pupil or the visible portion of the limbus. Scanning methods require either head fixation or attachment of a scanning mechanism to the head, directly or using fibre optics.

Many implementations of limbus and pupil tracking techniques yield good results for a reasonable range of eye movements. The main disadvantage is the requirement of rigid head fixing or alternatively a head-mounted device, as a 1mm shift in the position of the pupillary disc or limbus may correspond to considerable rotation of the eyeball. Although the pupil has a number of advantages over the limbus, its diameter varies as a result of both psychological and physiological influences.

8.7.4 Contact lens method

The most precise measurements of the eye movements are those employing a contact lens or a mirror tightly attached to the eye. Conventional corneal lenses are too mobile, and for this reason all measurement systems employ special lenses consisting of two individually ground spherical surfaces fitting snugly over the cornea and sclera. For accurate recording of eye movements it is essential for the lens to move with the eye. Tight fit is achieved by close grinding tolerances and by the suction effect of 20mm Hg resulting in a negative pressure between the contact lens and the surface of the eye. Yarbus achieved the stabilisation by withdrawing a small amount of fluid through a valve after applying the lens. Withdrawing air from under the contact lens is also effective but of limited time use because of lack of corneal irrigation. All contact lens systems cause discomfort and the very tight ones usually require the application of a topical anaesthetic.

The most commonly used contact lens system is the "optical lever", in which one or more plane mirror surfaces, ground on the lens, reflect light from a light source to a photographic plate, a photocell, or a quadrant detector array. The latter is a solid state photodetector arrangement which produces a pair of voltages proportional to the x and y coordinates of a spot of light falling on it. There are three advantages of plane mirror reflection over the corneal reflection and these are:-

1. Angle of reflection depends only on the eye rotation and is independent of pure linear displacements

2. Imperfections related to the cornea are eliminated

3. The change in reflection angle is twice eye rotation

Because of the high inherent accuracy of contact lens systems, very careful head stabilisation relative to the recording device is usually used. The good resolution offered by such systems (5-10 arc sec) is at the expense of range and consequently they are normally applicable for the study of miniature eye movements. The expense and discomfort of a contact lens makes it suitable for use on a few subjects rather than for wide spread investigation on the normal population.

The dangers of fitting a contact lens with negative pressure are considerable as there is the possibility of deforming the cornea and the worst hazard of damaging the accommodation muscles as a result of the pressure stress exerted on them.

8.7.5 Point of regard measurement techniques

It is often of interest to know a fixation point of a subject as it falls in space, i.e. point of regard, rather than the position of the eye with respect to the head, or in other words to determine where a subject is looking regardless of whether his eye got there through rotation or head motion. In the approaches discussed so far, lateral head motion generates the same effects as rotation of the eye and the point of regard is derived from eye position by carefully stabilising the head. Unless the head is fixed, lateral motion introduces large errors-approximately 1 deg. of error for every 12.7×10^{-5}m of lateral motion[92]. It is difficult to eliminate head motion completely by any fixing method, for even when the head is rigidly held steady with a bite board and head constraints, the eyeball

can still translate in its socket, as it is sitting in a bag of fat with 6 muscles pulling on the sides of it[94]. Alternatively the head position can be monitored and the head position signals are combined with eye motion signals to obtain a measure of the point of regard.

A desirable procedure is one which would allow relatively free, natural head motion and measure the position of some parameters of the eye in such a manner that they would indicate eye rotation only. This can be achieved by monitoring positions of various details on the eye which behave differently under translation than under rotation. Two systems which adopt such an approach are the double Purkinje image tracking system and the corneal reflection - pupil centre measurement system.

8.7.5.1 The double Purkinje image method[92,94]

The second Purkinje image is relatively dim, while the third is formed in a plane far from the others. For these reasons they are not utilised for eye movement measurements. When the eye rotates, both the first and fourth Purkinje images move, but through different distances;whereas if the eye translates, they move through the same distance. Therefore change in separation between these two images is directly related to the angular rotation of the eye and is independent of head translation.

In order to put together a practical system for recording eye position using the first and fourth Purkinje images, Cornsweet and Crane[92] assembled a complex collection of optical and electronic equipment. A simplified schematic diagram of this equipment is shown in Fig. 8.8., and briefly it works as follows. The eye of a subject

is illuminated with flickering infra-red light. This light is reflected as if it is emanated from two small spots, the first and fourth Purkinje images, and these images are dealt with independently. Collection optics view the eye and image the two Purkinje reflections. Light from both images is reflected from a servo-controlled mirror, and the first image is directed towards a quadrant photodetector. Whenever the image moves off the centre of the photodetector, the output of its associated circuitry causes the mirror to rotate in such a way as to return the image to the centre of the quadrant photodetector. The signal used to drive the mirror is proportional to both translation and rotation of the eye.

Light from the fourth Purkinje image is reflected from the same mirror before it is projected onto another quadrant photodetector, whose position is servo-controlled so that the detector follows the image wherever it goes. It is the position of this second servo-driven quadrant photodetector that is the true output of the system. It is proportional to the component of the movement of the fourth Purkinje image that is due exclusively to rotation of the eye. The servo mechanisms are relatively fast and respond in about 5 msec to a saccade of about 5 deg. The overall accuracy of the system is limited largely by the photodetector noise level and is of the order of 1 min. of arc.

8.7.5.2 The corneal reflection - pupil centre measurement technique[83,92]

Two features which can be used to avoid translation artefacts are the corneal reflection and the centre of the pupil. Each has been independently used for eye-position measurements, but

differentially they provide a powerful tool. First, the image of the point on which a subject is fixating is coincident with the centre of the pupil. Second, a subject's angle of gaze with respect to a light source is approximately proportional to the distance between the image of the light source and the centre of the pupil. These two properties are equivalent to each other. Techniques based on the first use multiple light sources, and the point of regard is determined by which light source is imaged on at the centre of the pupil. For the second case, a single light source is used, and the error between

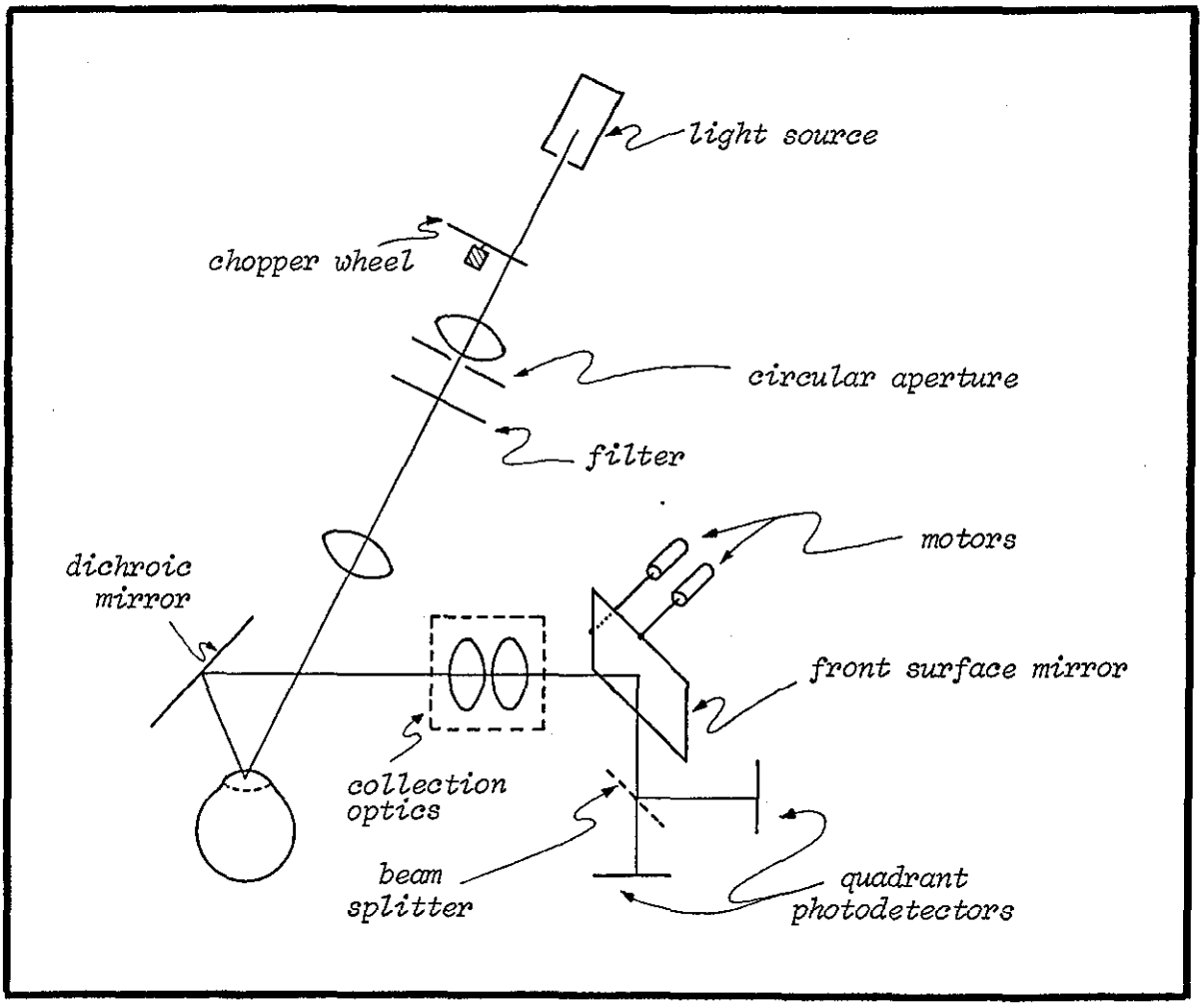


Fig. 8.8: Simplified Schematic Layout of the Double Purkinje Image Eye Tracker

it's image and the centre of the pupil is measured.

Techniques based on multiple light sources are not amenable to high resolution analysis as the entire field is imaged over a small area of the cornea. The principle of using a single light source was put into practice by the oculometer developed by Merchant.[83] In this oculometer the pupil is viewed by observing optics that are coaxial with the illumination optics (Fig. 8.9). The corneal reflection always appears in line with the centre of corneal curvature as shown in Fig. 8.10. The apparent displacement of corneal reflection from the centre of the pupil is thus equivalent to the apparent displacement of the centre of corneal curvature from the centre of the pupil. This displacement (D) is given by;[82]

$$D = K\sin\theta$$

where θ is the angle between the eye's and the oculometer's optical axes and K is the distance from the centre of corneal curvature to the observed pupil centre. The observed pupil is the entrance pupil which is the virtual corneal image of the true pupil and lies anterior to it[95] (Fig. 8.11).

For small angle of rotation the relationship simplifies to $K\theta$. Thus eye rotation is a linear monotonic function of the distance between the projections of the corneal reflection and the pupil centre. If $\overline{A}$ and $\overline{B}$ represent displacement between corneal reflection and pupil centre in the horizontal and vertical directions respectively, then the resultant displacement ($\overline{P}$) is given by;-

$$\overline{P} = \overline{A} + \overline{B}$$

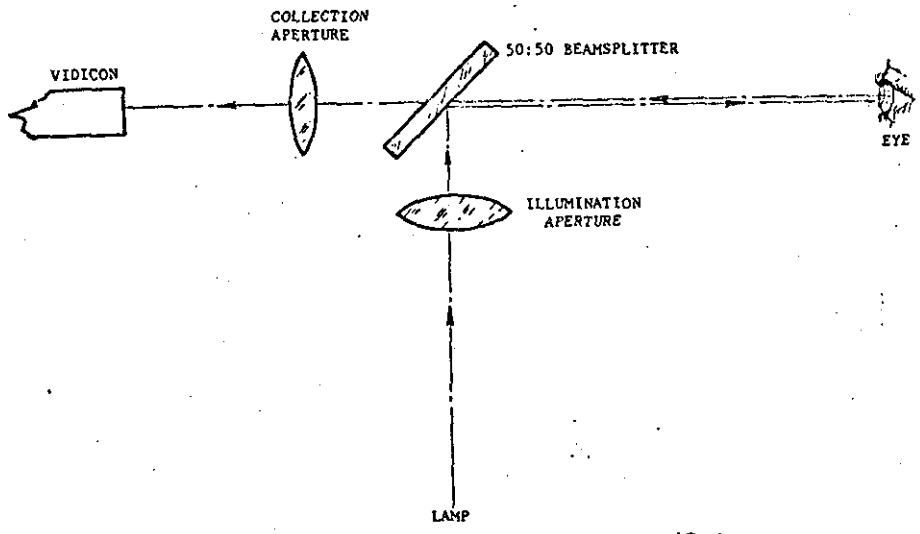


Fig.8.9: 'Bright-pupil" optical system.(83)

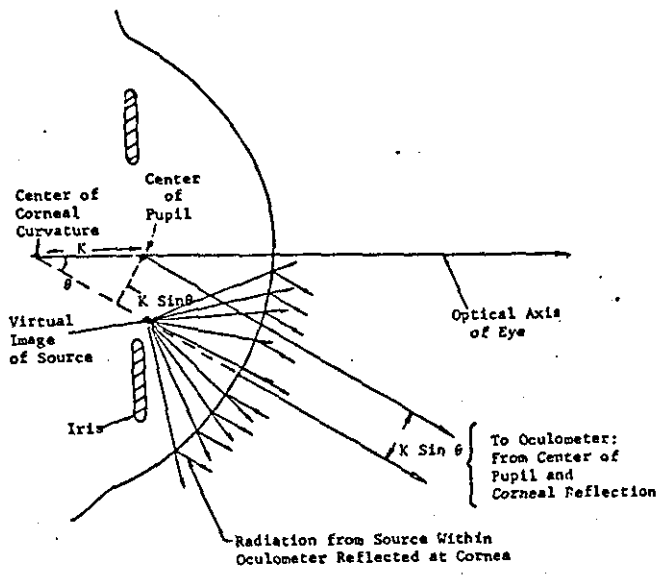


Fig.8.10: Basic sensing principle. Displacement of the corneal reflection from the center of the pupil by $K \sin \theta$ is proportional to the angular direction θ of the eye, and is independent of the position of the eye.(83)

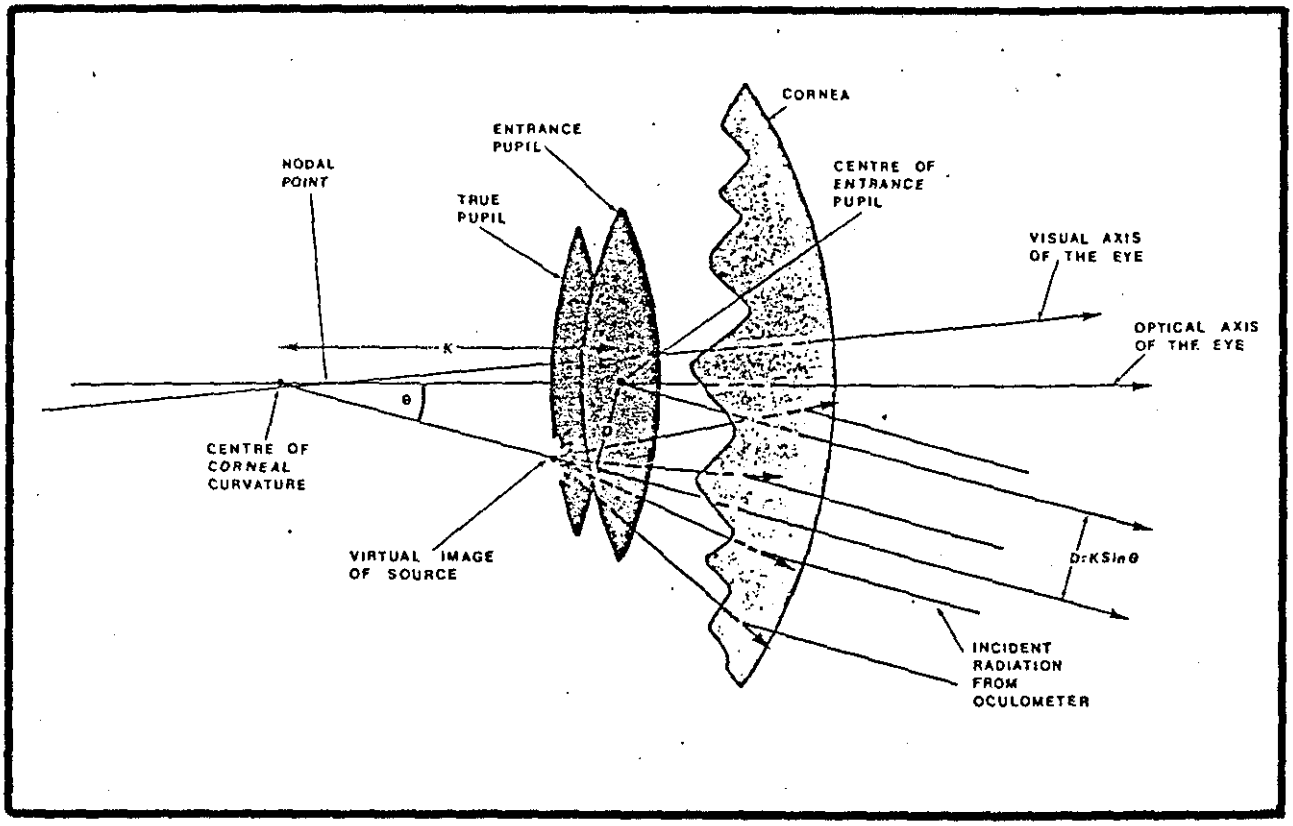


Fig. 8.11: Position of Entrance Pupil Relative to True Pupil (95)

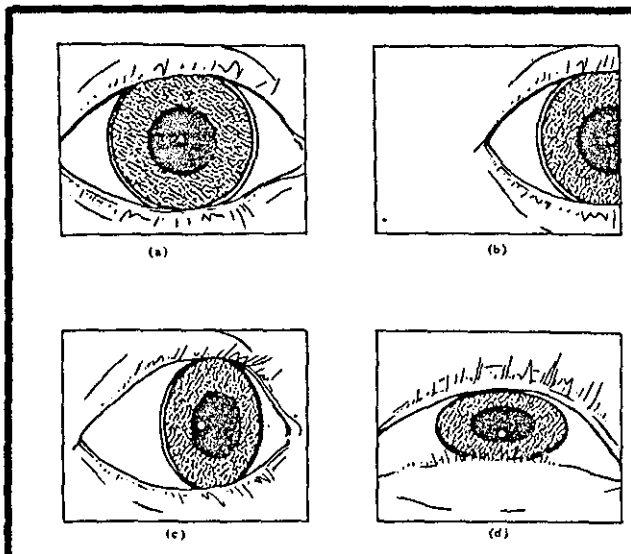


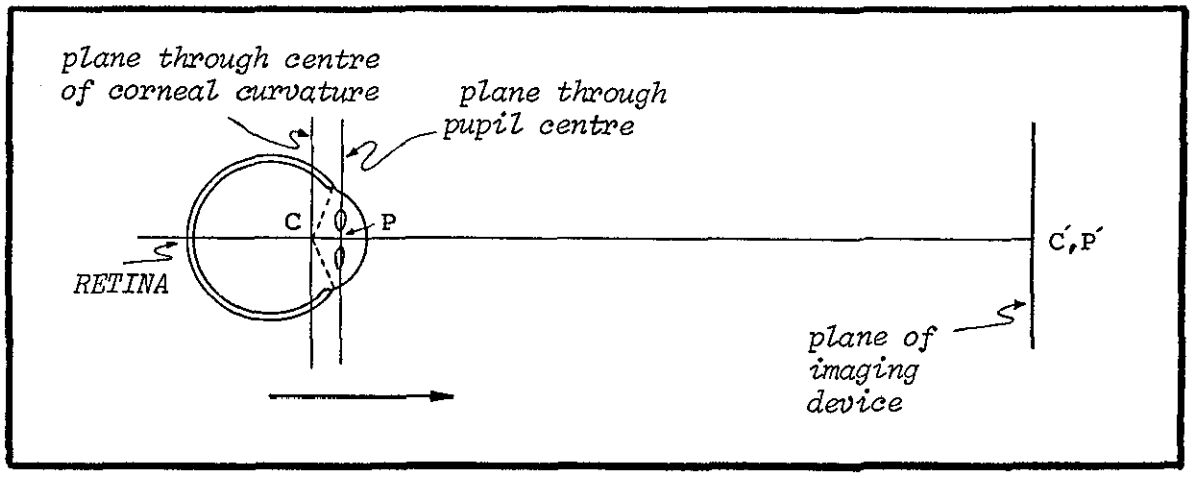
Fig.8.12: Effects of eye translation and rotation on corneal reflection–pupil center (Merchant & Morrissette, 1974): (a) Eye looking straight ahead–note corneal reflection is at center of pupil. (b) Eye looking straight ahead but laterally displaced–note corneal reflection still at center of pupil. (c) Eye looking to side–corneal reflection displaced horizontally from pupil center. (d) Eye looking up–corneal reflection displaced vertically from pupil center. (83)

Fig 8.12 demonstrates how the eye image on a sensor plane appears as the eye undergoes rotations and translations. The corneal reflection with respect to the centre of pupil remains unchanged as a result of lateral head motion and changes for eye rotation only. It has been shown that the corneal reflection moves relative to the pupil by approximately 7.62×10^{-5}m (3×10^{-3}in) per degree of eye rotation.

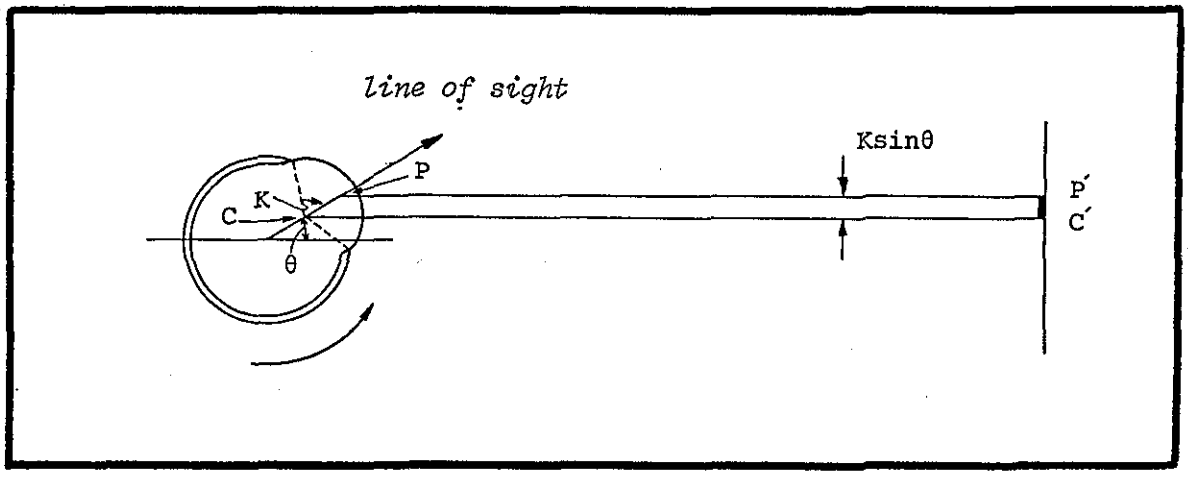
A simplified diagram illustrating projections of the pupil centre and corneal reflection on a plane of an imaging device is shown in Fig. 8.13. When an eye is looking straight ahead the pupil centre and the corneal reflection, i.e. centre of corneal curvature, lie in parallel planes which are in turn parallel to the plane of the imaging device (Fig. 8.13(a)). Here the projections (p', c') of the centres of pupil and corneal curvature (p, c) on the imaging device coincide. When the eye is rotated by angle θ (Fig. 8.13(b)) the projections are separated by $K\sin\theta$. If the eye, maintaining the same angle of rotation θ, is laterally displaced, the distance between the two projections is unaffected (Fig. 8.13(c)).

To implement the pupil centre-corneal highlight technique all that is needed is to image the eye and determine the centre of the pupillary disc and the centre of the corneal reflex in an x-y plane. Then the relative vector between these two centres is a measure of the point of regard. In earlier work an image dissector tube was used as the image sensor. In present systems, the dissector has been replaced by a vidicon tube operating with a standard TV raster scan. One advantage of the vidicon is the lower cost for both tube and the associated camera electronics.

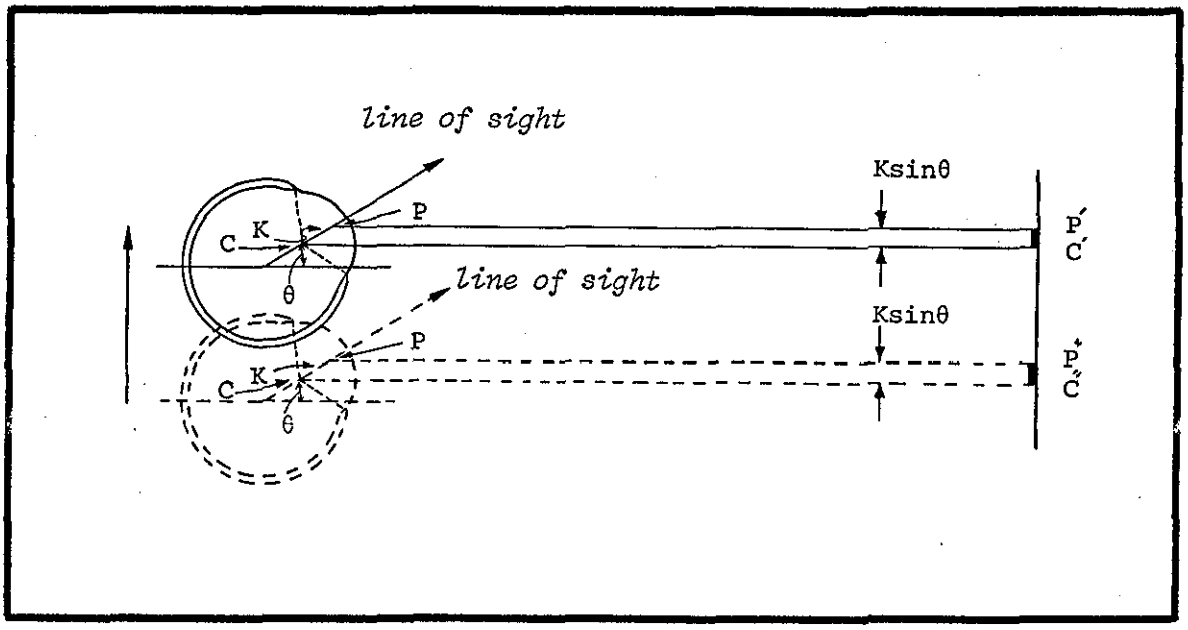
Small head movements do not interfere with eye movement measurements based on the corneal reflection-pupil centre technique. The



a) Projections of an Eye Looking Straight Ahead



b) Projections of an Eye Rotated by θ Degrees in the Direction Shown



c) Projections of an Eye Laterally Displaced While Maintaining the Same Angle of Rotation

Fig. 8.13: Projections of the Pupil Centre and the Corneal Reflection on an Imaging Plane

constraints which limit the extent of head movements are;-

1. Illumination must be coaxial with the pupil
2. The pupil image must be kept on the planar surface of the imaging device
3. The resolution of the electrooptical sensor must be better than 7.62×10^{-5}m so that eye rotations of a fraction of a degree may be resolved. This can be achieved with a vidicon tube when the field-of-view of the image sensor is limited to 6.45×10^{-4} m^2(1 in^2)at the eye by means of a telephoto lens combination. This limitation together with the finite depth of focus of the oculometer optical system confines the effective instantaneous sensor eye space allowable for head movements to within 2.54×10^{-2}m cube (1 in.cube)

A number of versions of oculometers implementing the corneal reflection-pupil centre measurement technique exists. These include the Honeywell oculometer, the EG & G/Human Engineering Laboratory oculometer , a laboratory oculometer developed by the University of Alberta and the Whittaker corporation eye view monitor.(83) Some of these oculometers employ a non-visible light source to locate the pupil centre and the corneal reflex. Honeywell and Whittaker oculometers use an invisible infra-red light source and a TV camera with sufficient sensitivity in the infra-red region. EG & G, in their original version use visible light. Here all light sources are polarised except for a small portion which is reflected from the eye. The returned light is passed through a polarising filter to a TV camera. In the Honeywell oculometer, the pupil is backlighted by the reflected

light coming from the retina. The resulting image is a bright pupil and even brighter corneal reflex. In some Whittaker oculometers and the EG & G systems the pupil is black. Some of the oculometers are designed to follow head motion within one cubic foot of space. This is achieved through the addition of a two-axes moving-mirror system and a servo-controlled focussing lens, which under the control of a signal processor, automatically scans 0.305m. cube (1 ft. cube).

8.7.5.3 Remarks

Maintenance of a fixed head or attachments to the head are difficult, uncomfortable, unsuitable for many subjects and may require a long set up time. Techniques for measurement of the point of regard offer considerable advantage since they tolerate head movements within certain limits. Usually a subject is seated in a chair with a head rest and there is no need to clamp him. The illumination which is invisible, is not annoying or distracting to him. Moreover, the output is a vector indicating the point of regard which is very amenable for processing. But generally advanced and sophisticated instrumentation for point of regard measurements are bulkier and more expensive than other simpler and more direct means for eye movement measurements.

The double Purkinje image systems have a small field of view ($\pm$15 deg.) compared to some other techniques and are pupil diameter dependent. They also require higher illumination in order to bring the fourth Purkinje image above noise, and the optics have to be fairly close to the eye. Normally for high-precision measurement, the head is stabilised with a bite board.

The corneal reflection-pupil centre measurement systems, when based on ordinary 50 fields/sec vidicon cameras are slow as they can operate only as fast as 50 determinations of eye position per second. Their accuracy ranges between 1-2 deg., which is not as good as that obtainable with double Purkinje image systems. For a field of view outside $\pm$15 deg. the cornea flattens out and the measurements become non-linear, although still monotonic. There are imperfections and asymmetries in some corneas that make the measurements non-linear within the $\pm$15 deg range. In addition, as the pupil constricts and dilates it does not normally maintain a fixed centre with respect to the eyeball. Another source of error is the tear film. All these imperfections and errors can be eliminated to a large extent by processing using a digital computer.

Although corneal reflection-pupil centre measurement techniques are neither the most accurate nor the fastest compared to other methods, yet they provide an adequate means for automatic detection of eye fixations and saccades.

8.8 Real Time Eye Motion Tracking and Analysis Systems

Most of the eye motion tracking and analysis systems involve the use of a digital computer for sophisticated processing of data, and for corrections and linearisation essential for eye movement measurement techniques. In the majority of systems a video tape recording of the picture of the eye is obtained for later processing by a computer, which is time consuming. In few systems the output of an oculometer is processed in real time by a standard minicomputer provided with a special interface to extract the eye-direction information.

Examples of real time systems include the EG & G/Human Engineering Laboratory System[96] and the PERSEUS, a system developed in NASA Ames Research Centre[97].

8.8.1 The EG & G/Human Engineering Laboratory System

The EG & G/Human Engineering Laboratory System occupies three compartments (Fig. 8.14). A subject sits in a chair with a head rest, and views a projection screen without knowing that his eye is being observed with a camera located underneath the screen. The system utilises the corneal reflection-pupil centre measurement technique. As the subject is being tested, eye movement data is collected on a video tape and is simultaneously sent to a computer. The computer is a PDP-11/20 with a disc and two tape units. It is programmed to determine the fixation time, number of fixations and variability of fixation duration. The system has a graphic display terminal to project eye movement behaviour in real time and is also used to collect calibration data at the onset of each experiment.

The system is provided with an automatic tracking facility. Initially an operator manipulates a joy stick to superimpose a circular cursor on the image of the pupil. Then an automatic tracker, employing a fairly fast servo loop, assumes control and tracks the pupil. The tracking system continuously monitors the video level at points defined by the cursor and if needed, repositions the cursor to overlay the area of maximum contrast gradient at the pupil/iris boundary. There is a second slower servo loop which compares the cursor position with the centre of the TV picture. Whenever the cursor moves from this position, appropriate signals are generated and fed to motors to

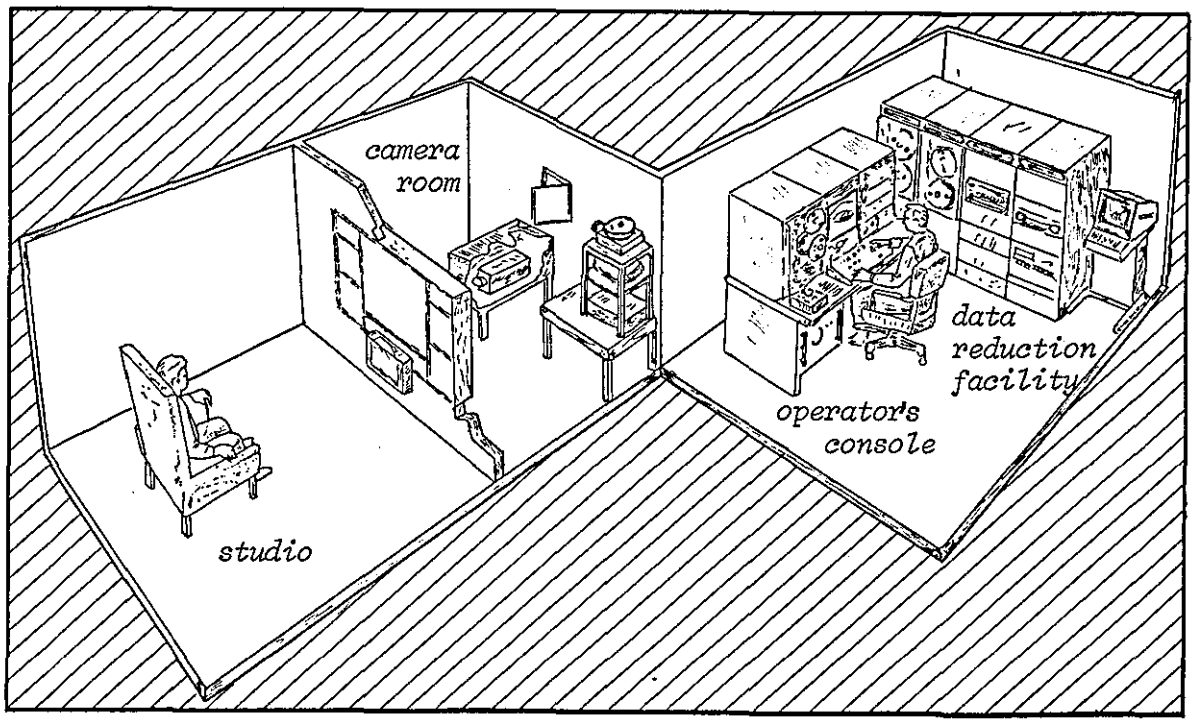


Fig. 8.14: EG & G/Human Engineering Laboratory System (96)

rotate tracking mirrors in front of the TV camera. The mirrors are commanded in such a way as to move the cursor back to the centre of the screen. The mirror servo loop does not respond to rapid eye movements, but is sufficiently fast to track normal head movements. When the computer detects loss of image as a result of an eye blink or a rapid movement, it tries to fit data between the point where data was last received and where it is next received. The uncertainty of the system is within 2 deg. over a 30 deg. field.

8.8.2 PERSEUS

PERSEUS is the acronym for "Programmed Eye-track Recording System and Eye-coupled Ubiquitous Scene-generator". It is a computer-based real time system with associated scenic display capability. The system incorporates a Cornsweet-Crane eye tracker, an SEL-840 minicomputer and two 21-in CRT's exercised simultaneously. One CRT is used as a monitor for the display of various analysis and experimental parameters, while the other is used to display stimulus patterns for viewing by subjects, derived from a digital data base. The subject's head is fixed and his eye position is monitored while inspecting displayed stimulus patterns. An advanced computer graphics system is employed, utilising the same SEL-840 minicomputer. The graphic system is used to generate either unnatural "realistic" scenesor conversely to generate fantastic "events" or sequences of events which would not be possible in the physical world. Images emerging from the data base can be selected or modified at a rate of 30 frames/sec. Each frame consists of a picture made from 1200 to 2000 points and/or lines connecting them.

CHAPTER 9

A REAL TIME SYSTEM FOR EYE MOTION ANALYSIS

9.1 Introduction

So far all real time eye motion analysis systems built by other workers incorporate the use of a minicomputer and hence are expensive and bulky[92,96-100]. VAS with its integral TMS9900 presents itself as a suitable alternative. To the author's knowledge it is the first time a microprocessor-based system has been used to perform all functions pertinent to on-line eye motion analysis. An outstanding feature of this system is its ability to process data in real time, so that information concerning instantaneous eye-point-of-regard is readily available for direct observation.

The eye-sensing device used to monitor the eye movements is an oculometer. Oculometers track two elements of the eye detail: the pupillary disc and the corneal reflection (Section 8.7.5.2), and have the following advantages:

1. Require no attachments to subject's eye, face or head.
2. Cause no physical risks to subjects.
3. They can be made to be unobtrusive.
4. Usable with subjects of different ages and backgrounds.
5. Require low level of subject cooperation.
6. Require a minimum length of time for preparation of subject.

Equipments based on the double Purkinje image method have the same advantages, but are more expensive and require:the placement of optics close to the eyes, higher illumination and very precise alignments

to obtain the proper image position. The output of oculometers is a video signal containing pertinent details of the eye and can be relayed directly to VAS, without the need for additional interface circuitry.

9.2 An Experimental Eye Motion Analysis System

A block diagram of an experimental system for eye motion analysis employing VAS is shown in Fig. 9.1. A subject is seated in a chair with a headrest and keeps fairly still, while his eye is monitored by a laboratory oculometer built in the Department of Human Sciences, University of Loughborough (95).

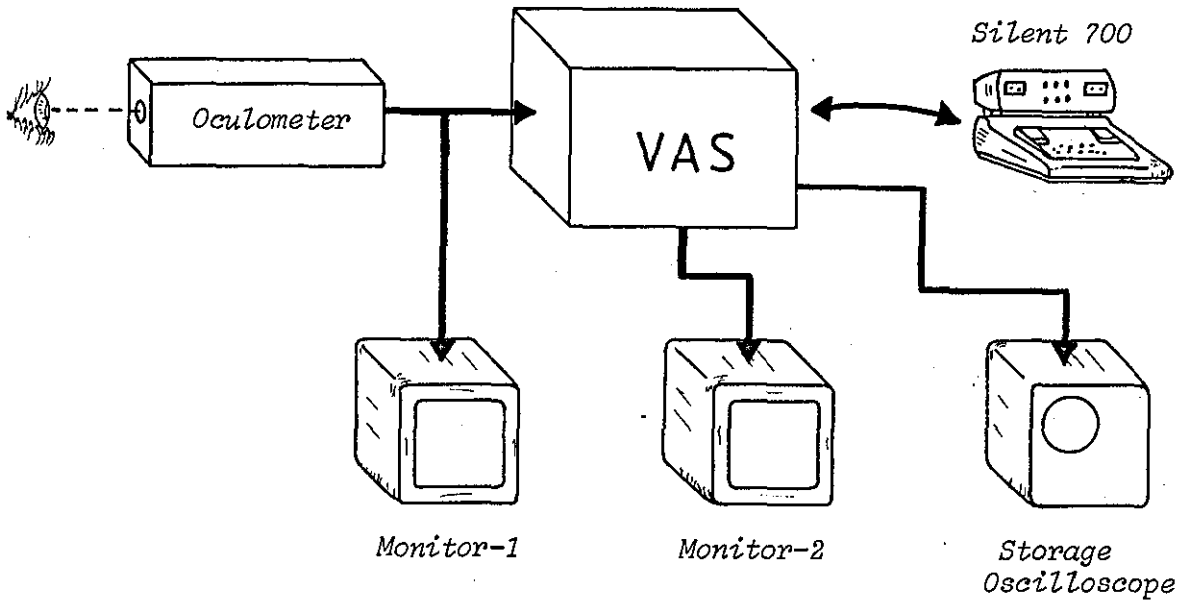


Fig. 9.1: Block Diagram of the Experimental System

The oculometer is connected to VAS, which detects the edges of the pupil and corneal reflections and computes the direction of the

subject's gaze. The result is displayed in real time on a Telequipment DM64 storage oscilloscope. Two Philips LDH 2110,20" monitors are used to display the output of the oculometer, and the edges of the pupillary disc and corneal reflection. At the onset of each experiment, the operator adjusts the oculometer to obtain a clear image of the eye. Following this he manipulates controls on the front panel of VAS to delineate the edges of the pupil and corneal reflections.

A software package has been designed for on-line acquisition of the edges, calculation of the point-of-regard and its display in real time on the storage oscilloscope .The coordinates of displayed data are stored to provide a sequence of points which represent the time history of the point_of-regard for further processing. A set of interactive programs allows an experimenter to reduce the raw data to a form where its salient features can be evaluated, and to display the eye scan patterns between original or processed fixation points on the storage oscilloscope. The Silent 700 is used to obtain a hardcopy of the display with annotations to indicate temporal features of fixation points to help the interpretation of the results. The interactive nature of the programs provides immediate feedback to the experimenter and allows him to proceed directly from monitoring a subject's eye to statistical analysis without any manual manipulation of data.

9.2.1 Description of the oculometer[95]

Fig. 9.2 shows a front view of the oculometer, and Fig. 9.3 illustrates its optical system. A diagram of the arrangement of the components of the oculometer is shown in Fig. 9.4.

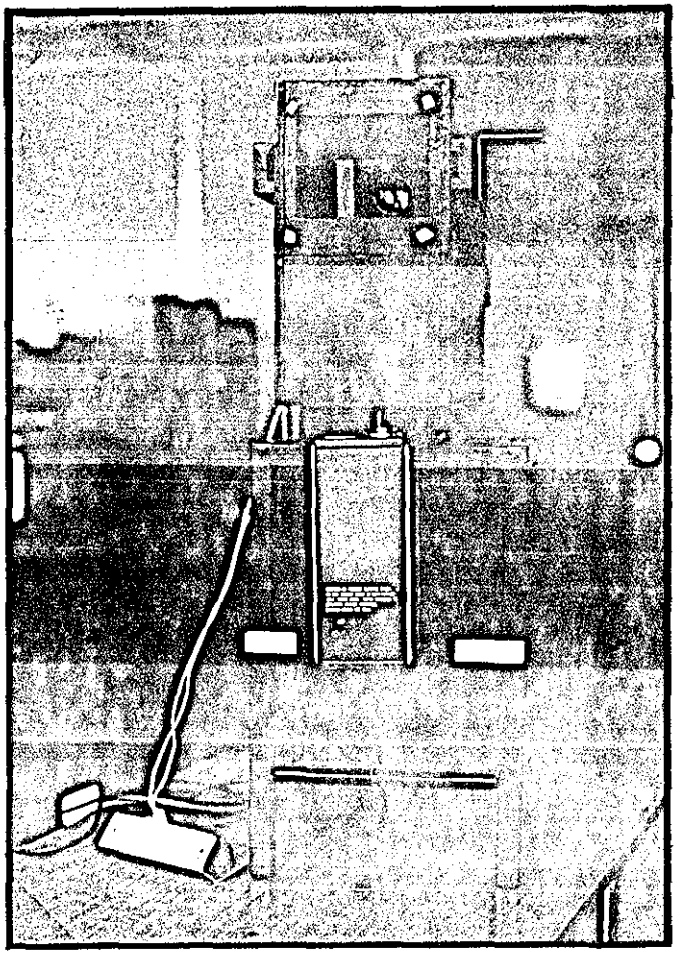


Fig. 9.2: Front View of the Oculometer

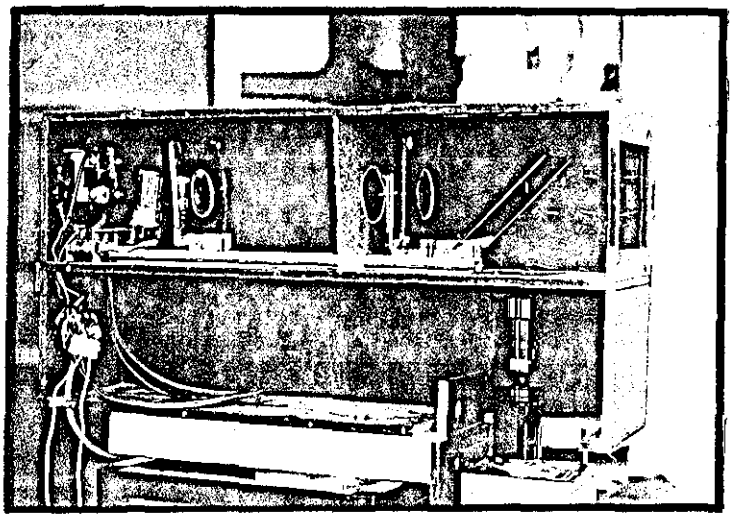


Fig. 9.3: Side View of the Oculometer with the Cover Removed

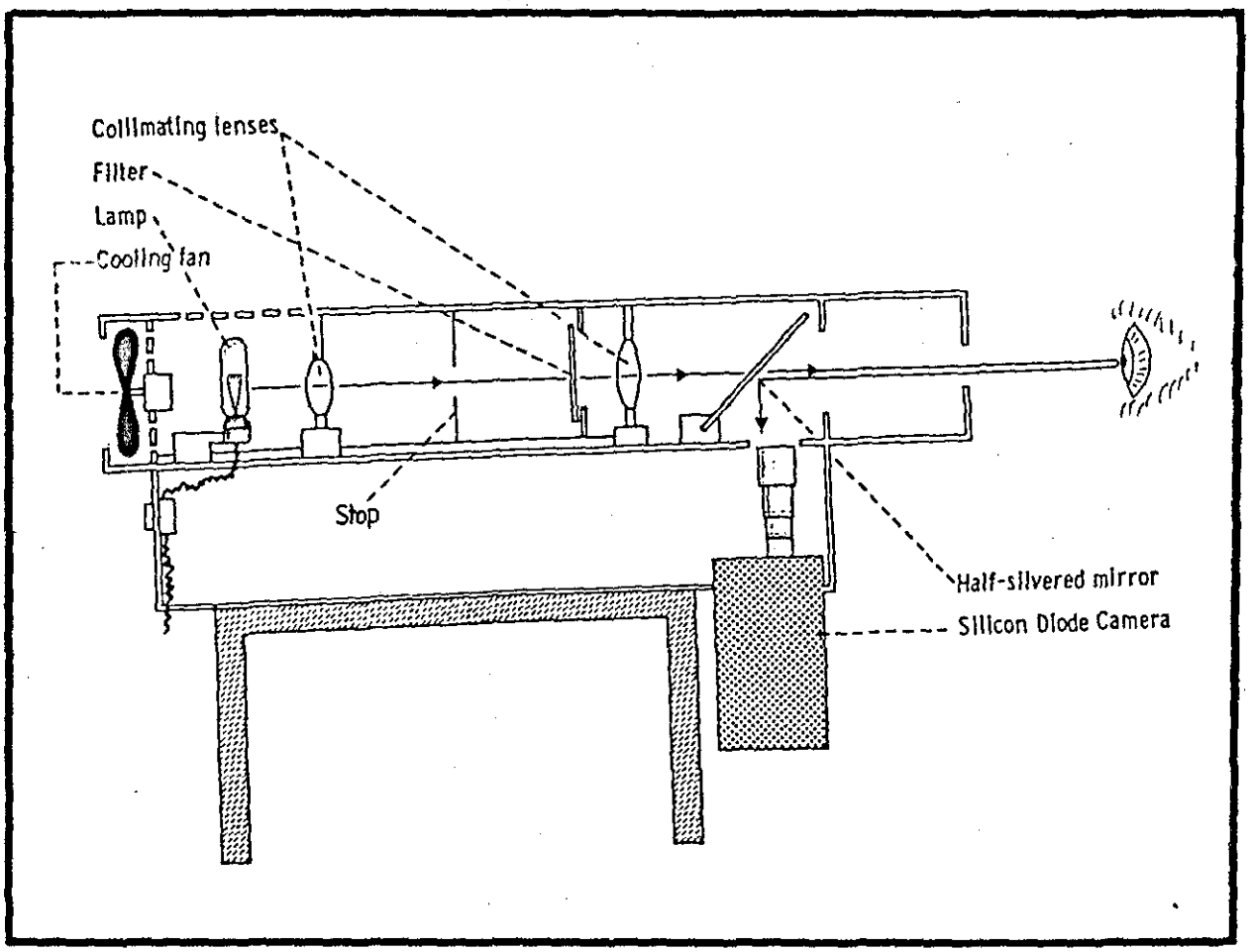


Fig. 9.4: Diagram Showing Components of the Oculometer

Illumination is provided by a 150 watt tungsten-halogen bulb (type SYL-182). The position of the light source and the two collimating lenses are adjusted so that the projected beam is slightly divergent. A filter (Corning CS-7-69) restricts the spectrum of the beam to the near infra-red region. A beam splitter is used to combine the illumination and collection optical paths. A fraction of the radiation reflected from the eye is captured by a TV camera (Shibaden HV40S) fitted with a silicon-diode tube.

The pupil is back-lighted and so the image of the eye as seen by the TV sensor, shows the pupil as a bright disc, with even brighter corneal reflection, against a relatively dark background of skin, sclera and iris. The image of the eye generated by the oculometer is shown in Fig. 9.5. In Fig. 9.6 the blanking level of the video signal is adjusted to eliminate low contrast areas (facial skin, eye-lashes, etc) and to enhance the image of corneal reflection. Unfortunately, the corneal reflex is not as conspicuous as when displayed on the monitor because of deterioration in resolution caused by taking a photograph of the display and then photocopying it.

9.2.2 Performance of the oculometer

The oculometer is an experimental one and was built on a modest budget of £2000[95]. Briefly it suffers from the following design and structural defects:

a) The optical assembly is not mounted on a heavy rigid framework and the angles at which some components are adjusted are not well defined (Fig. 9.3). When the instrument is switched on the cooling fan causes the whole unit to vibrate and the image of a subject's face jitters. To overcome this the fan is provided with a separate switch, and is switched off whenever eye movements are recorded.

b) In a strict sense, the illuminating and imaging optics do not provide collimated light. Radiation is not emerging from a point source and the circular hole in the stop is very badly cut.

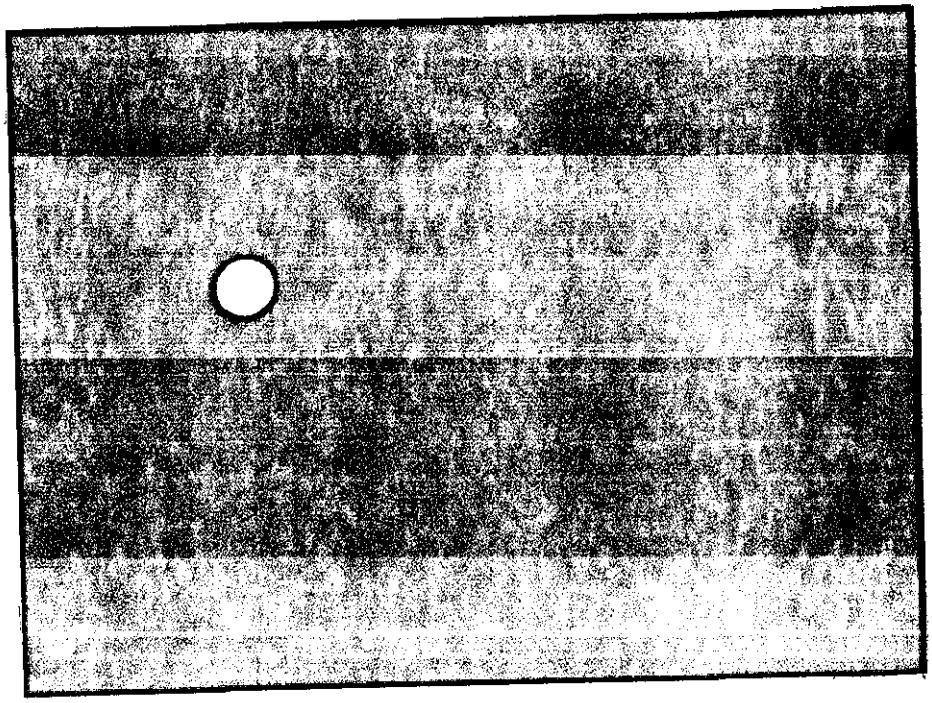


Fig. 9.5: Image of a Back-lighted Pupil

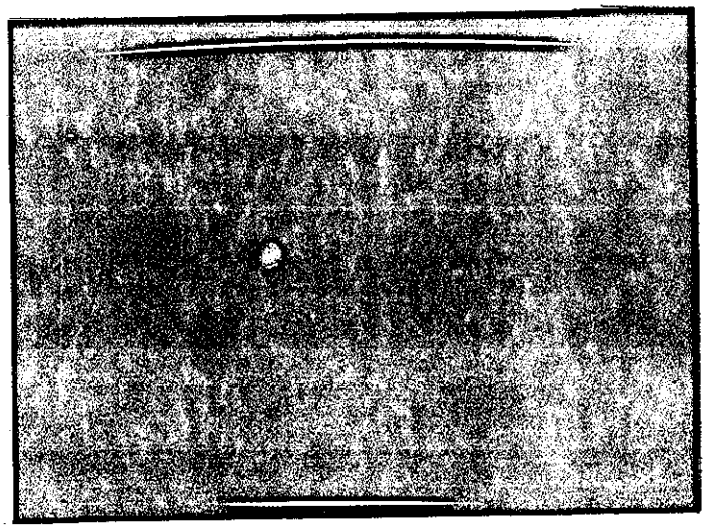


Fig. 9.6: Image of a Back-lighted Pupil after Adjusting the Blanking Level of the Video Signal to Eliminate Low Contrast Facial Details

c) Illumination in the plane of the subject's eye is not even. When the line of sight is directed towards the beam splitter, filaments of the tungsten bulb can conspicuously be seen, glowing within a dull red disc.

d) The stop and the optical bench do not fit tightly into the box and an appreciable amount of visible light is collected by the camera, whose response covers part of the visible light spectrum.

e) The optical system is not coaxial and the camera has to be tilted to bring an eye into its field of view.

f) The camera tube has many burn spots which have decreased its sensitivity.

These defects have rendered the oculometer far from being a precision instrument. The poor structure of the base and the attachments which support the optical components made it extremely difficult to readjust and align the optical system.

The range of operating angles is approximately $\pm$10 degrees vertical and $\pm$10 degrees horizontal. This is the angular range subtended by a 22 x 22 cm scene viewed at 62 cm, and is achieved when the scene is fixed to the front side of the oculometer, with a hole in the middle along the illumination and collection optical paths.

A measure of resolution of oculometers can be derived from the relationship

$$D = K\sin\theta$$

mentioned earlier in Section 8.7.5.2. Since D can be determined to one TV line pair, then the angular resolution θ is given by

$$\theta = \sin^{-1} \frac{1}{K*L}$$

where L is the number of raster lines / mm at the image of the eye[82]. Accordingly resolution of an oculometer depends both on the number of scan lines and the size of the pupillary disc. The larger the number of lines intersecting the pupil the better is the resolution. On the other hand, the bigger the image the less head movements the system can tolerate. An optimum size is usually based on a trade off between accuracy and allowable head movements. However, in many systems the pupillary image is made to cover about a third of the total area of the screen, permitting reasonable head movements.

In the oculometer used in proving trials of the system,the pupil occupies about 30 scan lines per field, out of the total of 287.5 i.e. about 10% of the whole screen. The number 30 is an approximate one as the actual size of the pupil varies from subject to subject. For 10 degrees of eye rotation in the horizontal direction, it has been found that the corneal highlight moves about 10 pixels, giving a potential angular resolution of 1 degree. In the vertical direction the measured resolution is half that amount due to the interlaced scanning (i.e. about 2 degrees).

9.3 Experimental Constraints

In the proving trials,the limitations inherent in the oculometer and other factors have imposed the following experimental constraints:-

1. Experiments are performed under very dim ambient illumination. This illumination is collected by the TV camera and any increase in its level compounds the problems already created by stray

visible light within the oculometer, and impairs the performance of the system.

2. The amount of radiation reflected from the eye changes appreciably with its rotation. In some instances the pupil becomes very bright and corneal reflection disappears altogether. An operator has to intervene many times during an experiment to readjust the level of the video signal to restore the corneal reflex.

3. The maximum range of eye movements is $\pm$10 degrees horizontal and vertical. This can be achieved by attaching stimulus patterns to the fore plate of the oculometer.

4. Limited space is available for performing experiments.

9.4 Edge Detection of Pupil and Corneal Reflections

Eye details as conveyed by a raster line intersecting both the pupil and corneal reflections are shown in the top trace of Fig. 9.7. This trace represents the video signal which is fed to VAS. An experimenter alters the D.C. levels of this signal and the reference voltage of the ADC such that the dynamic range of the video signal is digitised and differentiated to obtain the middle trace. The signal represented by the middle trace is processed to eliminate features with low contrast values. The only information contained in the processed signal (bottom trace) is the edges of the pupil and corneal reflections, and is used to generate the coordinates of the edges which are collected by the microprocessor. The adjustments performed by an experimenter initially to obtain the edges are simple and take a very short time, usually

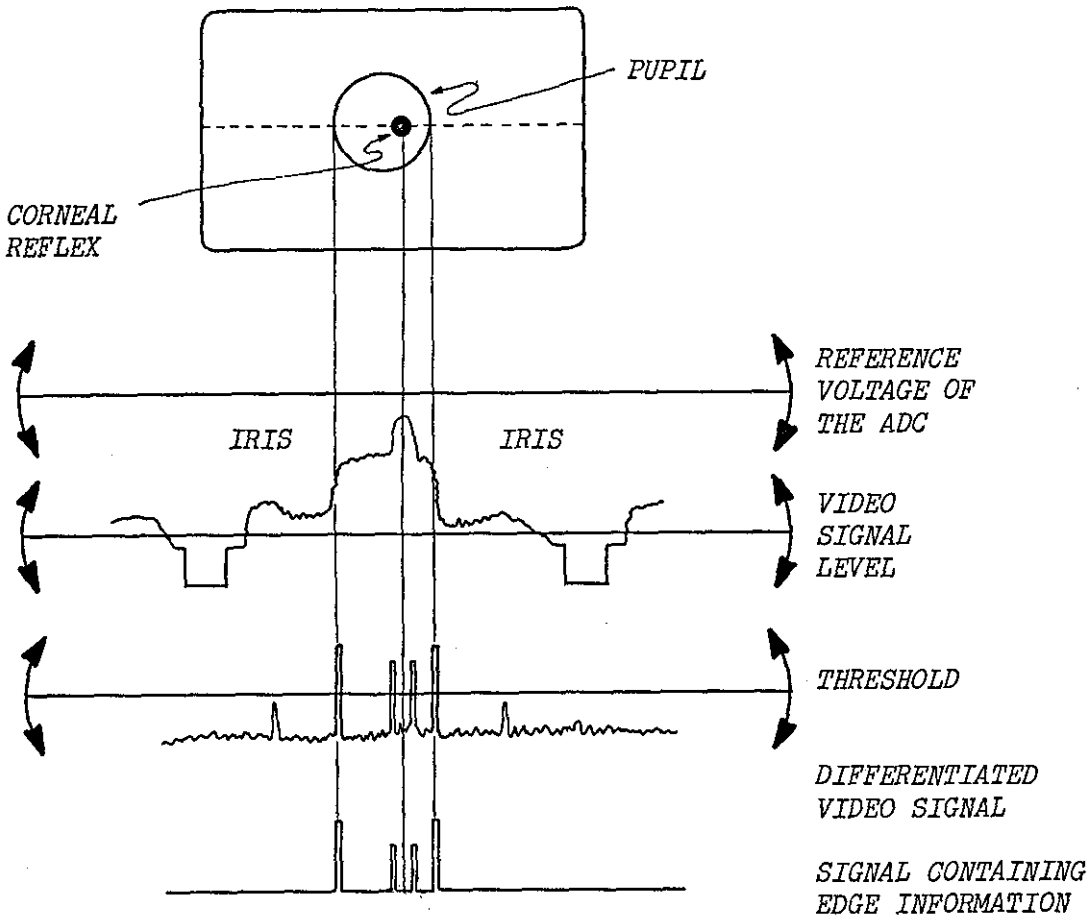


Fig. 9.7: Edge Detection of Pupillary Disc and Corneal Reflection

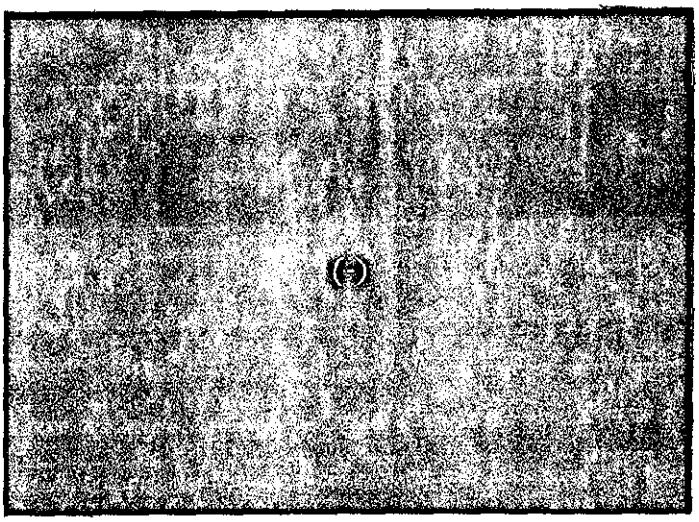


Fig. 9.8: Edges of Pupillary Disc and Corneal Highlight

in the order of a few seconds.

Edges detected by VAS are shown in Fig. 9.8. The quality of the edges demonstrates the ability of the edge detection technique to extract the pupillary disc and corneal reflex from the output of the oculometer, whose performance is degraded by:an inaccurate alignment of its optical system and a camera tube suffering from many spot burns.

9.5 Determination of Pupil Centre and Centre of Corneal Reflection

Fig. 9.9 is a schematic diagram of projections of an eye image on the sensor plane, showing the measurements to be taken for determining point-of-regard. The two measurements are the pupil centre and the centre of the corneal reflection cluster. From these two measurements the components Δx, Δy of the vector $\overline{R}$ are computed . These are a measure of the point-of-regard.

The centre of the pupillary disc can be determined by either evaluating mid-points of mutually perpendicular chords, or by locating the mid-point of the horizontal diameter of the disc. These two methods are explained in detail in Section 7.3, and both give good estimate of the centre of the pupil. The corneal reflection appears as a small cluster within the pupillary disc (Fig. 9.8) and its centre can be determined by averaging the x- and y-coordinates of the cluster points. By subtracting the coordinates of the centre of the corneal reflection from those of the centre of the pupillary disc, the two components Δx and Δy of the vector $\overline{R}$ are obtained.

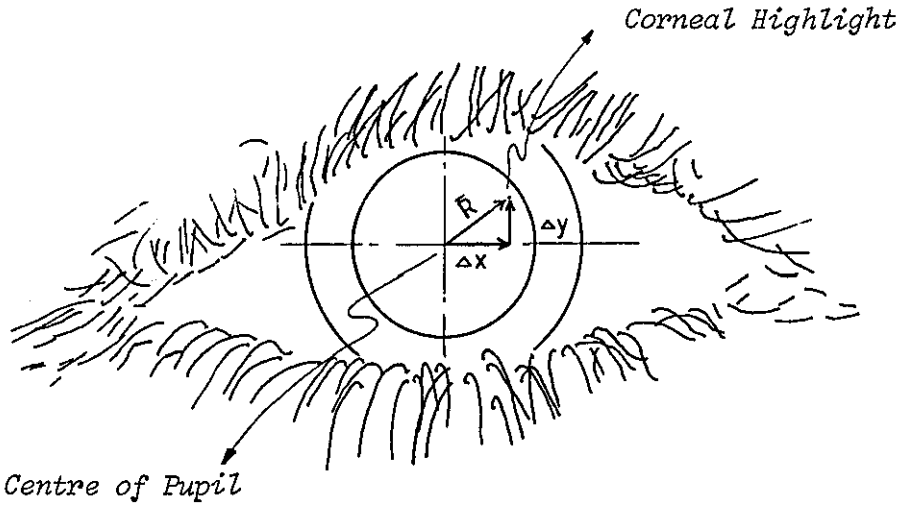


Fig. 9.9: Measurements Taken in the Sensor Plane to Determine the Point-of-Regard

9.6 Evaluation and Display of Instantaneous Point-of-Regard

A program, called COLDAT, stores the edges of pupillary disc and corneal reflection, computes, displays and stores instantaneous point-of-regard in real time. It occupies about ½ K words of memory. Fig. 9.10 illustrates the overall structure of the program. A brief description of the program is given in the following subsections, together with simplified flowcharts.

9.6.1 Data acquisition and storage

Fig. 9.11 is a flowchart of the first part of COLDAT which is responsible for on-line acquisition and storage of coordinates of edges of pupillary disc and corneal cluster.

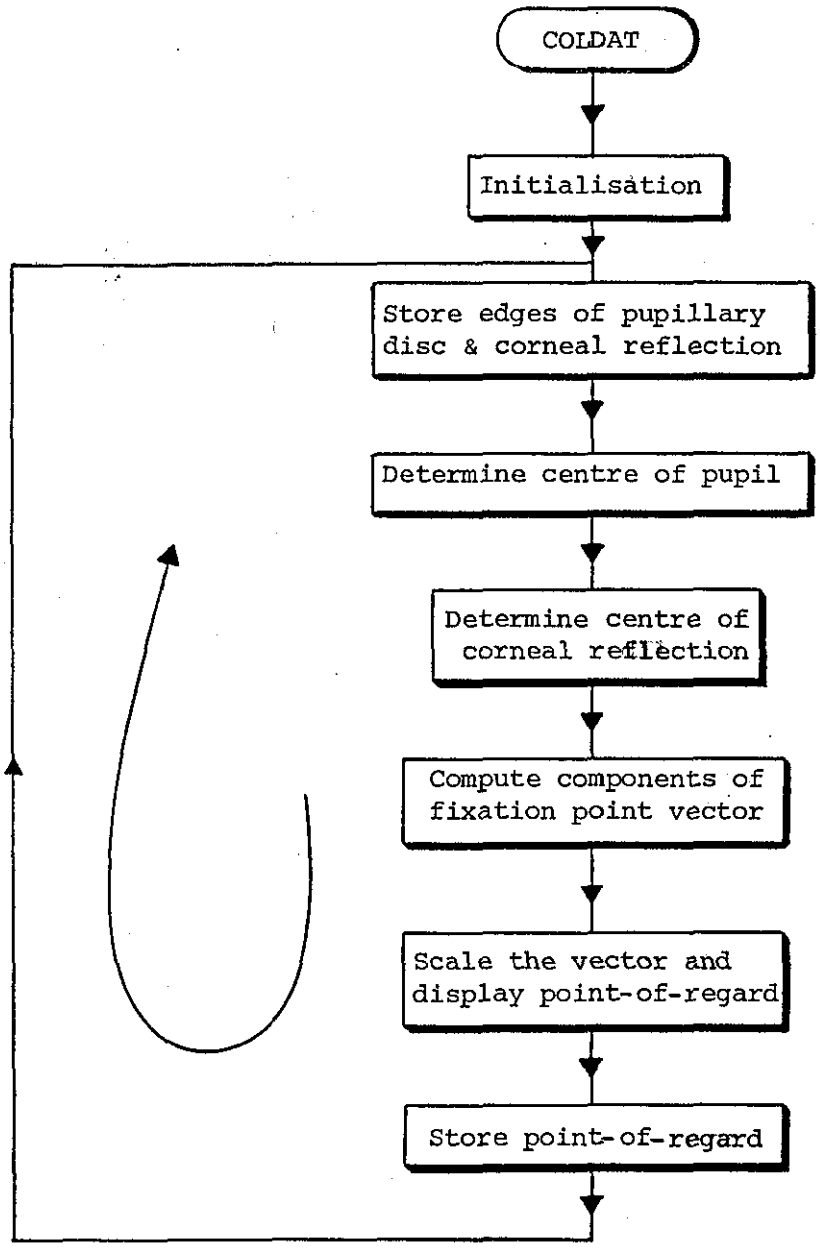


Fig. 9.10: Basic Flowchart of COLDAT

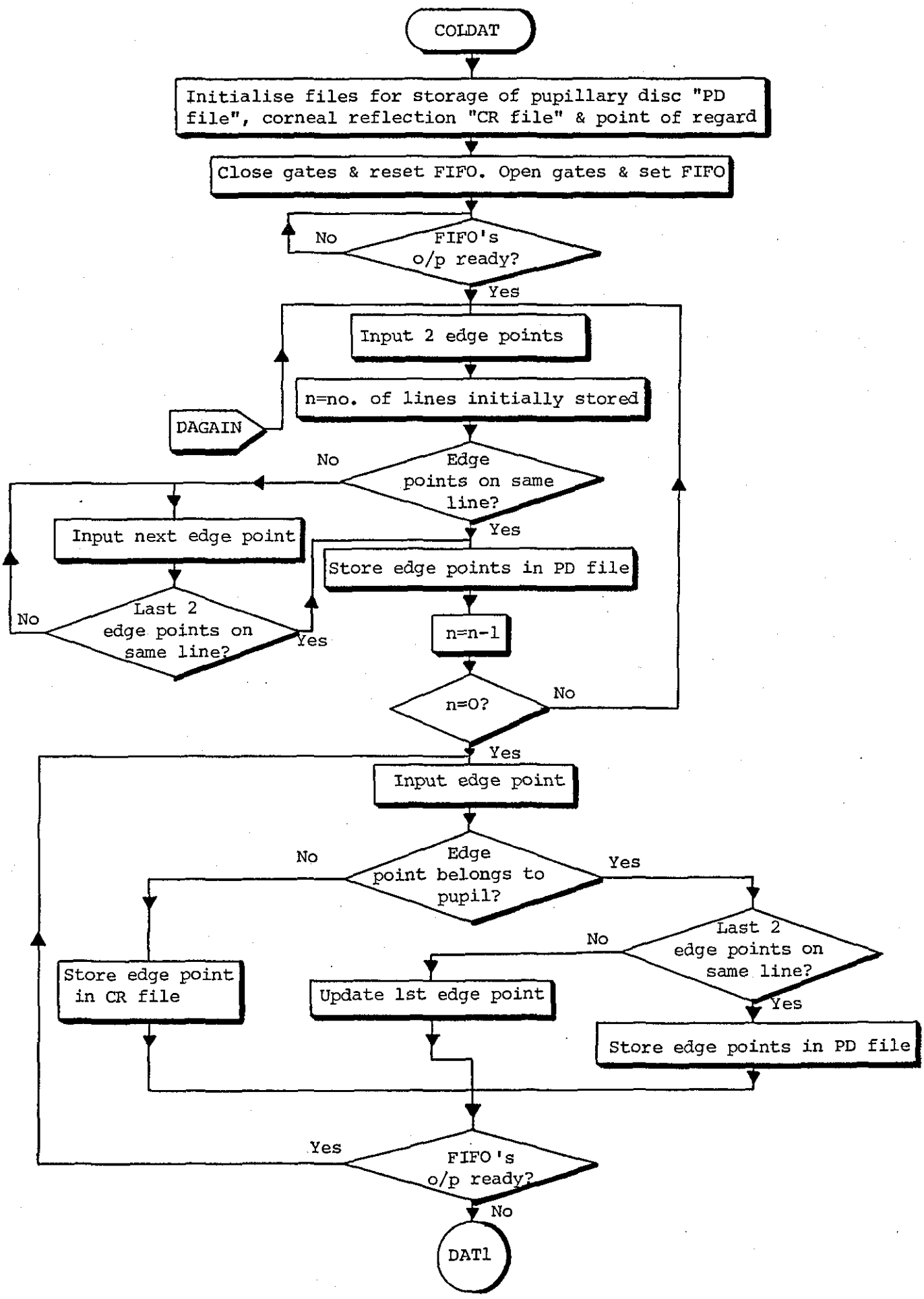


Fig. 9.11: Flowchart of COLDAT (data acquisition and storage)

Initialisation consists of:opening files for storage of coordinates of edges and instantaneous fixation point, clearing the FIFO buffer store,and setting counters and test flags. Edges of pupillary disc and corneal cluster are assigned to two separate files. As edge points are collected the program decides whether they belong to the pupillary disc or corneal cluster so as to store them in different files. Edge points are routed to separate files for quick data retrieval and processing to determine the centroids of the pupillary disc and corneal cluster. This approach requires less storage space and execution time than when the whole set of raw data is stored in one file and then separated and stored in two files prior to processing.

The method by which the two sets of data points are separated is derived from the pattern of the edges. A typical representation of such a pattern is shown in Fig. 9.12. The program accepts an edge point as belonging to the pupil if its distance from a previous point on the circumference of the pupil is less than a specific number of pixels. This distance is adjustable and for most cases a distance of

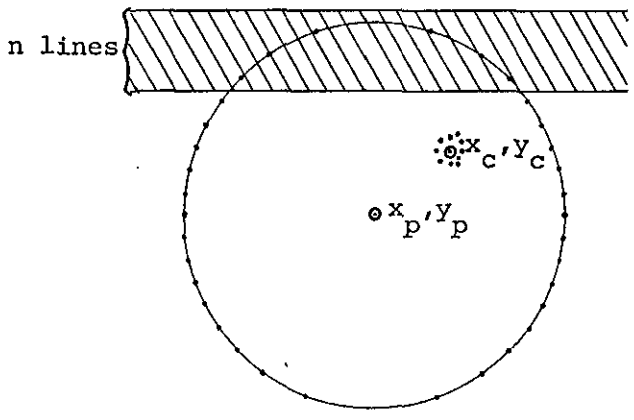


Fig. 9.12: Edges of Pupillary Disc and Corneal Cluster

3 pixels has been found to be adequate. To attain a quicker rate of storage of data, edge points lying on the first n lines (Fig. 9.12) are stored without testing them to find whether they belong to the pupillary disc or corneal cluster, because of the non-existence of corneal edge points on these lines. The number n is adjustable, and a value of 4 or 5 is found to be suitable for most cases. As data is separated into two sets, concerning the pupil, the program only stores pairs of edges lying on the same line of the raster scan. This is achieved by comparing the y-coordinates of two consecutive edge points. If they are equal both edge points are stored. Otherwise, the first edge point is updated and compared with the next incoming one, until two points lying on the same horizontal line are found. Thus single edge points are rejected as they occur in very few instances and their rate of occurence does not warrant the punishment of real time interpolation.

9.6.2 Evaluation and display of instantaneous point-of-regard

Fig. 9.13 shows a flowchart of the second part of COLDAT. This part determines the centroids of the pupillary disc and corneal reflex, computes, displays and stores instantaneous point-of-regard.

If data collected does not contain enough information to determine the eye position, the program rejects it, sets pointers to the starting addresses of data storage files and waits for a new set of data. Lack of enough information is mainly due to blinks and distortion in the eye image as a result of changes in amount of radiation reflected from the eye (Section 9.3).

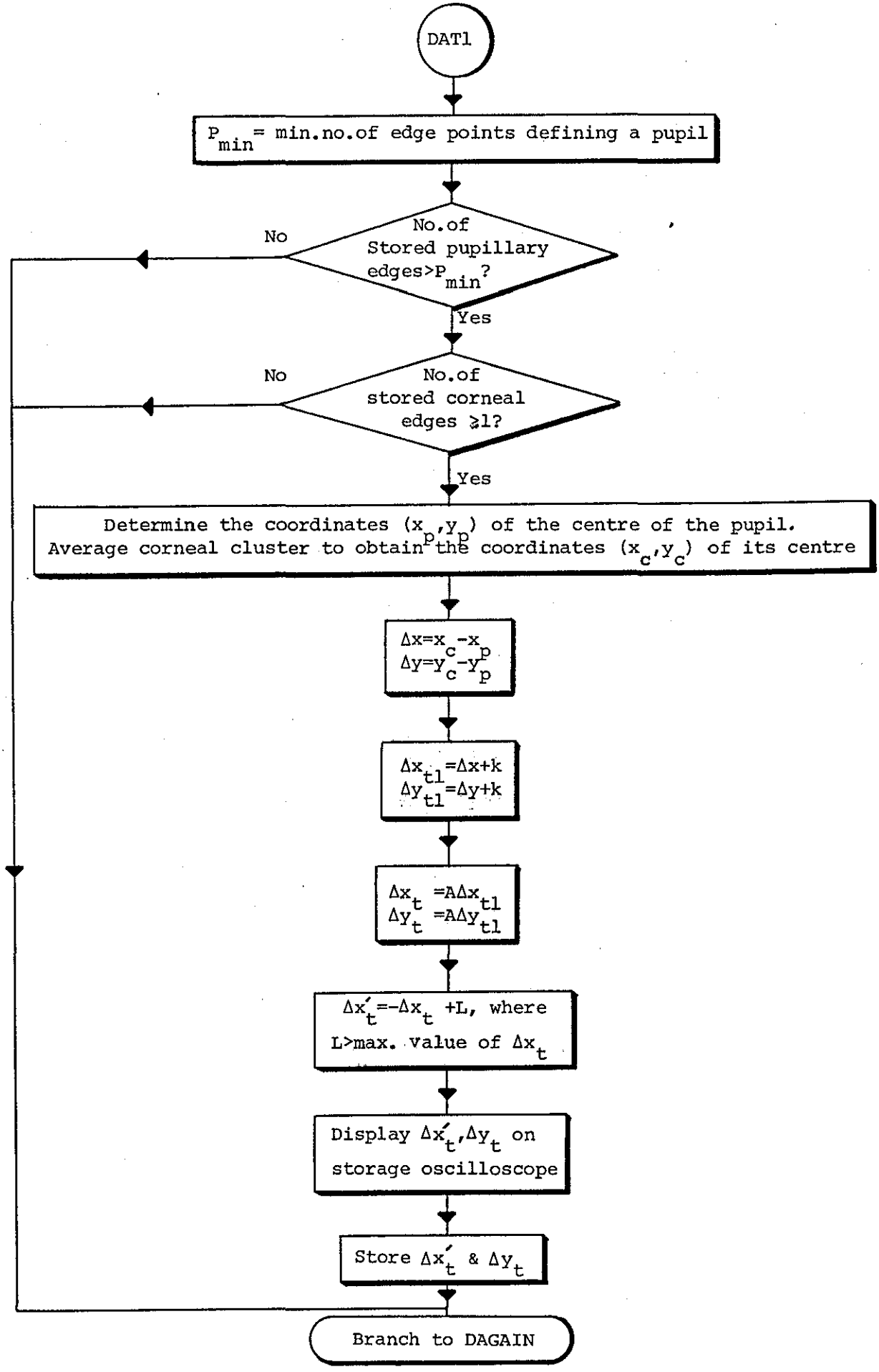


Fig. 9.13: Flowchart of COLDAT (Computation and display of point-of-regard)

After determining the centroids of the pupil (x_p, y_p) and corneal reflection (x_c, y_c), the program computes Δx and Δy, the two orthogonal components of the vector $\bar{R}$. Then the following transformation is performed to render fixation points more convenient for display on the storage scope.

$$\Delta x_t = A(\Delta x + k)$$
$$\Delta y_t = A(\Delta y + k)$$

where A and k are constants. k is added to offset negative values of Δx and Δy, and A is a scaling factor to increase the size of the display, to make it cover a large part of the screen. Values of A and k are adjustable.

Before a pattern is displayed, the program performs another transformation to obtain its mirror image about a perpendicular plane. To explain the reason for this, consider an eye viewing two points (A & B) lying on the same horizontal line (Fig. 9.14). For A the centre of the pupil is to the left of the centre of corneal reflection, while for B it is to the right. Consequently the x-coordinate of fixation point A is greater than that of point B. So when displayed on a storage scope, with the origin of coordinate axes at the bottom left hand corner, point A appears to the right of point B. When directing the eye line of sight towards two points (D &E) on the same vertical line, then for D the centre of corneal reflection is below the pupil centre, while for E it is the opposite. With origin at the top left hand corner of a TV monitor screen, the y-coordinate of fixation point D is greater than that of point E. Hence, when these two fixation points are mapped on a storage scope their position relative to each other is

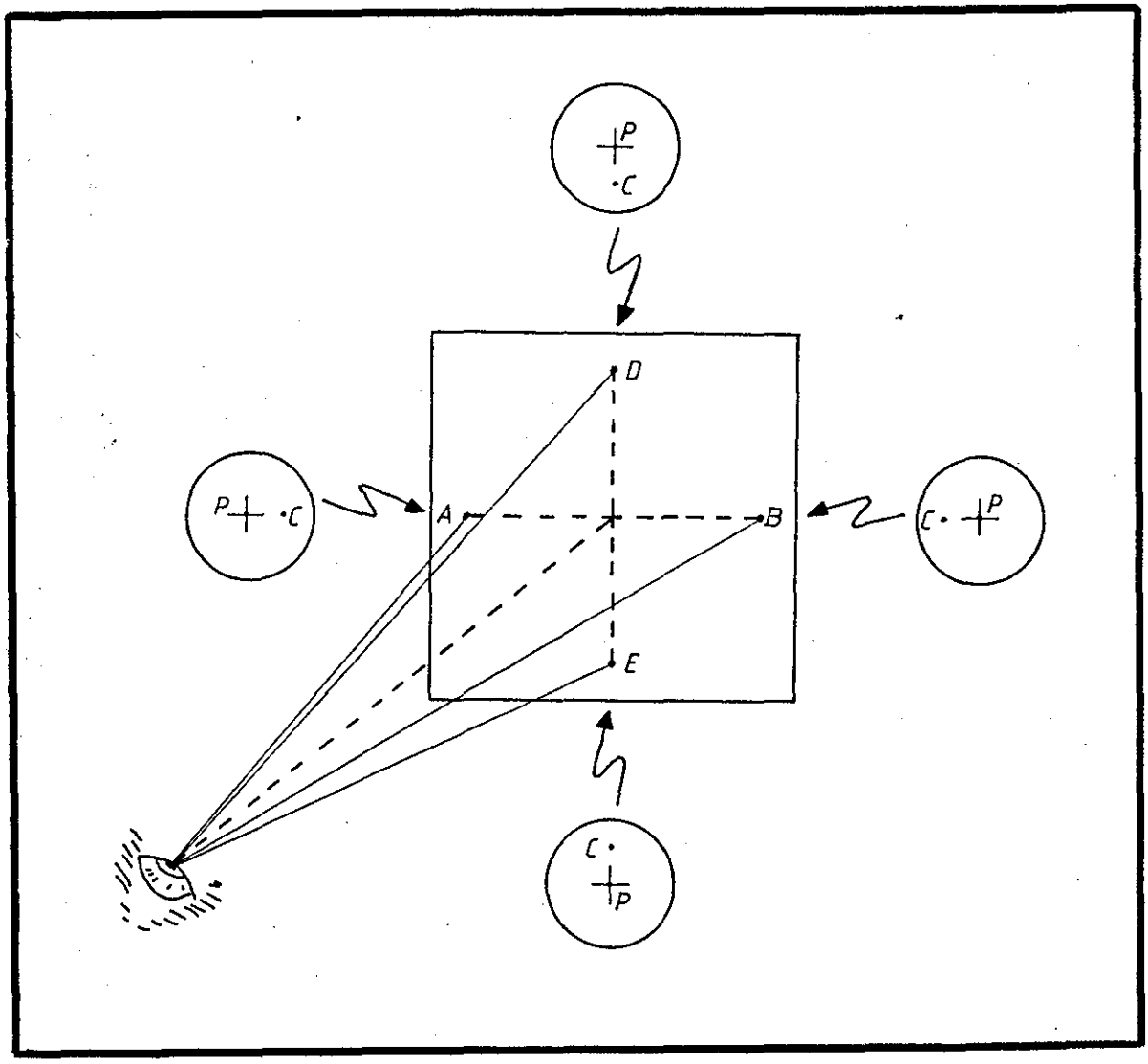


Fig. 9.14: Position of Centre of Corneal Reflection Relative to Centre of Pupil

the same as in the viewed pattern.

For the reasons stated above, fixation points displayed on a storage scope, without modifying them, are mirror images about a vertical axis of fixation points in the plane of a stimulus pattern.

The program restores fixation points to their proper shapes by subtracting their x-coordinates from a constant L. At the beginning of each experiment a subject is asked to fixate his gaze at 4 points arranged at the corners of a square on the screen on which a stimulus pattern is to be fixed. Data collected is calibrated through the choice of suitable values of A, K and L, and is displayed on the oscilloscope.

The part of the program responsible for displaying point-of-regard on the storage scope is similar to the program described in Section 6.3. Points-of-regard are displayed as a sequence of dots on the oscilloscope screen. Here results are displayed in real time by omitting the instructions controlling the flag circuit.

9.6.3 Remarks

Information for obtaining point-of-regard is contained in less than one field. The TMS9900 (3 MHz speed) takes about 1¼-1½ field duration to process raw data, display and store results. Consequently, instantaneous fixation points are given at a rate of 25 samples/second. The new generation of 16-bit microprocessors, which are faster and more powerful than the TMS9900 (e.g. the 8 MHz 8086-2 and the 10 MHz Z8003A microprocessors[101,102]) can perform all functions pertinent to real time evaluation and display of point-of-regard within one field duration, thus achieving a rate of 50 fixation points per second, available for display.

9.7 On-line Analysis of Point-of-Regard Data and Display of Results

When a subject is asked to fixate his gaze at a particular point in space, fixation points gathered over an interval of time form a cluster around that particular point. Accordingly, data collected and stored when he is examining a stimulus pattern consists of clusters, the sequence and distribution of which show the manner in which he has examined the pattern. Since fixation points arrive at a constant rate of 25 samples per second, then the total number of points within each cluster indicates the dwell time, i.e. the time the subject has spent examining a particular point.

Programs have been written to reduce voluminous raw point-of-regard data and display the results in a form amenable for easy analysis. Data reduction is achieved through the use of a clustering algorithm. The algorithm acts to remove spurious points, when they exist, and compresses the complete set of raw data to a sequence of annotated fixation points. Clusters of many points are replaced by a single point, the cluster mean. The sequence of cluster means is numbered to show the order of their occurrence. A second label, representing the dwell time within a cluster is attached to each cluster mean. The dwell time gives an estimation of the degree of interest at each fixation point or area around the cluster mean. Straight lines joining centres of clusters are interpolated. Both cluster centres and interpolated lines are displayed to obtain fixation-point scan patterns which enable an operator to follow a subject's eye as it travels from one cluster centre to another. A permanent display of scan patterns can be obtained using the Silent 700. This permanent display, besides cluster centres joined by straight lines, contains annotations affixed to each cluster centre.

The annotations indicate the sequence at which clusters occur (1, 2, 3...) and the dwell time within each cluster domain.

9.8 Clustering Criteria[3]

Clustering criteria can be guided by intuition and experience or they may be based on minimisation or maximisation of a certain performance index. One of the most often used indices is the sum of the squared errors index, given by

$$J = \sum_{j=1}^{N_c} \sum_{X \in S_j} ||X - m_j||^2$$

where N_c is the number of cluster domains, S_j is the set of samples belonging to the j^{th} domain, and

$$m_j = \frac{1}{N_j} \sum_{X \in S_j} X$$

is the sample mean vector of set S_j or in other words the cluster mean. In the second equation N_j represents the number of samples in S_j. The performance index J is the sum of the squared distances between cluster mean and points within that cluster summed over all clusters. Domain boundaries are established to minimise J and clustering is applied to all points in parallel.

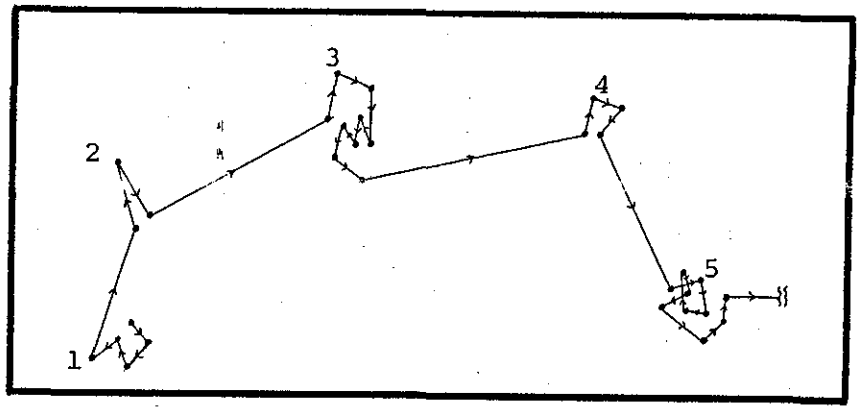
Eye point-of-regard data is sequential and is a function of 3 variables; two spatial coordinates and a temporal coordinate (time). Fixation points have to be grouped and a measure indicating temporal component of each group should be established. Because of the

sequential nature of data and its temporal feature, points can be deleted but not interchanged. Thus a cluster-seeking algorithm must proceed in a sequential manner. Most standard approaches such as the minimum mean squared errors do not work on sequential data.

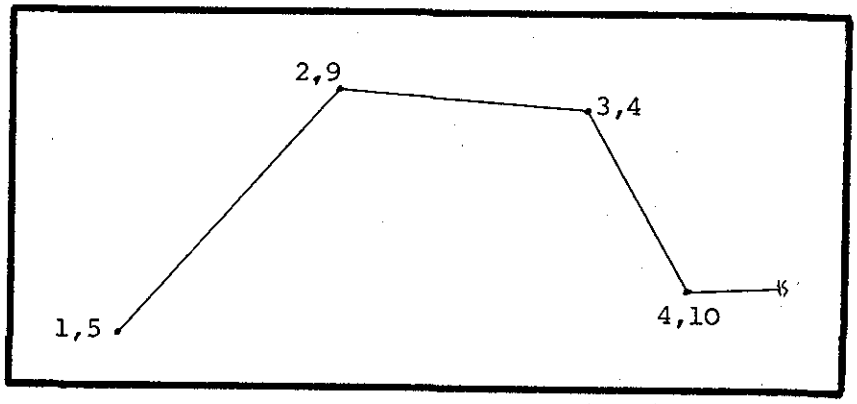
As it is difficult to define a performance index which can be minimised or maximised by an algorithm manipulating sequential data, then a solution based on intuition and dependent on the type of data being analysed is the only alternative. The solution utilises an arbitrary non-negative distance metric threshold in addition to threshold testing of the temporal component.

A typical representation of fixation-point scan patterns is shown in Fig. 9.15(a). In regions 1, 3 and 5 large concentration of fixation points in a relatively small area forms the main data clusters. This indicates that a subject has been examining interesting features in those regions. Number of fixation points in regions 2 and 4 shows that the subject has little interest in parts of a stimulus pattern falling within them, so he fixates his gaze for a short while and then moves it elsewhere. The excursion in region 2 implies that the subject's eye has gone astray for a short time and then brought back to a point of interest,

A clustering algorithm with two parameters, one indicating the maximum size of the cluster and the other the minimum number of data points within a cluster, i.e. minimum dwell time, can be used to simplify data represented by Fig. 9.15(a). Such an algorithm works on sequential data and assumes that the first point is a centre of a cluster. Then it computes the distance between it and the next point. If this distance is less than a threshold that point is assigned to the first



(a)



(b)

Fig. 9.15(a) Fixation-Point Scan Pattern
(b) Data Reduction by Clustering

cluster domain and the cluster centre is updated. Otherwise, another cluster with a new centre is started. In a similar fashion the algorithm manipulates other points of the sequential data. After each cluster is established, its mean and the dwell time are calculated. Then the dwell time is compared with a threshold. If it is greater than the threshold the cluster is labelled with two numbers, indicating the sequence of its occurrence and the dwell time, and is included in the results; otherwise, it is deleted. The ability of

this approach to compress point-of-regard data and give information about its salient features, is shown by the sketch in Fig. 9.15(b). Here,by setting the minimum dwell time at 4, region 2,where the subject is not showing sufficient interest,is eliminated.

The algorithm requires only one pass through a sample set to establish the cluster domains and is therefore computationally attractive. In practice,the procedure normally needs experimentation with some values of the two thresholds to gain useful insight into the geometrical distribution of fixation points. Information obtained in the first or second pass is usually adequate to estimate optimum values for the thresholds. Moreover,choice of thresholds is to some extent influenced by the distribution of raw eye track data over a stimulus pattern, for instance,whether fixation points form characteristic natural cluster or whether they form a continuous track.

9.9 Description of Programs for Data Processing and Display of Results

Interactive programs for point-of-regard data processing and display of results occupy about 2K words of memory and perform the following functions:-

1. Data reduction by a clustering technique.
2. Generation of straight lines joining centres of clusters.
3. Adjustment of centres of clusters not falling at the intersections of straight lines.
4. Manipulation of data to render it in a format suitable for hardcopy display.

5. Scaling of processed data to obtain a larger display, and increase the number of interpolated points representing straight lines.
6. Hardcopy display of annotated fixation-point scan patterns.
7. Display of scan patterns on a storage oscilloscope.

The program responsible for displaying results on the storage oscilloscope (subroutine VIDUD) is explained in Section 6.3. Brief descriptions of the other programs are given in subsequent subsections.

The last six functions are performed by the microprocessor to generate pictorial representation of the results. In doing so, the storage oscilloscope and the Silent 700 are used as passive graphics devices. A passive graphics device draws patterns under computer control, i.e. it allows the computer to communicate graphically with the user.

The problem of displaying the results could have been solved by making use of relevant computer graphics techniques, which are based on matrix algebra for representation and transformation of lines, and data base structures for storage of data arrays[103]. These techniques are rather complex and require large storage space. For this reason they are avoided, and an approach employing basic mathematical procedures is adopted.

In fixation-point scan patterns a line is defined by its end points. Because a hardware vector generator, which allows the drawing of lines with minimum amount of data is not available, vector drawing is achieved using software. The software interpolates points between two end points and stores them in a file (array) prior to being used to present scan patterns. A vector is constructed as

a series of points. If points are plotted close together, they will appear to the eye as a solid line.

9.9.1 Data Reduction by a clustering technique

A flowchart of a subroutine for data reduction, labelled DATRED, is shown in Fig. 9.16. The subroutine processes data sequentially and accomplishes data reduction employing two parameters, R and T, specifying the maximum size of a cluster and the minimum dwell time respectively. To illustrate the procedure, with notation to Fig. 9.17, the first sample (A), is assumed to be belonging to a cluster and is set as a datum. Then point B is fetched and the distance metric D_{AB} is computed according to the relation;

$$D_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Since D_{AB} is less than R, B is absorbed into the cluster, and the sum of the coordinates $x_A + x_B$, $y_A + y_B$ is calculated and stored, to be used later in computing the cluster mean. The datum is updated by shifting point A to B and the process is repeated sequentially for other points. Point C is absorbed into the cluster as D_{BC} is less than R. Then x_c and y_c are added to the accumulated sum of the x- and y-coordinates respectively. Similarly, point D is absorbed into the cluster. Concerning point E, D_{DE} is greater than R, so the assumption is that it belongs to a new cluster. At this stage the program calculates the cluster mean C_1 according to the relation;

$$C_{1x} = \frac{\Sigma x}{n}$$

$$C_{1y} = \frac{\Sigma y}{n}$$

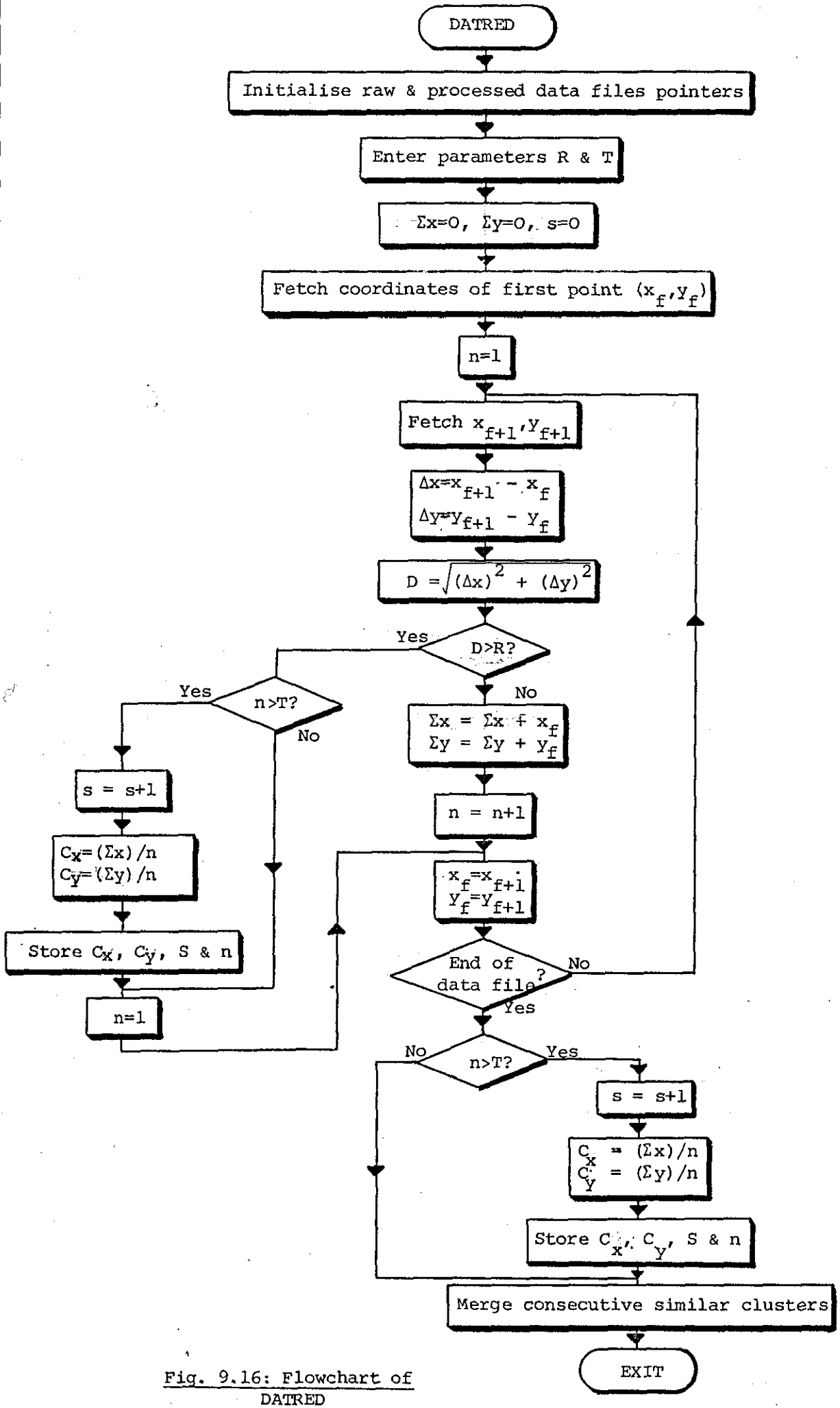


Fig. 9.16: Flowchart of DATRED

where C_{lx}, C_{ly} are the components of the mean, n is the total number of data samples within the cluster and x,y are coordinates of sample points.

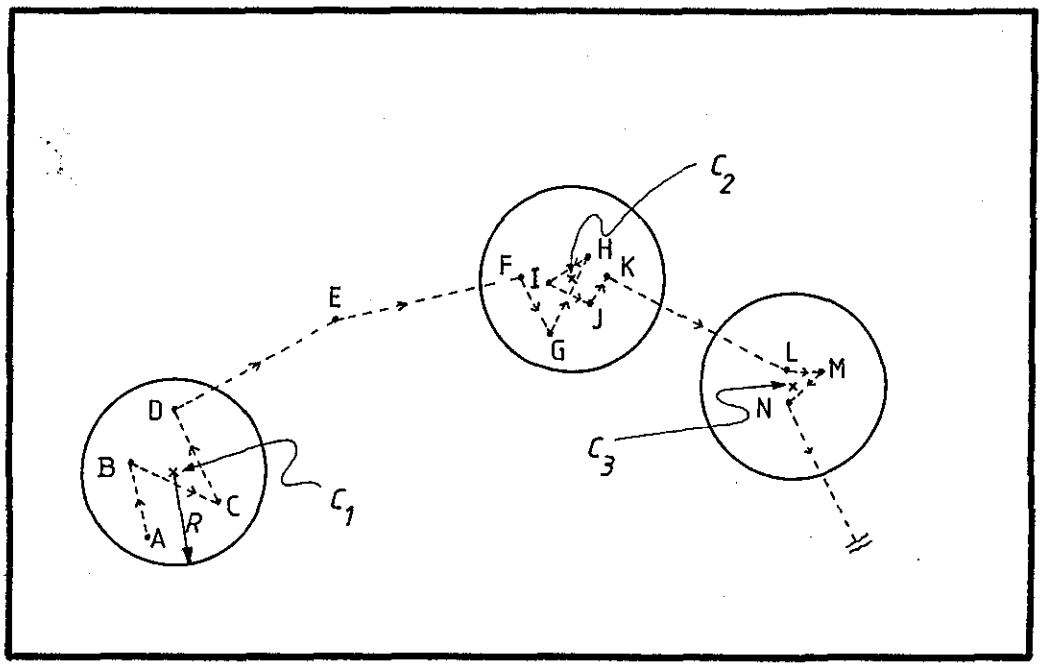


Fig. 9.17: Sample Points to Illustrate Clustering Technique Employed by DATRED

The program shifts the datum to point E and fetches point F. D_{EF} is greater than R and point E will be rejected as the dwell time threshold is usually set greater than 1. The program replaces E by F and the procedure is repeated for subsequent points. After examining all data samples, cluster centres C_1, C_2 and C_3 are available with subscripts denoting their sequence of occurrence and the dwell time

within each cluster. Since eye fixation points are given at a rate of 25 samples/second, then dwell times are in units of 40 msec. In the above case, if the minimum dwell time is set equal to 4, then C_3 will be deleted.

Because of the sequential nature of the analysis, when there is a large excursion in an area where there is a concentration of data samples (Fig. 9.18), the program identifies two cluster domains in the same area instead of one.

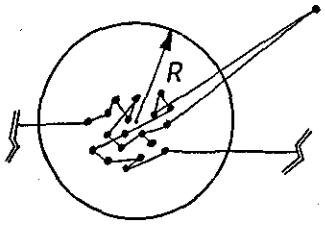


Fig. 9.18: A Sample Pattern with an Excursion

To avoid this situation, the program upon completion of its cluster seeking procedure, examines the result and merges adjacent similar clusters into one.

9.9.2 Generation of straight lines joining centres of clusters

A subroutine, labelled SUBLINE, generates straight lines joining annotated centres of clusters. The collection of lines forms the picture which represents a fixation-point scan pattern. Lines ultimately consist of points and the coordinates of these points are stored to provide a graphic data base for the display subroutines.

When a Silent 700 or a VDU is used as a display unit, teletyped characters represent points in a two-dimensional plane, and here the greatest limitation is the poor resolution, and the fact that only 80 points can be displayed in the horizontal direction. If some of the characters are used to label y-coordinates of horizontal lines and others to provide margins on both sides of a display, then the display space available will be less than 70 points wide. Patterns which are more than 70 units wide can only be displayed after their size is reduced, a procedure to be avoided because of the deterioration in resolution. Due to the limited space available, and for convenience, it was decided that each character along the two orthogonal axes respresents a unity. Scaling the axes with fractions limits the maximum width which can be displayed.

There are two factors which have greatly influenced the design of an algorithm for interpolating straight lines, and these are:

1. Coordinates of displayed points should be whole numbers.
2. Mathematical operations employing fixed point arithmetic are not efficient in dealing with fractions due to the amount of rounding up entailed, when operations are based on magnitudes.

The algorithm is based on solving one of the two equations representing straight lines:

$y = mx + C$, where $m = \Delta y/\Delta x$, and C is a constant

$x = my + C$, where $m = \Delta x/\Delta y$, and C is a constant

depending upon whether $\Delta y > \Delta x$ or vice versa.

When $\Delta y \geq \Delta x$ solution is based on the first equation, rather than on the second for two reasons. First, the result of division is not

less than one, and most probably rounding up of the fraction introduces less error than when the result of the division is less than one. Second, when $\Delta y > \Delta x$, values of y-coordinates change at a greater rate than those of x-coordinates. As a consequence, if y is taken as the independent variable and is assigned integers, some values of x-coordinates might contain fractions and cannot be displayed unless the fractions are rounded up. If rounded up, they might not fall in a straight line. The same above argument is applicable when $\Delta x > \Delta y$.

Because of this approach, where m is an integer, the spacing between interpolated points varies as a function of m. As shown in Fig. 9.19, the spacing is minimum when a straight line lies on either the vertical or horizontal axis. In the vicinity of either axis the spacing is infinite and decreases as the slope approaches one.

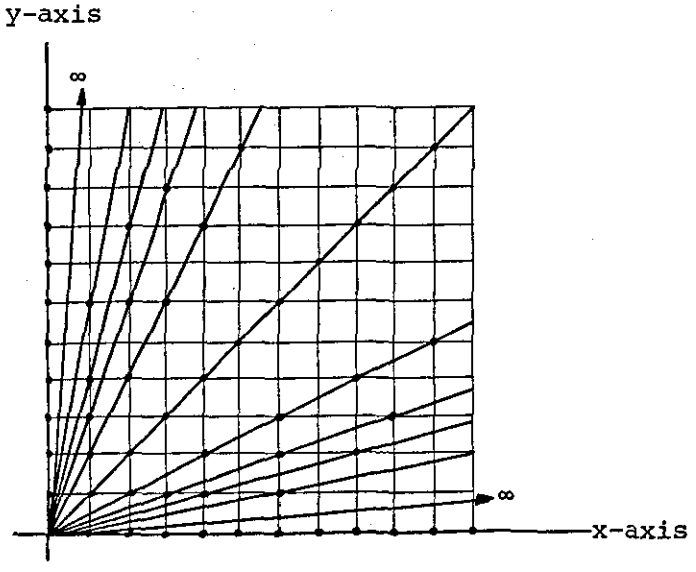


Fig. 9.19: Variation of Spacing Between Interpolated Points for m = integer

The algorithm,when solving the two equations, instead of determining C the point of intersection of the straight line joining two cluster centres with one of the coordinate axes, shifts the origin of the coordinate system to the first centre and calculates incremental values of the dependent variable using the slope of the line and incremental values of the independent variable. Points are generated by adding the incremental values to the coordinates of the first cluster centre. In mathematical terms, assuming that x_n, y_n and x_{n+1}, y_{n+1} are two cluster centres, the algorithm first computes:

$$\Delta y = y_{n+1} - y_n$$
$$\Delta x = x_{n+1} - x_n$$

and compares them to find which is greater. If $\Delta y \geq \Delta x$ it calculates:

$$m = \frac{\Delta y}{\Delta x}$$

and interpolates points x_i, y_i according to the relation:

$$x_i = x_n \pm \sum_{j=0,1,2\ldots}^{r} j \qquad \text{for: } x_i \leq x_{n+1}, (x_i \text{ increasing})$$
$$x_i \geq x_{n+1}, (x_i \text{ decreasing})$$

and $$y_i = y_n \pm \sum_{j=0,1,2\ldots}^{r} jm$$

If $x > y$ interpolation is based on

$$y_i = y_n \pm \sum_{j=0,1,2\ldots}^{r} j \qquad \text{for: } y_i \leq y_{n+1}, (y_i \text{ increasing})$$
$$y_i \geq y_{n+1}, (y_i \text{ decreasing})$$

and $$x_i = x_n \pm \sum_{j=0,1,2\ldots}^{r} jm$$

Values of the independent variables (x_i, y_i) can be increasing or decreasing, and the slope m assumes positive or negative values depending on the position of the two cluster centres relative to each other. By shifting the origin to the first point (x_n, y_n) the second point (x_{n+1}, y_{n+1}) lies either in one of the eight sectors of the circle shown in Fig. 9.20, or along one of the two coordinate axes. Each sector subtends an angle of 45° at the origin and on each values of Δx relative to Δy is shown together with the behaviour of coordinates of points to be interpolated. At the onset of interpolation, the program determines whether the two centres lie along one of the axes. If, for instance, both points are on the x axis i.e. $y_n = y_{n+1}$, then starting at x_n, y_n, the program interpolates points by incrementing or decrementing x_n, and assigns to each point the same value of the y-coordinate (y_n). If the two centres are not on one of the axes, the program finds out in which sector x_{n+1}, y_{n+1} lie, calculates m and solves one of the equations shown above. A flowchart of SUBLINE is shown in Fig. 9.21 through Fig. 9.23.

9.9.3 Adjustment of centres of clusters

Due to the rounding up of fractions when slopes of straight lines connecting centres of clusters are calculated, the straight lines will not intersect at the centres, unless all original slopes are integers. For example, referring to Fig. 9.24, suppose that $\Delta y > \Delta x$ and the division $\Delta y/\Delta x$ yields an integer plus a fraction. If the fraction is ignored, then interpolated points will lie along c_1, c_2' which has a lower slope than c_1c_2. This will result in a discontinuous display,

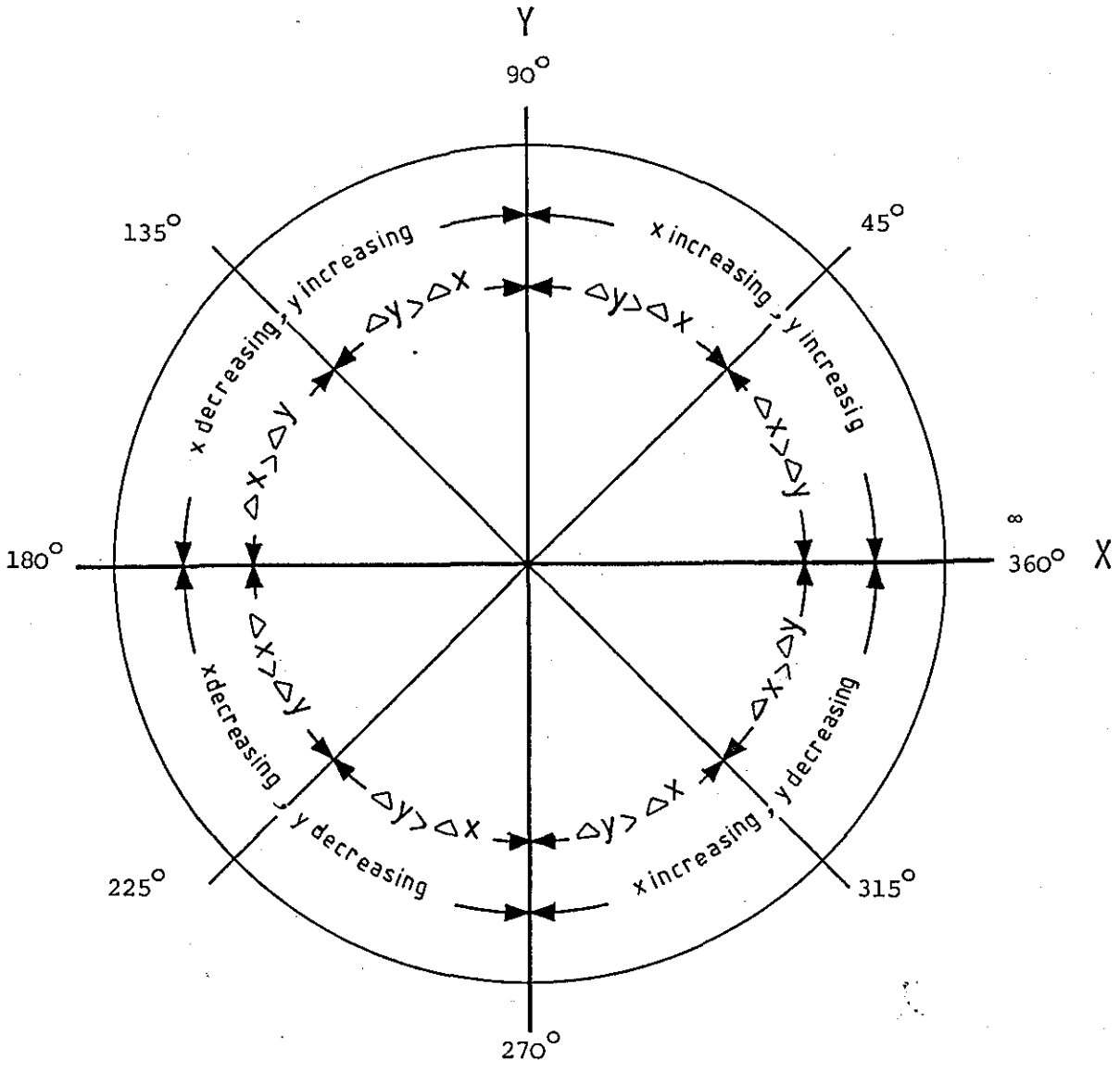


Fig. 9.20: Relationship Between x_n, y_n and x_{n+1}, y_{n+1}

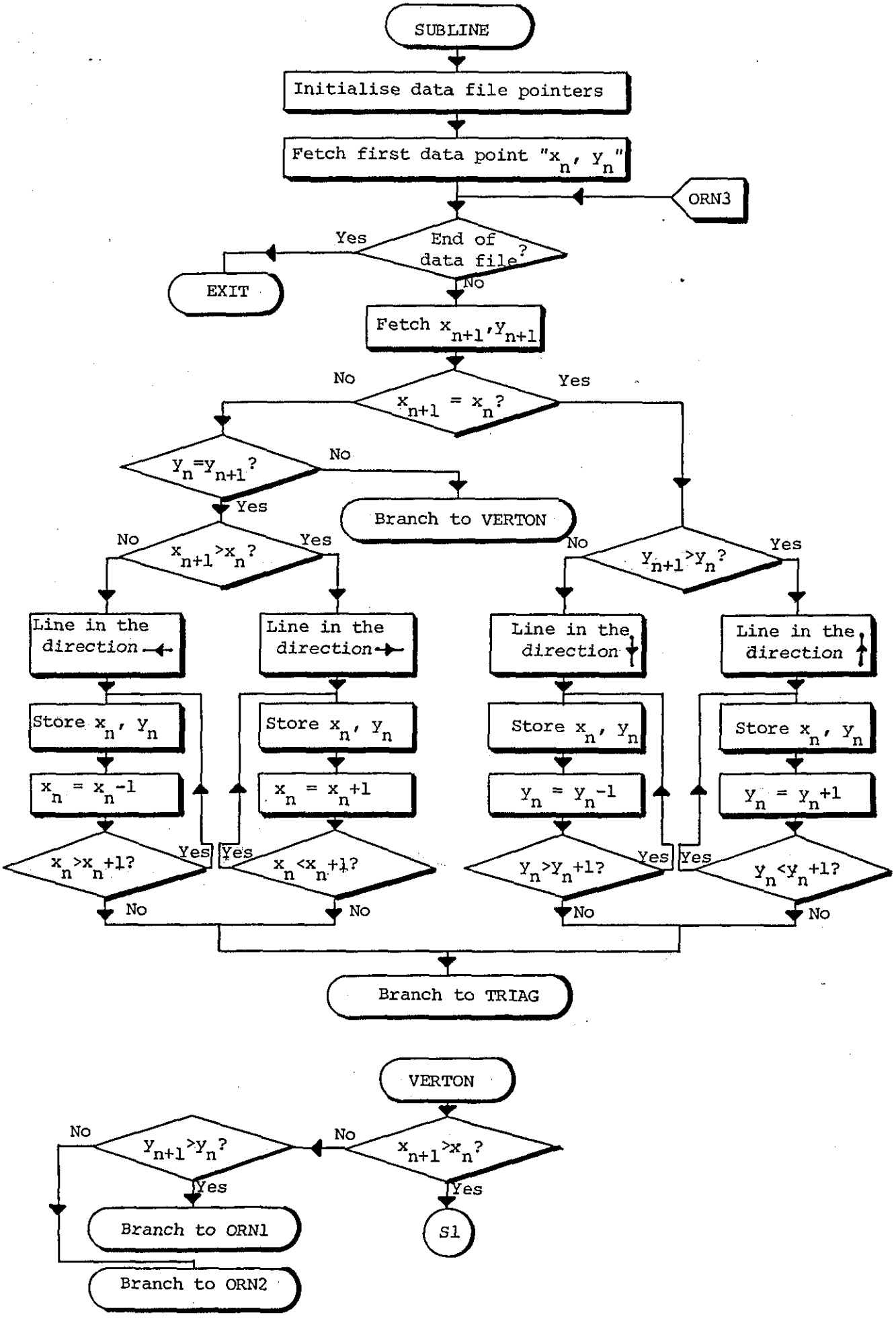


Fig. 9.21: Flowchart of SUBLINE

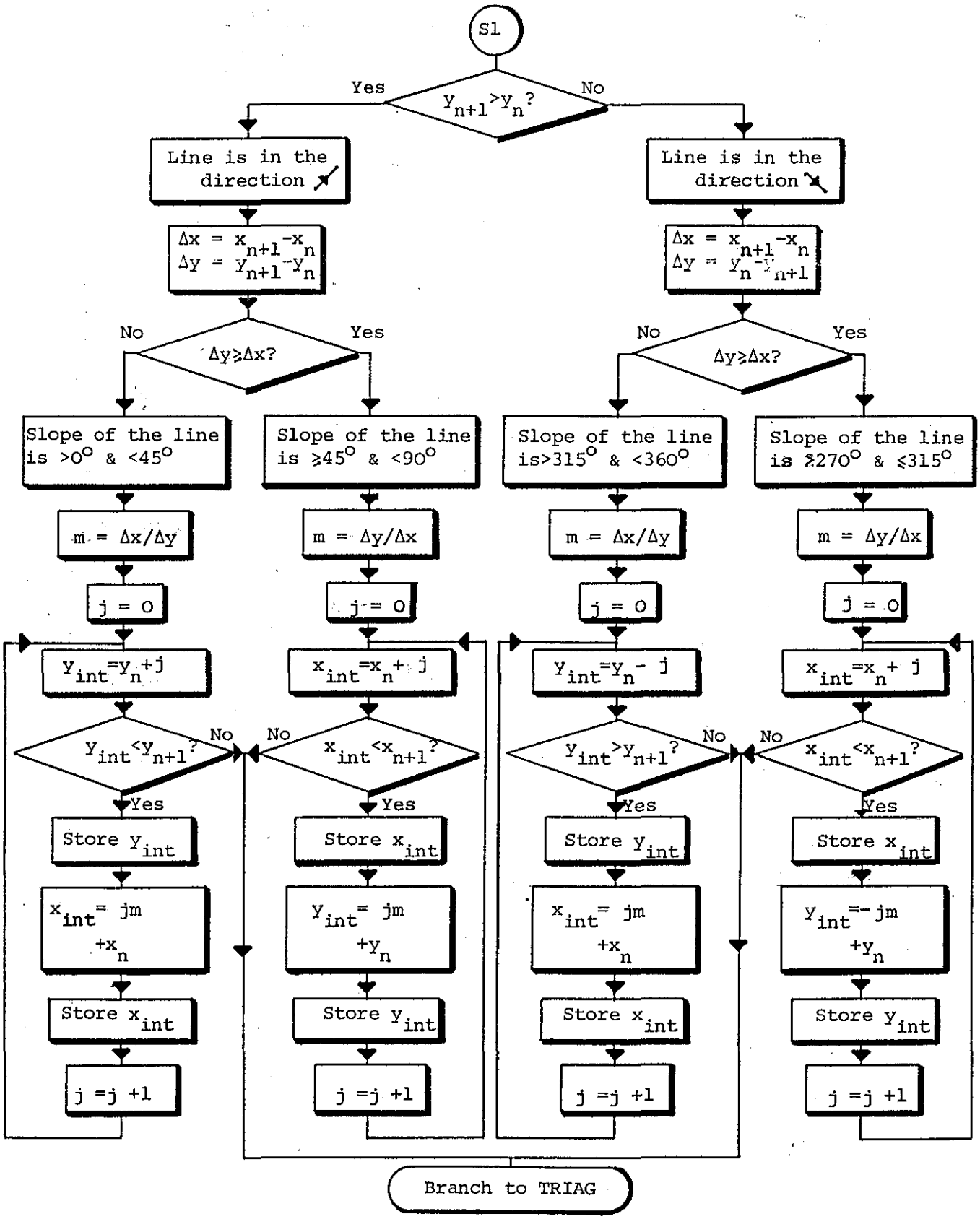


Fig. 9.22: Flowchart of SUBLINE (continued)

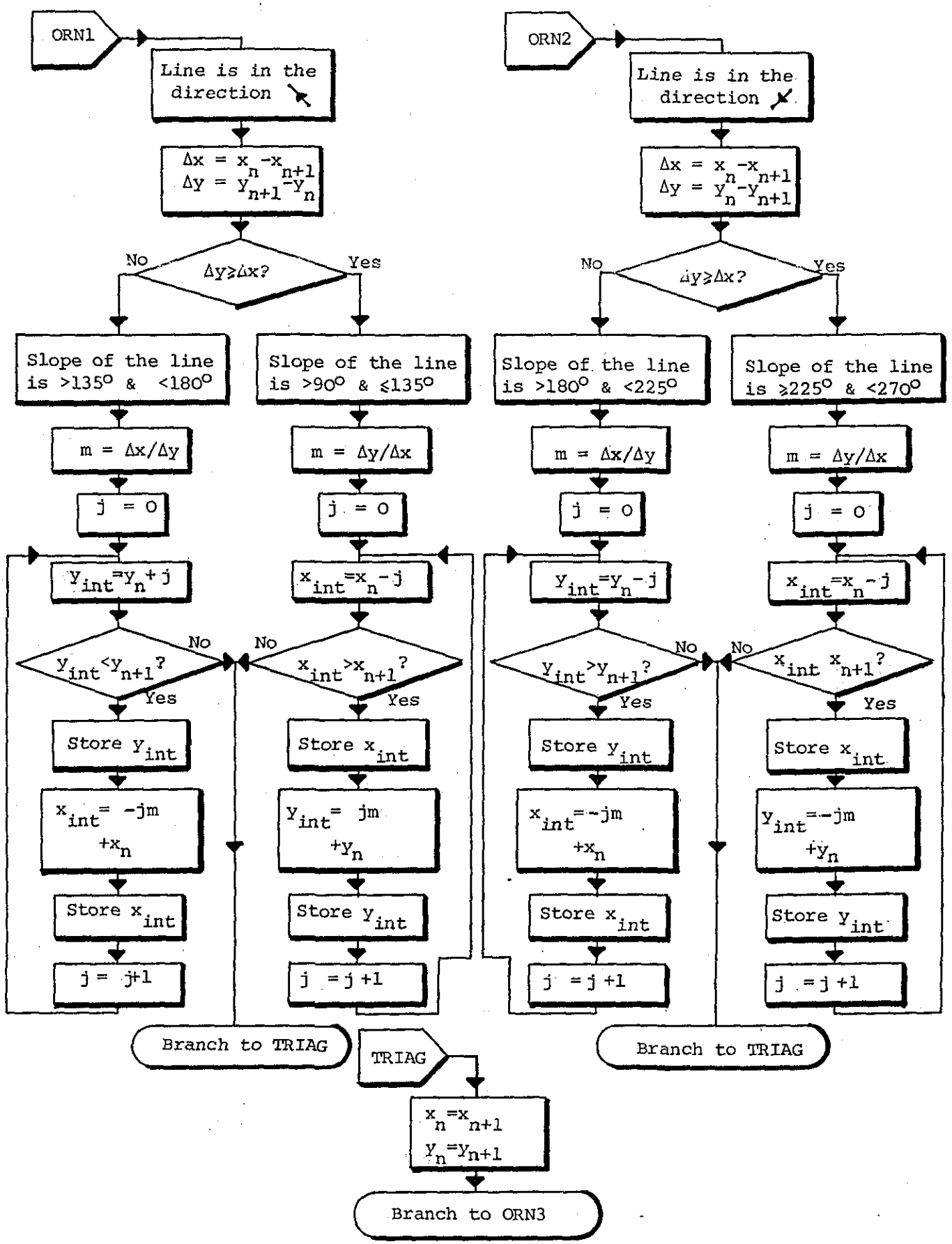


Fig. 9.23: Flowchart of SUBLINE (continued)

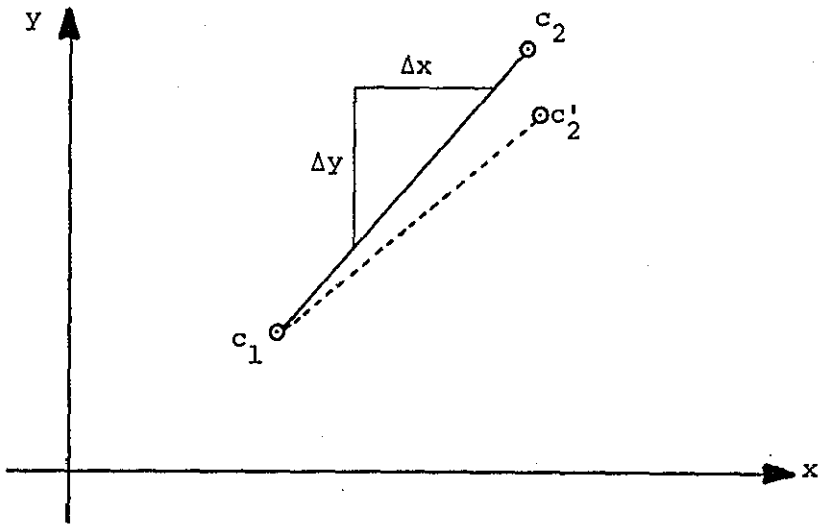


Fig. 9.24: *Effect of Rounding up* Δy/Δx

with straight lines not passing through their original end points.

To overcome this, a subroutine called MODCEN, examines the centres of clusters before running SUBLINE. If slopes of lines joining them are not integers, it replaces the centres by new ones, in their vicinity, such that slopes of all lines connecting the new centres are integers. A flowchart of MODCEN is given in Fig. 9.25 through 9.27.

The procedure is illustrated in Fig. 9.28, where c_1, c_2, c_3 and c_4 represent original cluster centres. Assume that the slope of c_1c_2 is not an integer. MODCEN rounds up the fraction and uses the modified slope to determine a new point c_2' near c_2. Then it computes the slope of $c_2'c_3$. If it is an integer it leaves c_3 as it is, otherwise it replaces it by c_3'. Similarly, c_4 may be replaced by c_4'. In case the display forms a closed loop, a different approach is adopted for $c_4'c_1$, because replacing c_1 by, say c_1' leads to the formation of a

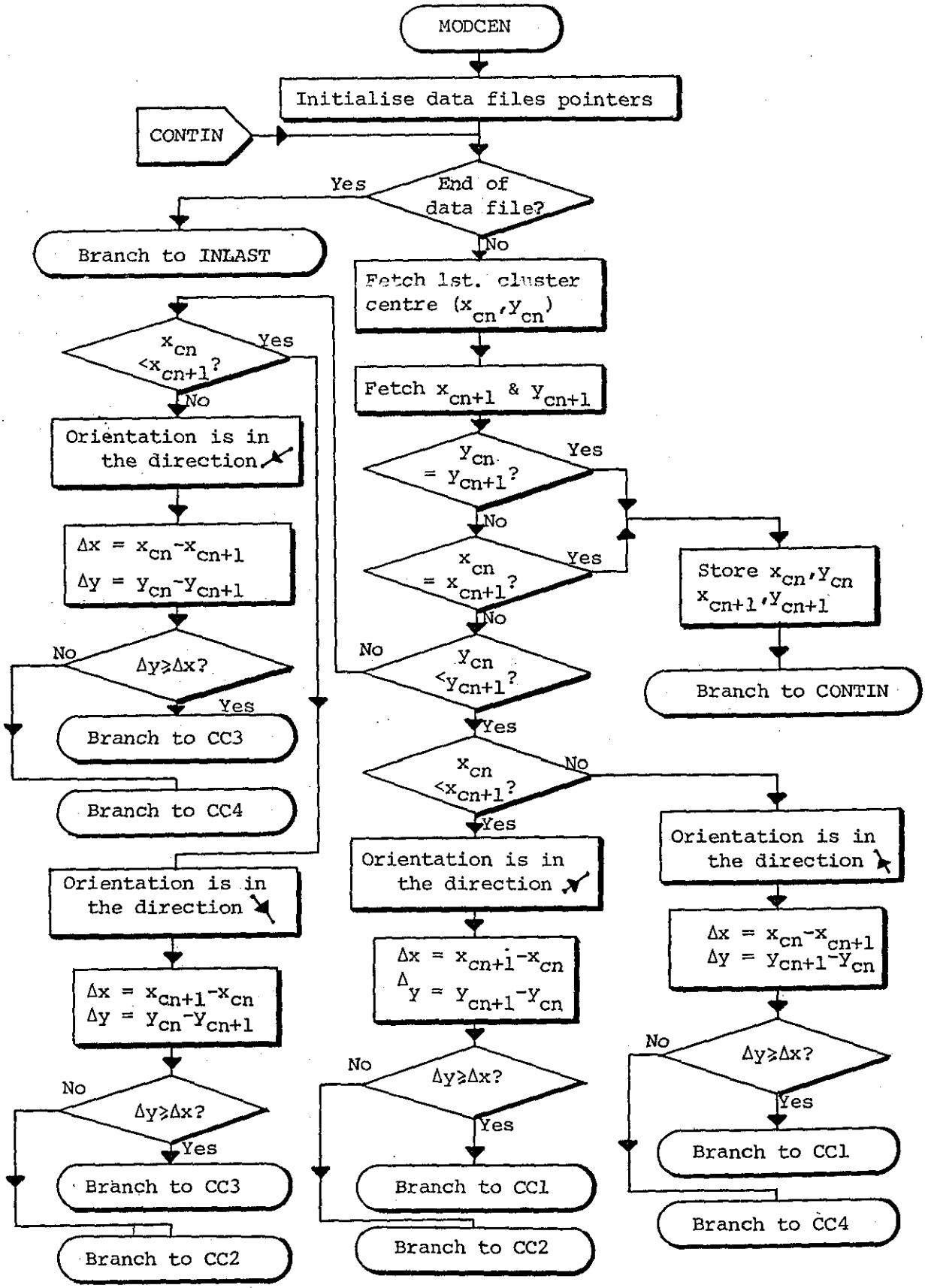


Fig. 9.25: Flowchart of MODCEN

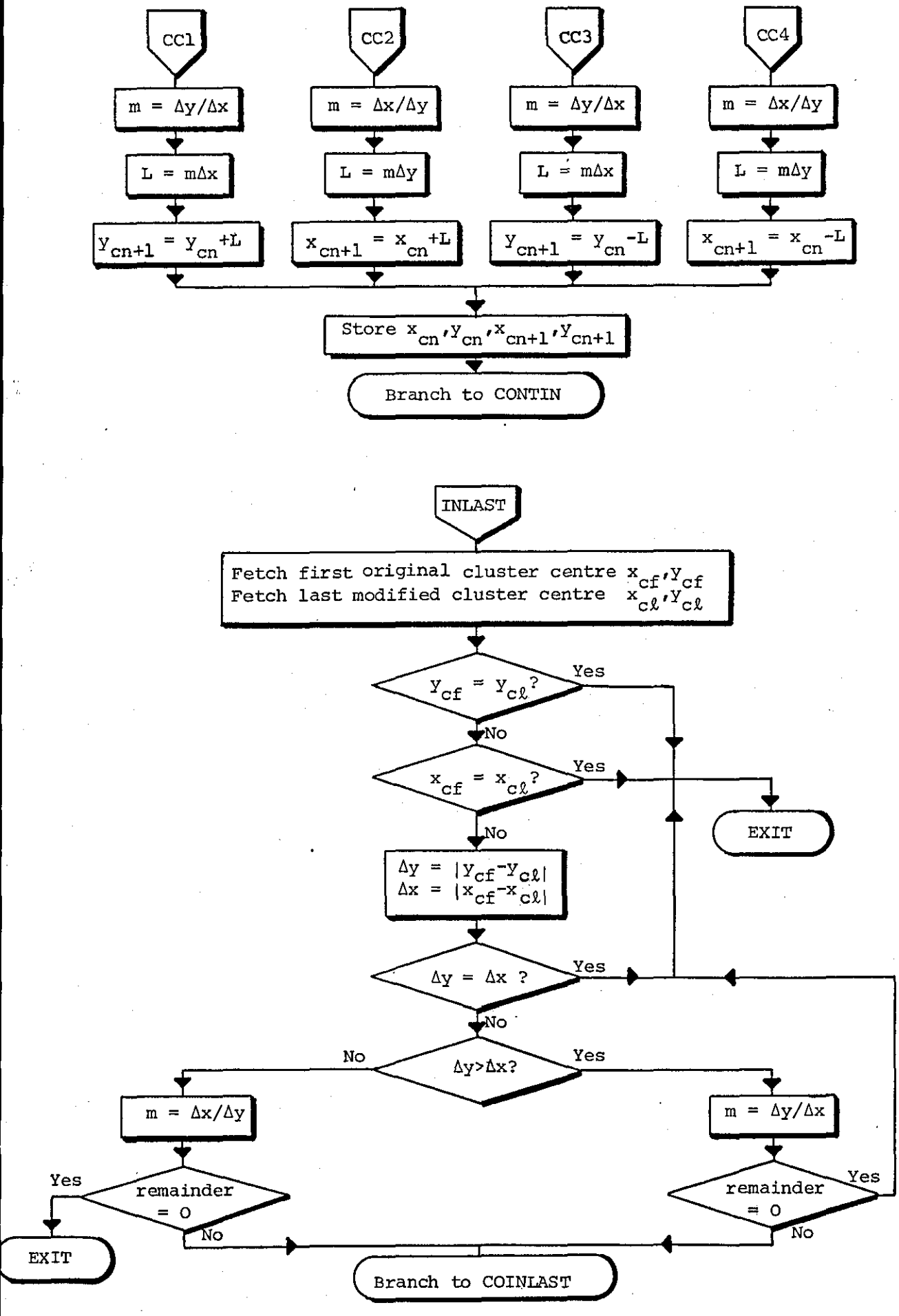


Fig. 9.26: Flowchart of MODCEN (continued)

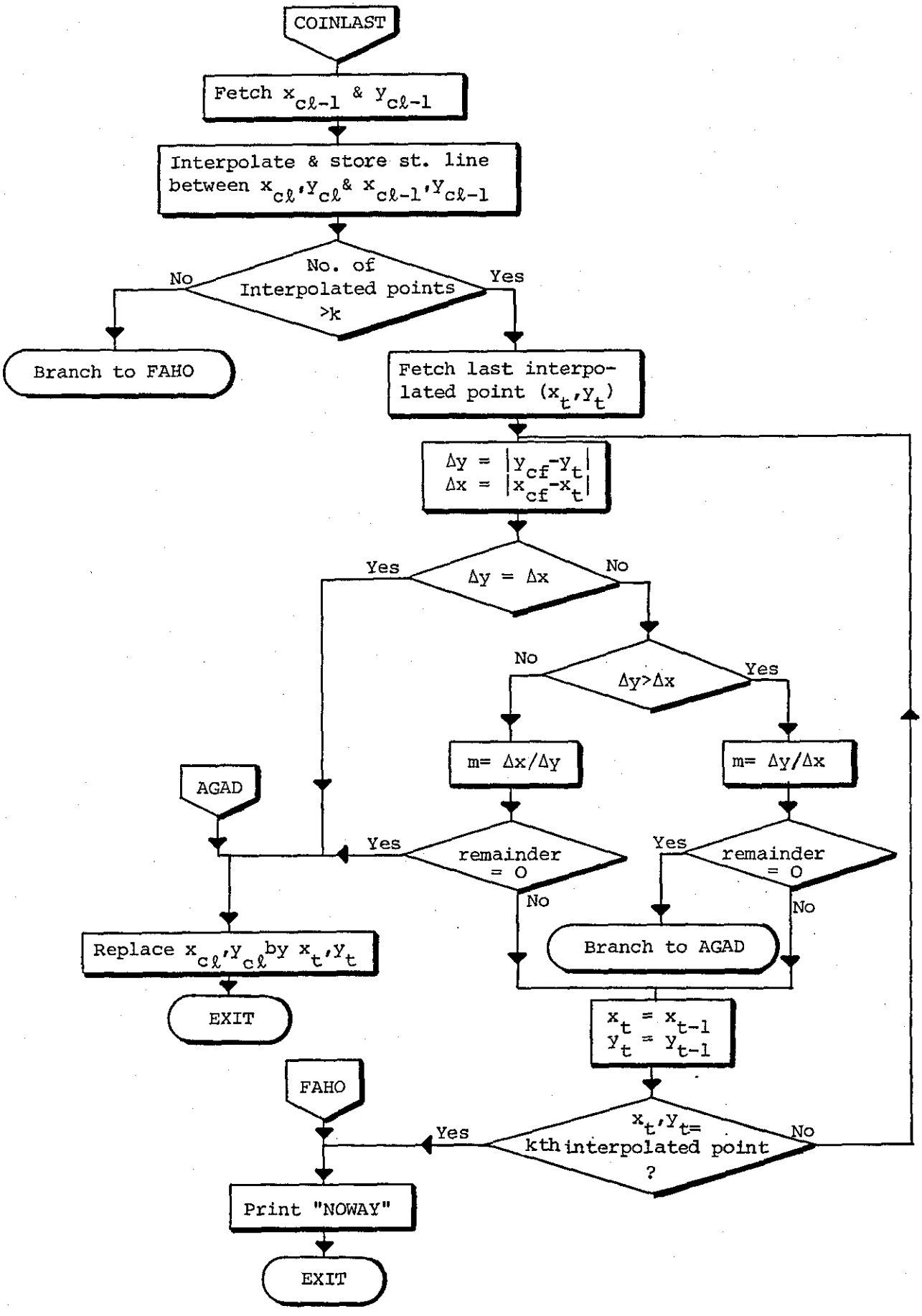


Fig. 9.27: Flowchart of MODCEN (continued)

new line $c_1'c_2'$ whose slope might not be an integer, and the procedure goes in cycles. The new approach is based on interpolating points lying on $c_3'c_4'$, and then starting at c_4' the program successively tests the interpolated points until it locates a point c_4'' in the vicinity of c_4', such that the slope of $c_4''c_1$ is an integer. If such a point is not available, the program passes a message to the operator, who can decide whether to continue modifying c_4' until the program finds a suitable point or leaves the display as an open loop. The example discussed above represents the worst case, i.e. when all cluster centres, except the first one, are modified.

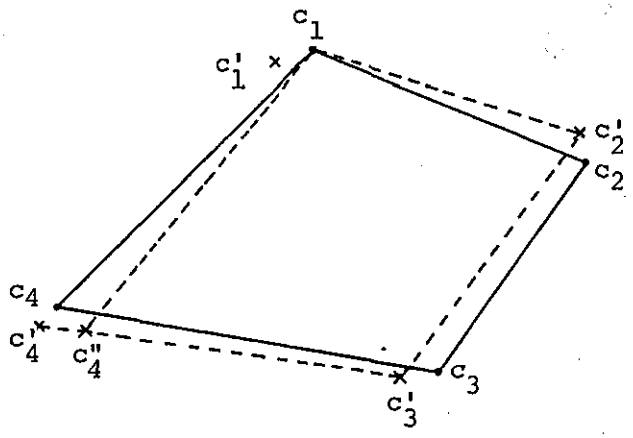


Fig. 9.28: Adjustment of Cluster Centres

9.9.4 Data manipulation for hardcopy display

After interpolation of straight lines, data is manipulated in a format amenable for display on a Silent 700. This is achieved by arranging interpolated points in a string such that their y-coordinates

are in ascending order, and simultaneously all points with identical y-coordinates are set such that their x-coordinates are in ascending order. A flowchart of a program performing this function, called DATMAN is shown in Fig. 9.29.

9.9.5 Scaling of edge point data

A subroutine labelled SCACO is used to multiply (or divide) the coordinates of points representing a pattern by a constant (2^n) to increase (or decrease) its size. When a pattern is small it can be enlarged to cover a great part of the space allocated for display. Similarly, when it is large it can be diminished to an appropriate size.

Enlarging a pattern is particularly useful when the storage oscilloscope is used as the display unit. The scope is not of good quality and the display tends to blur when the two V/Div switches are set to low values. The pattern can be enlarged while these switches are set to higher values to offset the increase in the size, resulting in a clear display with relatively sharply defined lines. Also, increasing the size of a pattern by, say 8, increases the number of interpolated points by approximately the same order. In the limit interpolated points merge into a continuum.

9.9.6 Hardcopy display of results

A subroutine, labelled HARCO, is used to obtain a hardcopy display of annotated fixation-point scan patterns. A flowchart of HARCO is given in Fig. 9.30 through 9.32. HARCO is more complicated than the similar subroutine HARDED (Section 6.4) because of the requirement for printing the annotations. The flowchart shows the

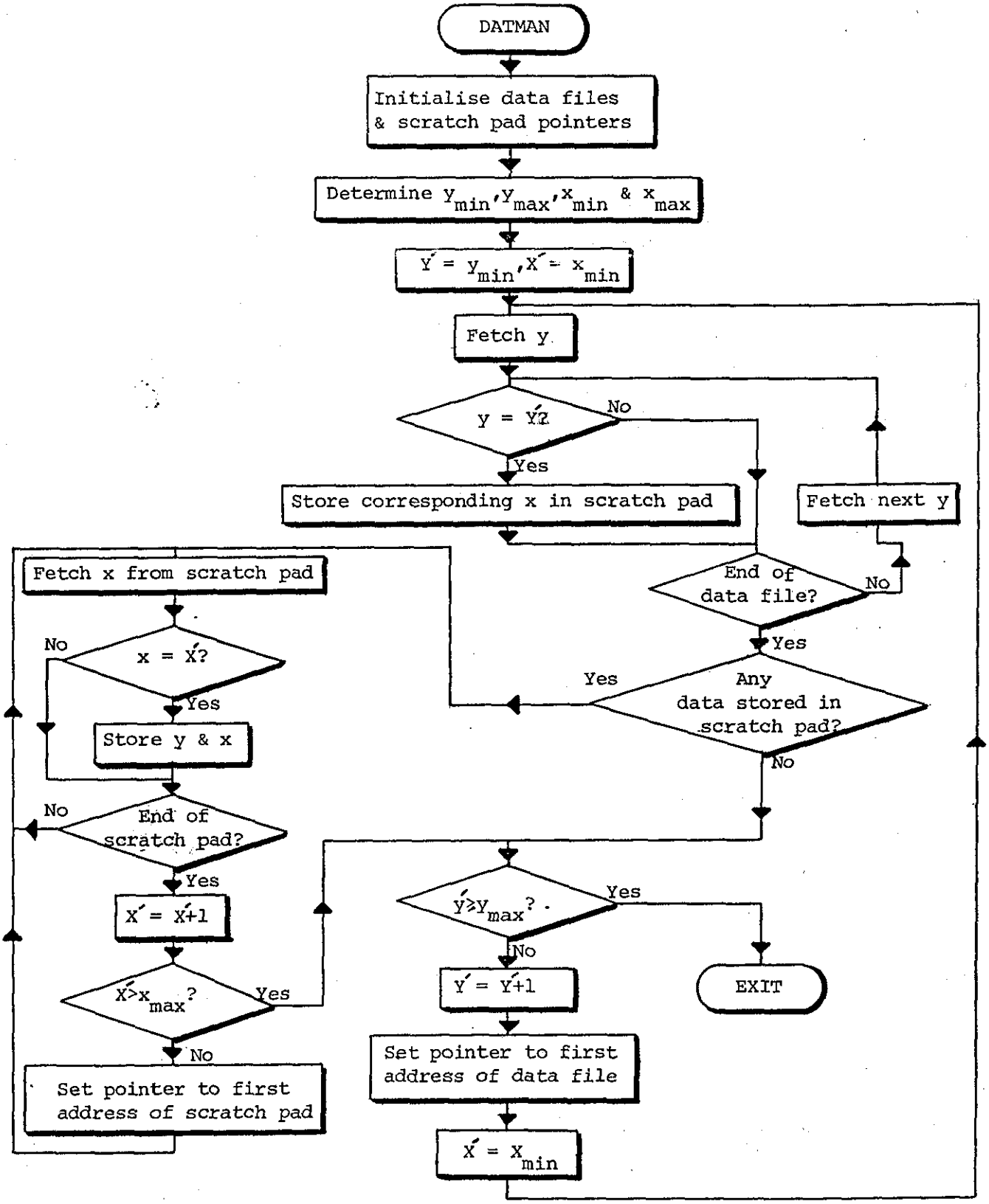


Fig. 9.29: Flowchart of DATMAN

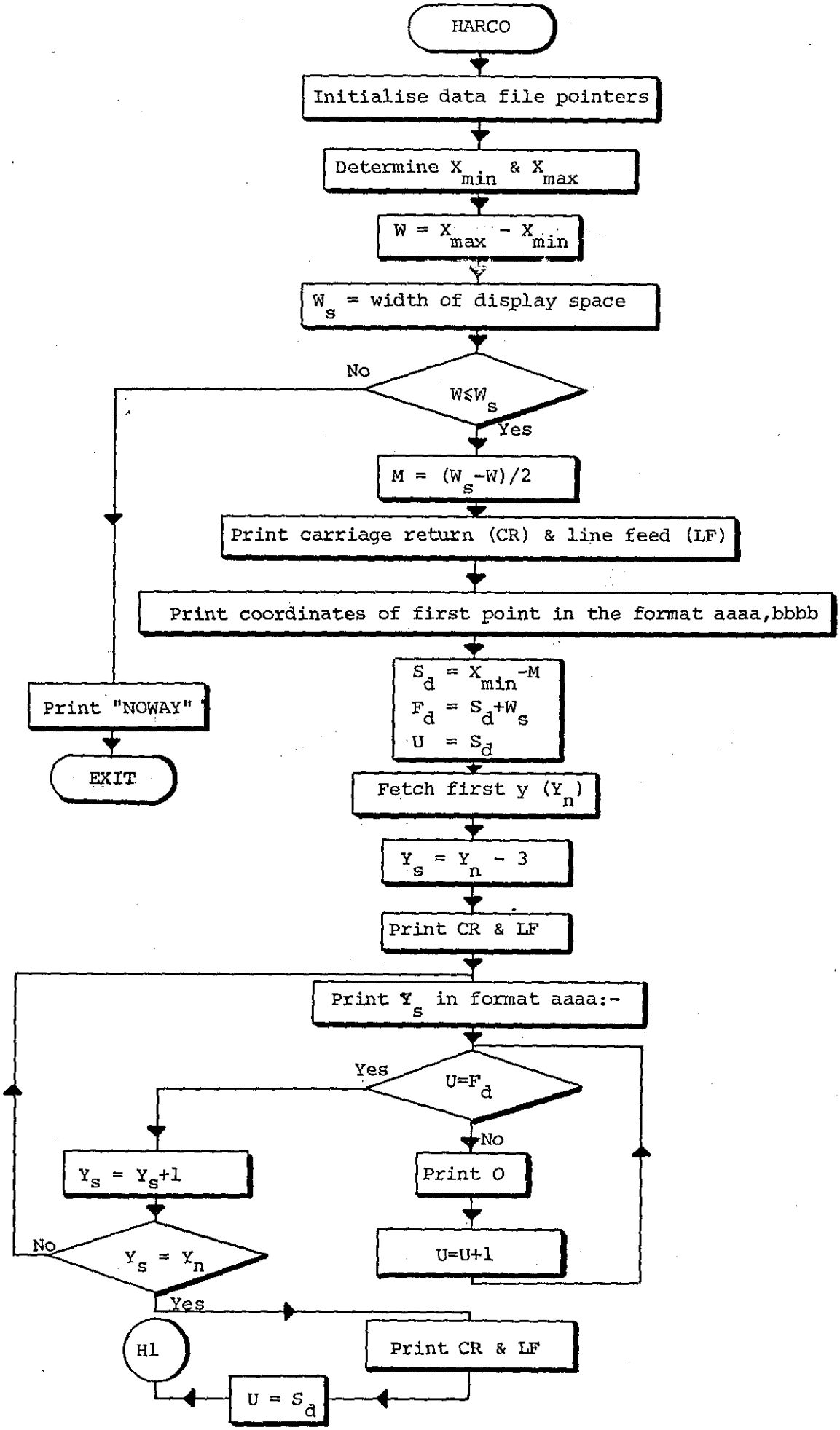


Fig.9.30: Flowchart of HARCO

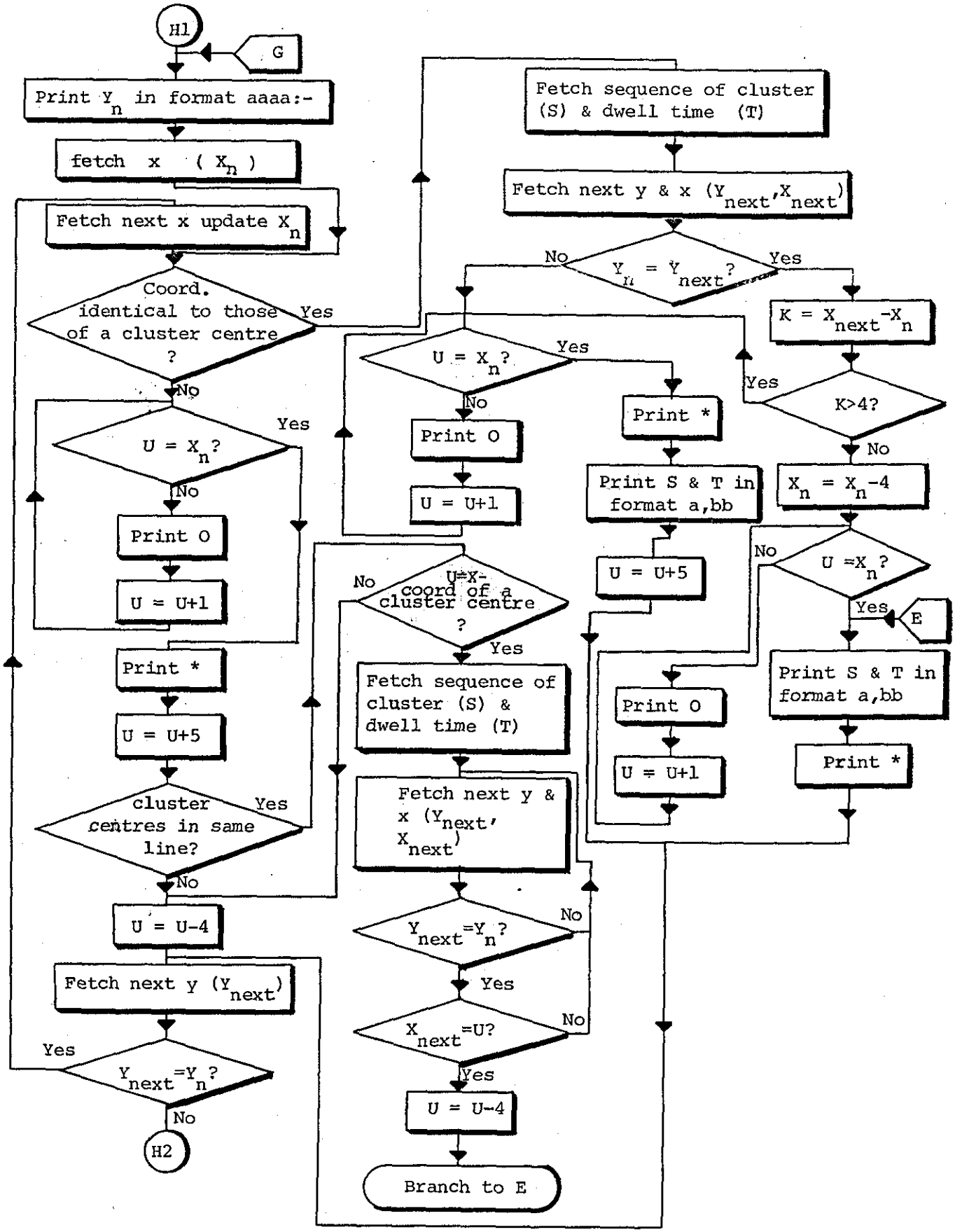


Fig. 9.31: Flowchart of HARCO (continued)

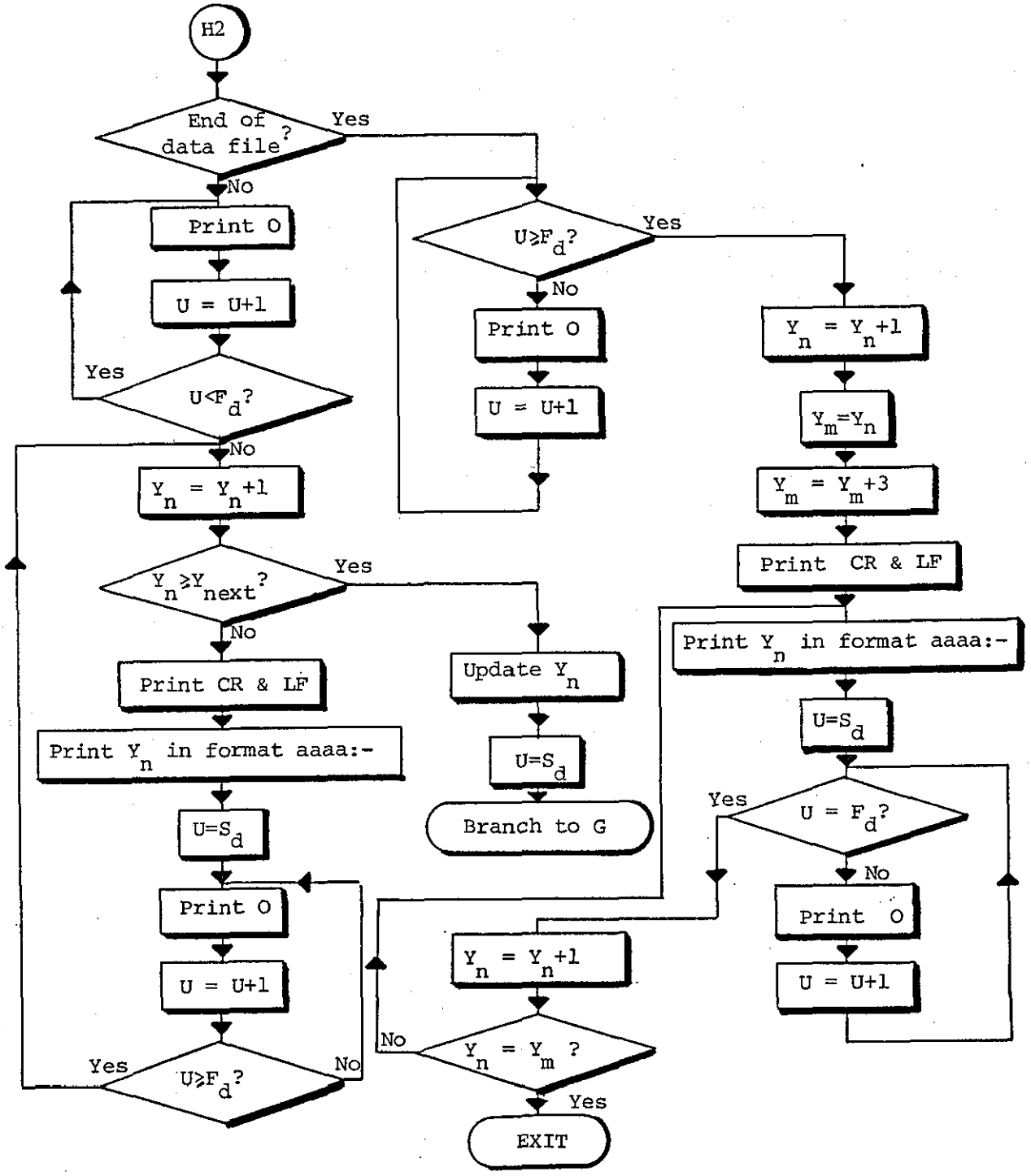


Fig. 9.32: Flowchart of HARCO (continued)

counters which are used to achieve this. Fig. 9.33 is an example of the output of HARCO.

Before any point belonging to a pattern is displayed, the program tests it to find whether it is a cluster centre or not. If so, the next point is fetched and its distance from the cluster centre is measured. In case it lies very near to the centre, annotations are printed on the left-hand side, otherwise they are printed on the right-hand side. If 3 or more cluster centres lie on the same horizontal line, annotations are printed immediately to the left of the centres in the middle part of the horizontal line.

9.10 Operator-Program Interaction

Programs for data processing and display of results are interactive to make it easy for an operator to obtain an optimal solution to the problem of data compression, and to facilitate the attainment of a suitable display. Here, interaction entails answering queries of the programs concerning the starting and final addresses of input data files, starting addresses of processed data files and values assigned to parameters. During the last stage of their execution, the programs print messages to show locations of processed data files. When subroutine HARCO is unable to display a pattern because its width is greater than that of the allocated display space, and when MODCEN fails to accomplish closed loop interpolation; messages are sent to the operator to make him aware of the situation.
To solve the first problem the operator has to reduce the size of the pattern using SCACO. Concerning the second problem, adjustments can be made to the last cluster centre, or the last stage of the closed

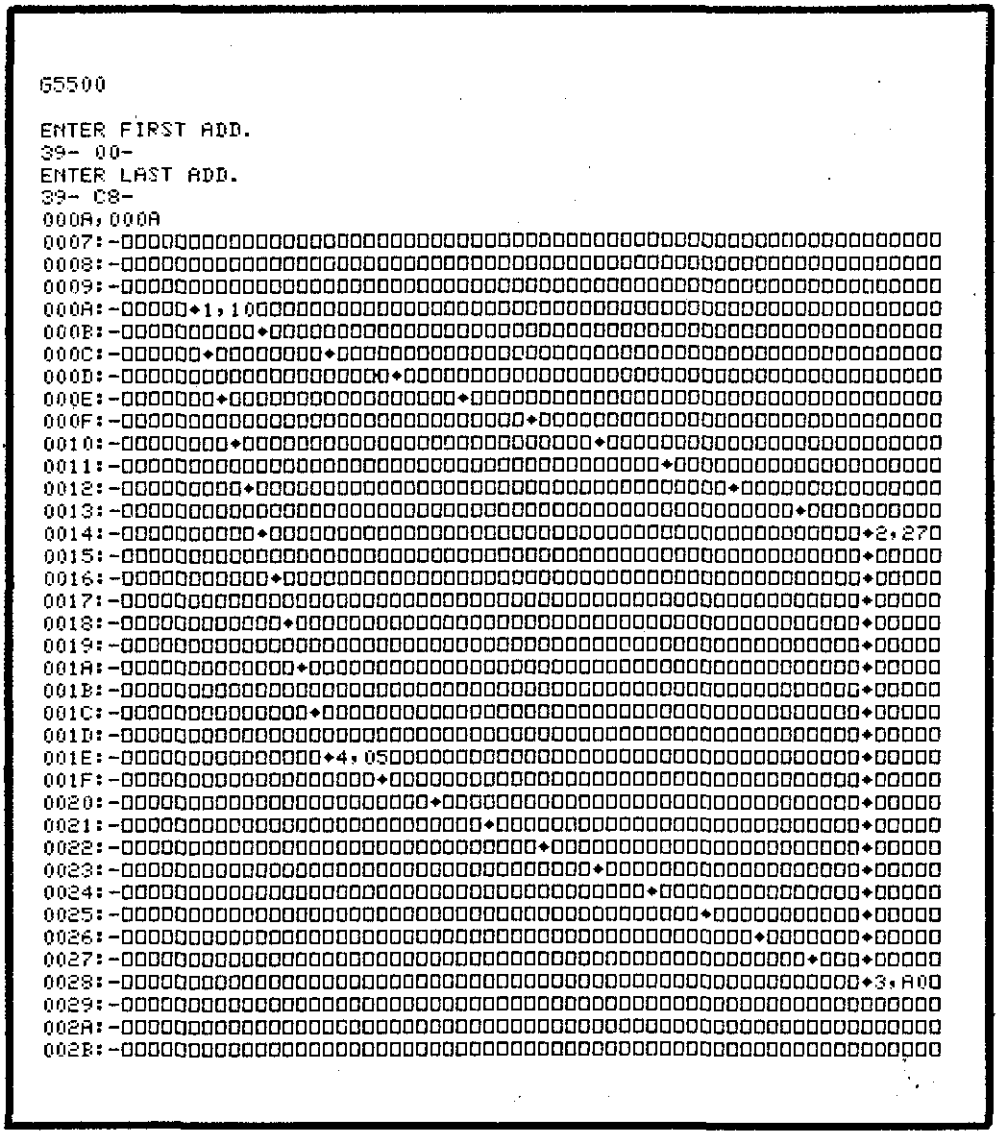


Fig. 9.33: An Example of a Hardcopy Display

loop may be omitted altogether (Section 9.9.3). However, closed loop interpolation is applicable when a number of fixation points are not collinear and at the same time the last point is identical to the first i.e. both x- and y-coordinates are equal, a situation expected to occur in very few cases. A display showing a small discrepancy between these two points will not affect interpretation of results.

The order at which subroutines may be executed is given in Fig. 9.34(a) and Fig. 9.34(b). Format of different data files is presented in Fig. 9.35. Here, subscripts affixed to x and y are self-explanatory. MODCEN (not shown in the figure) retrieves x- and y-coordinates from the statistical analysis data file, processes and stores them in the same memory locations.

Following collection of point-of-regard data, the first subroutine executed is DATRED. Besides data reduction, it can be used to get rid of noise or spurious points when they exist. If a scan pattern of unprocessed fixation points is desirable, DATRED can be used to manipulate the coordinates of raw data in the format amenable for analysis by SUBLINE. This is achieved by setting R=0 and T=0 to ensure that no fixation point is deleted, and successive identical points are combined together.

The second subroutine which may be executed is MODCEN, followed by SUBLINE and SCACO. If a hardcopy of the display is desirable DATMAN is run after SUBLINE. VIDUD and HARCO output coordinates of scan patterns to the display peripherals. SCACO is also used to multiply coordinates of raw or processed fixation points by a constant prior to execution of SUBLINE, to increase the number of interpolated

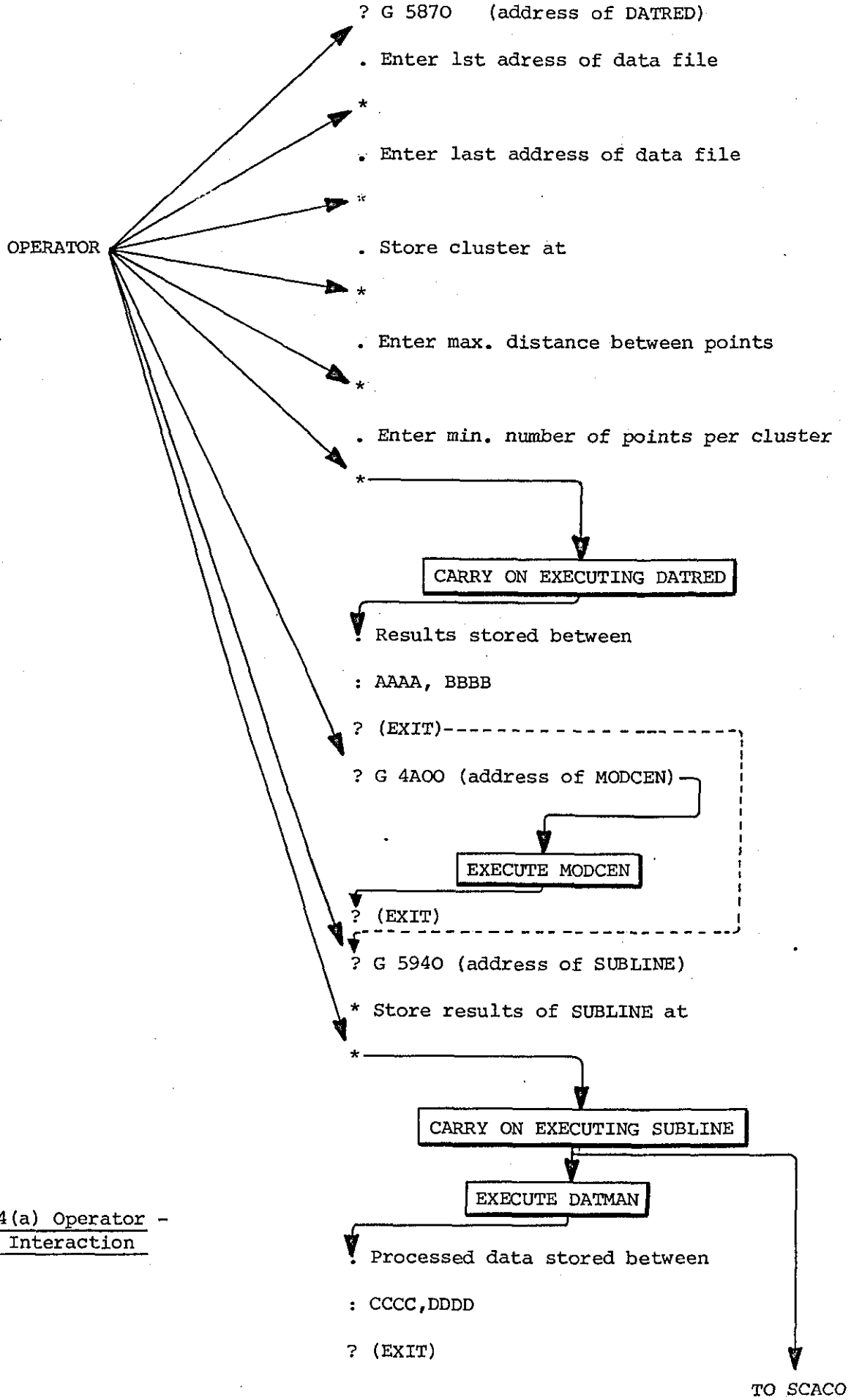


Fig. 9.34(a) Operator - Program Interaction

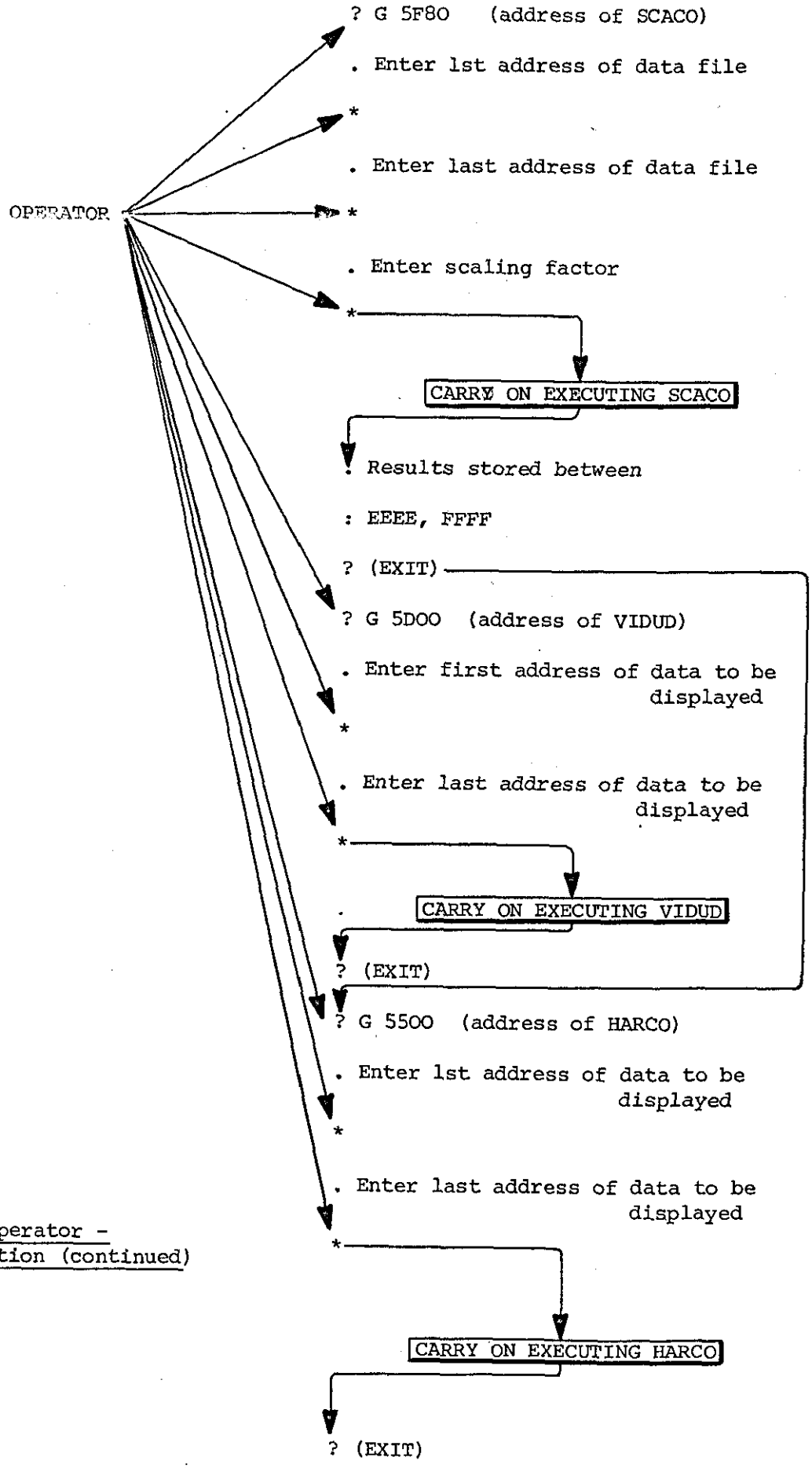


Fig. 9.34(b) Operator - Program Interaction (continued)

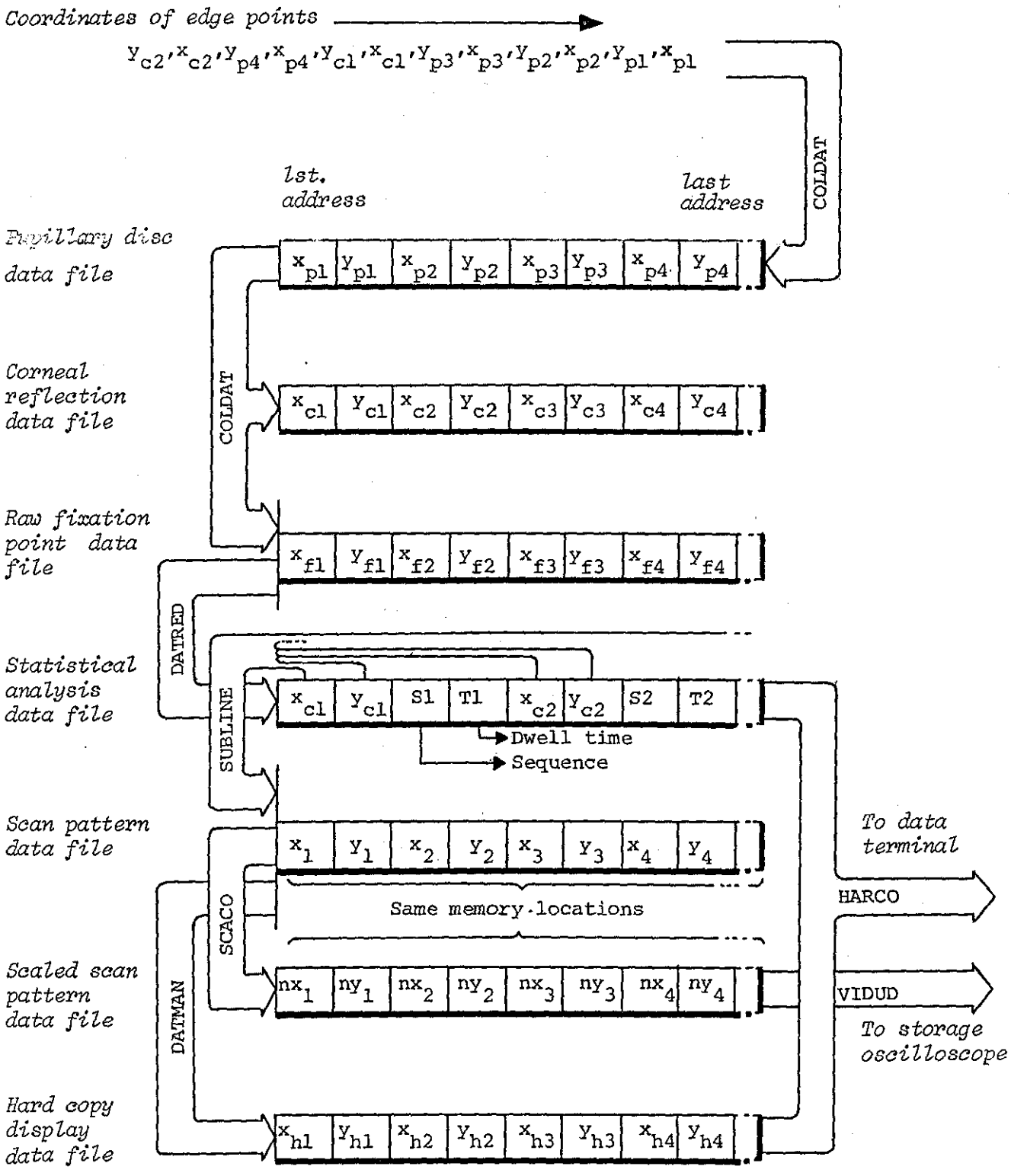


Fig. 9.35: Format of Data Files

points, (Section 9.9.5). It is advisable to obtain a display with and without running MODCEN to see whether original end points of interpolated lines have been adjusted or not, and if they are, to what extent the original scan pattern is altered.

It would have been more convenient to design the interaction such that all processing and display subroutines are joined together, and the operator need not be aware of locations of data files. Moreover, setting of scaling factors can be performed automatically by software and interaction confined to answering queries of the program about the degree of data reduction desirable, display unit to be used, whether to execute MODCEN or not and information relayed to the operator about raw and processed data. Implementation of such procedures is not feasible now because of the limited store available.

9.11 Results

This section contains photographs of results displayed on the storage oscilloscope. The photographs were taken with Polaroid 7.4 x 9.4 cm films. It should be stressed that no set of still photographs can convey the expedient manner by which information is displayed on the oscilloscope, where an operator can observe point-of-regard plotted in real time, and records of raw and processed eye track traced by a moving spot in accordance with the sequence of fixations.

The storage oscilloscope used has a rather large spot size, low persistence and the intensity of the spot is not the same everywhere in the viewing area. Moreover, the trace tends to smear and lines in

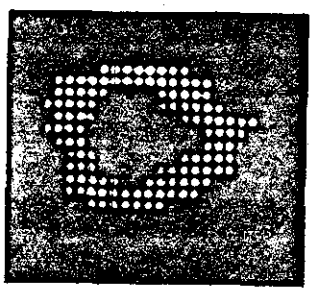
close proximity merge together. For these reasons, sometimes the display is not of good quality.

The first set of results is based on simulation data, which is used to test different subroutines. This is followed by analysis of eye movements of subjects fixating their eyes on stationary objects.

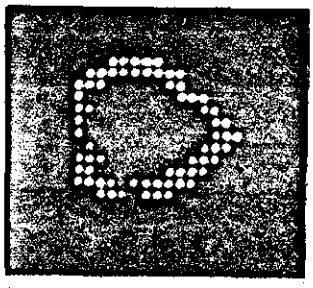
9.11.1 Simulation data

The sequence of photographs in Fig. 9.36 illustrates the ability of the clustering algorithm to work on a set of sequential data which does not fall into separate natural clusters. The simulation data, consisting of 1500 samples, was generated by tracking a circle with a spot inside it rotating in an approximately circular path, and evaluating the position of the spot relative to the centre of the circle. Here,the strategy is to run the clustering algorithm while setting the cluster radius to a small value (R=1), and eliminating clusters with small numbers of data samples through the adjustment of parameter T to proper values, Such simulation data (with no separate cluster domains) is somewhat similar to eye movement records due to discrete stops and small involuntary saccades, observed when a subject tries to follow lines of geometrical figures with his eyes smoothly.

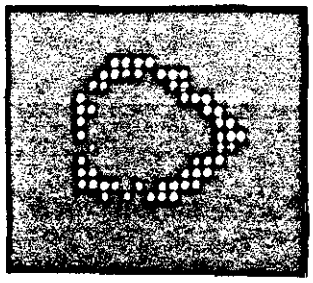
The ability of the clustering algorithm to deal with data, which forms separate cluster domains is shown in Fig. 9.37. Simulation data is generated in a manner similar to the one explained above, i.e. tracking a moving circle with a spot inside it. The four clusters represent position of the spot relative to the centre of the circle for four different parts of the circular motion (Fig. 9.37(a)).



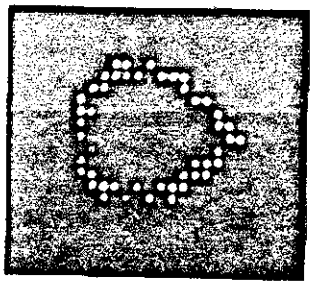
(a) Raw Simulation Data



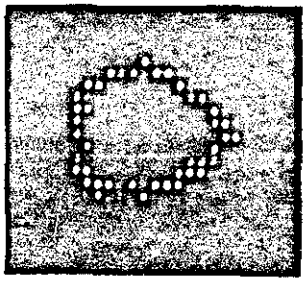
(b) Processed Data
(R=1, T=3)



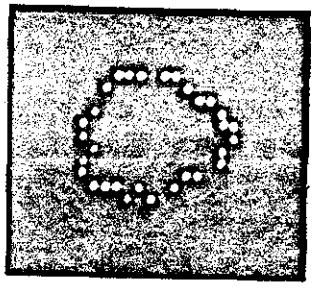
(c) Processed Data
(R=1, T=4)



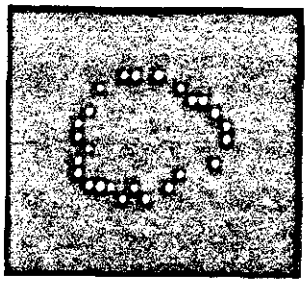
(d) Processed Data
(R=1, T=5)



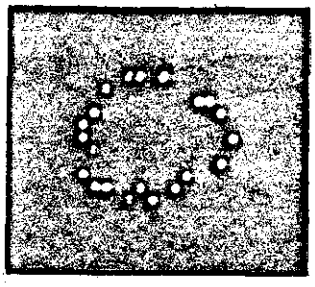
(e) Processed Data
(R=1, T=6)



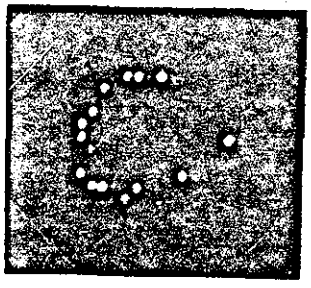
(f) Processed Data
(R=1, T=7)



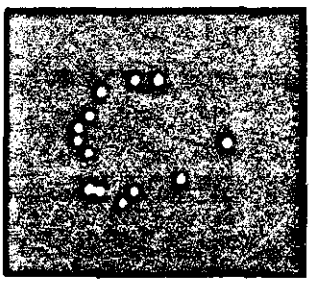
(g) Processed Data
(R=1, T=8)



(h) Processed Data
(R=1, T=9)



(i) Processed Data
(R=1,T=10)



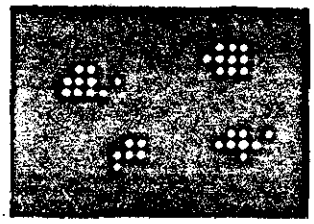
(j) Processed Data
(R=1, T=11)

Fig. 9.36: Data Reduction of a Continuous Set of Samples

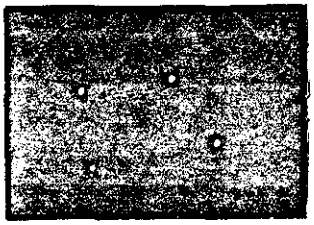
The clustering algorithm identifies the four clusters, determines their centres, sequence of occurrence and total number of data samples within each cluster domain. The centres of clusters are shown in Fig. 9.37(b).

The sequence of Plates (c) through (l) illustrates options for displaying centres of clusters connected by straight lines. Subroutines involved are SCACO, SUBLINE and MODCEN. As explained in Section 9.9.5, to obtain a continuous straight line, the number of interpolated points should be increased till they form a continuum of points which appear to the eye as a homogeneous straight line. At first raw data is scaled by multiplying its coordinates with 2^4 and the oscilloscope is adjusted such that the display covers a suitable area of the screen. Then cluster centres of original data are determined and their coordinates are scaled by 2^1, 2^2, 2^3 or 2^4 before running SUBLINE, the bigger the scaling factor the more interpolated points obtained. Coordinates of interpolated points are scaled to bring a pattern to its proper size; for instance, if the scaling factor 2^1 is employed for cluster centres, then the output of SUBLINE must be scaled by 2^3. Similarly, if centres are scaled by 2^2, 2^3 or 2^4, then resulting patterns are scaled by 2^2, 2^1 or 2^0 before displaying them. The plates illustrate the effect of increasing the scaling factor. The best display is obtained when the scaling factor is set to 2^4 (Plates (k) and (l))

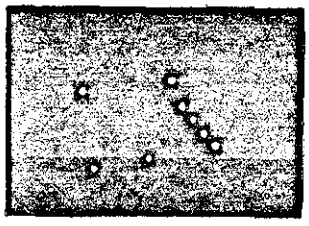
This approach is adopted rather than assigning a fixed value to the scaling factor because the storage space allocated for the graphic data base is about 1.5K words, which is too small for complex scan



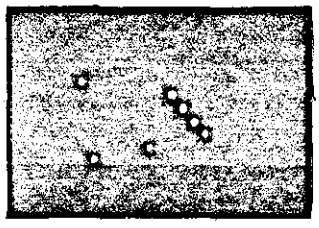
(a) Raw Simulation Data



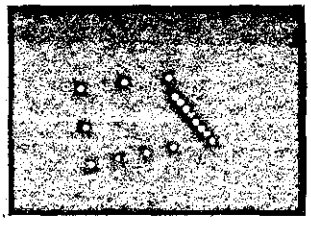
(b) Centres of Clusters



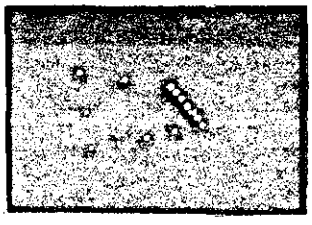
(c) Original Centres, scaling factor = 1



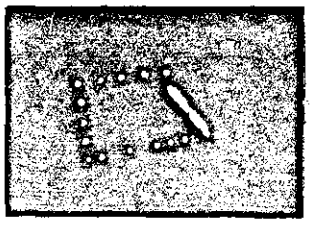
(d) Centres modified, scaling factor = 1



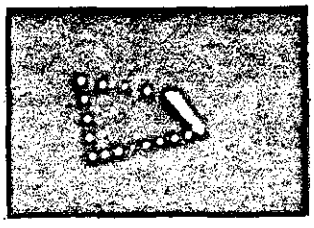
(e) Original Centres, scaling factor = 2



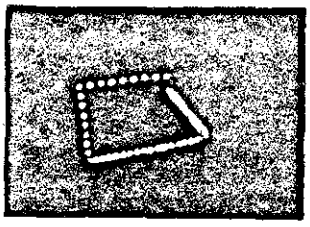
(f) Centres modified, scaling factor = 2



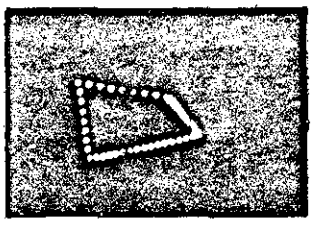
(g) Original Centres, scaling factor = 4



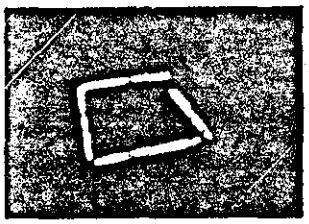
(h) Centres modified, scaling factor = 4



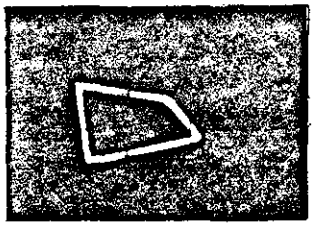
(i) Original Centres, scaling factor = 8



(j) Centres modified, scaling factor = 8



(k) Original Centres, scaling factor = 16



(l) Centres modified, scaling factor = 16

Fig. 9.37: Interpolated Straight Lines Joining Centres of Clusters

patterns. As mentioned above, when the scaling factor is increased, so is the number of interpolated points and thence the storage requirement of scan patterns. Variability of scaling factor offers the flexibility of dealing with different sizes of sets of data, representative of cluster centres, to obtain the best feasible display for each set. Accordingly, if the number of cluster centres is small and/or the distance between them is short then a large scaling factor can be used; while if the number of centres is large and/or the distance between them is long, a small value of the scaling factor must be used to avoid overflowing the storage space allocated for interpolated points.

It must be recalled that raw fixation points are multiplied by a constant A before displaying them on the storage oscilloscope (Section 9.6.2). It has been found that when the value of the constant A is 2^4, continuous straight lines are given in all directions and for this reason it has been chosen for calibrating the oscilloscope. If a scan pattern is complex, the cluster centres are scaled by a constant B, which is less than 2^4. Then after straight lines are generated, the coordinates of the points which constitute them are multiplied by (2^4-B) to bring the displayed pattern to the proper size. When B is less than 2^4; interpolated points in some directions do not merge into a continuum.

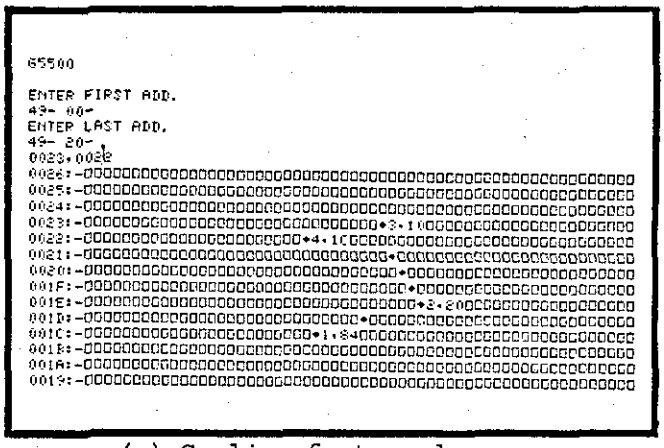
The plates are grouped into two sets, those to the left show the result when original cluster centres are employed, while those to the right illustrate the effect of running MODCEN. Referring to Plates (c) and (d); in the latter MODCEN causes the cluster centre in the extreme right to be displaced a distance, equivalent to one pixel,

to the left. Starting at this centre and moving in an anticlockwise direction, displacement of the next centre is more pronounced. The other two centres assume their original values. Depending upon the type of experiment, an operator has to compare a display with and without using MODCEN to appraise the amount of distortion introduced. This distortion is due to the rounding up of fractions when straight lines are interpolated (Section 9.9.3).

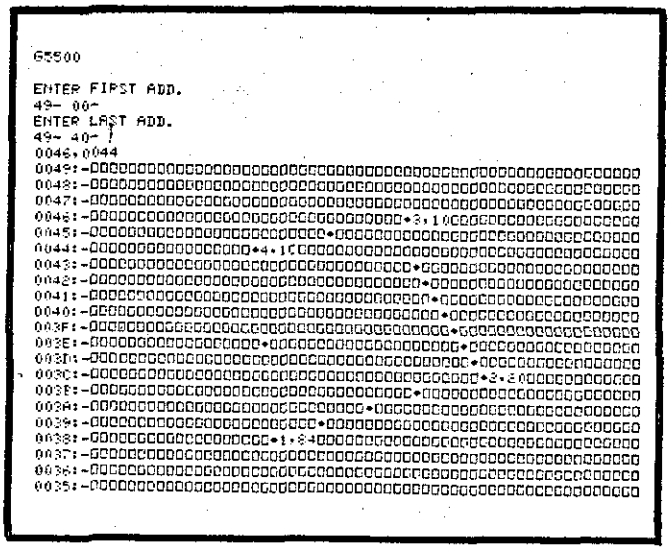
Patterns in Fig. 9.38 and Fig. 9.39 are hardcopy displays of some of the results obtained above, with annotations, indicative of the sequence of cluster centres and number of points within each cluster domain, tagged to each cluster centre. These Figures give a quantitative measure of the displacements of the cluster centres caused by running MODCEN. Hardcopy displays are slightly elongated in the vertical direction because spacing between characters is greater in that direction than in the horizontal direction. Patterns are shown for different scaling factors. The maximum scaling factor used is 2^2, and any value greater than this makes the pattern too big to be accommodated by the display space. Hardcopy displays lack both the flexibility and speed of storage oscilloscope displays. Due to limited space and poor resolution, it is not possible to obtain continuous straight lines. However, they offer the advantage of presenting the annotations and they give quantitative measure of cluster centres and distortions induced by MODCEN.

9.11.2 Real-Data

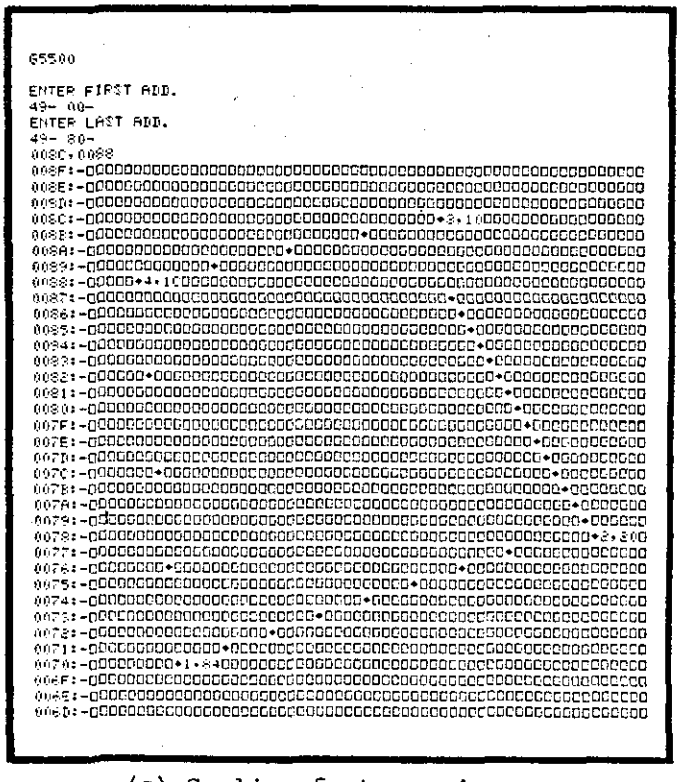
Photographs shown in subsequent figures are results obtained using real-data, collected while two subjects were viewing different



(a) Scaling factor =1

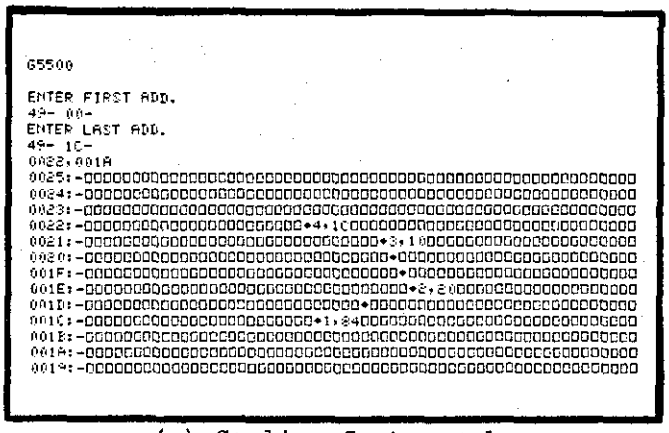


(b) Scaling factor = 2

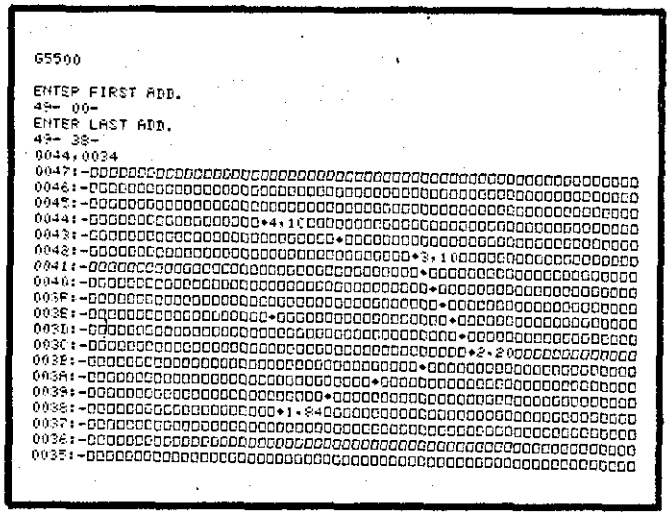


(c) Scaling factor = 4

Fig. 9.38: Hardcopy Display of Straight Lines Joining Original Centres of Clusters



(a) Scaling factor = 1



(b) Scaling factor = 2

(c) Scaling factor = 4

Fig. 9.39: Hardcopy Display of Straight Lines Joining Modified Centres of Clusters

stimulus patterns. Reference to scaling factors and whether MODCEN was employed or not is avoided unless it is essential. However, in the real-data analysed, very often fixation points lie in close proximity, sometimes separated by one pixel, and consequently errors introduced by MODCEN are very few.

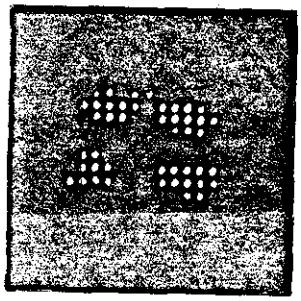
The sequence of photographs in Fig. 9.40 was obtained when a subject was asked to view a set of numbers (1, 2, 3, 4) arranged at corners of a rectangle. The numbers were arranged as follows

3 2

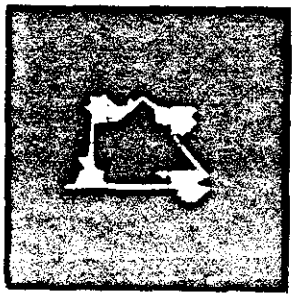
4 1

and the subject was asked to transfer his point of gaze according to the sequence 1, 2, 3, 4; and return it back to 1 again. The numbers were clearly written in large numerals and the subject had prior knowledge about how they were arranged, so he did not adopt any particular search pattern, but rather directed his point of gaze directly from one number to the next.

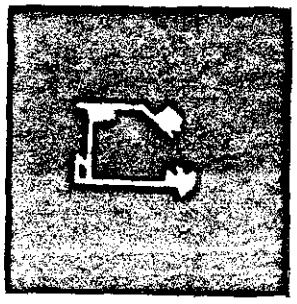
Plate (a) shows points of fixation displayed in real time. Plate (b) depicts the original fixation-point scan pattern. An experimenter can observe on the oscilloscope screen the sequence by which the subject transfers his point of gaze from one point to another. Plates (c-g) illustrate the ability of the clustering algorithm in compressing the raw data and extracting salient features of the scan pattern.



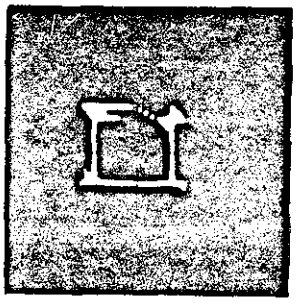
(a) Fixation Points evaluated in real-time



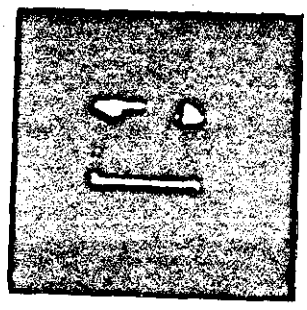
(b) Raw Eye Pattern Track



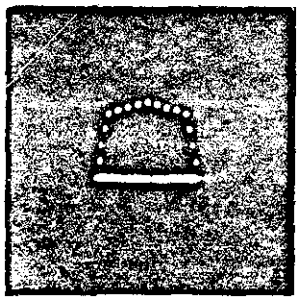
(c) Processed Eye Track (R=1, T=2)



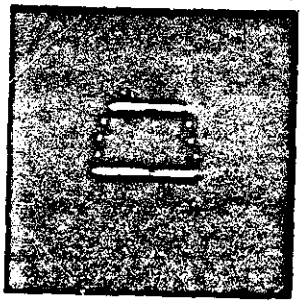
(d) Processed Eye Track (R=2, T=2)



(e) Processed Eye Track (R=4, T=2)



(f) Processed Eye Track (R=5, T=2)



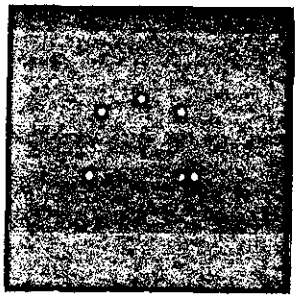
(g) Processed Eye Track (R=5, T=5)

Fig. 9.40: Eye Track of a Subject Viewing Numbers Arranged at the Corners of a Rectangle

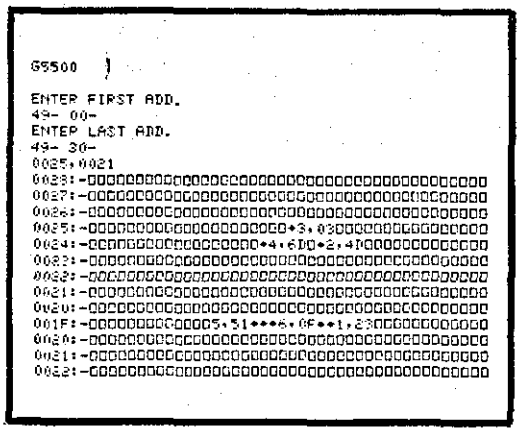
The records of eye movement show that the subject did not move his gaze directly from point 2 to 3. He fixated his gaze for a short while at a point somewhere between these two. By increasing the dwell time threshold to 5, the clustering algorithm eliminates this point (Plate (g)). Centres of clusters for R=5, T=2, corresponding to vertices of scan pattern of Plate (f), are shown in Fig. 9.41, together with a hardcopy display of annotated eye track. The hardcopy is shown for two values of scaling factor. Annotations indicate that the subject transferred his look point in accordance with the sequence he was asked to follow. Fig. 9.42 illustrates the five cluster centres, after elimination of the point between 2 and 3, and the corresponding hardcopy display. Because of the limited space available for the hardcopy display, the number of interpolated points is very small.

The sequence of plates in Fig. 9.43 shows results when a subject was asked to fixate his gaze at point 1,then look straight ahead before transferring his eye to point 3. The plates show raw fixation points and processed eye track obtained for different values of the clustering algorithm parameters,R and T. The last two plates illustrate scan pattern and corresponding centres of clusters for R=10, N=35.

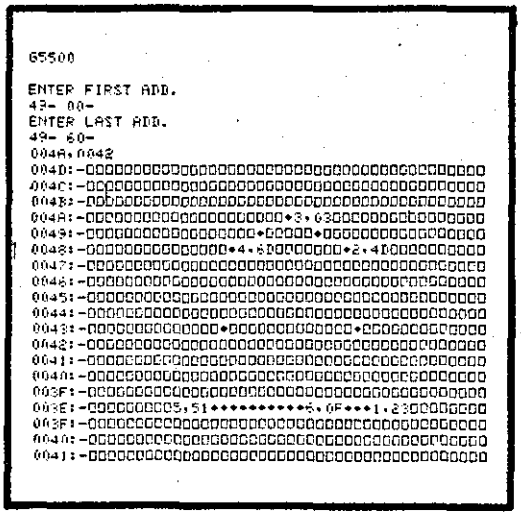
In Fig. 9.44,the subject was asked to transfer his look point from 2 to 3 to 4. Comparing Plates (b) and (c), in the latter, a greater value of R is chosen so as to merge all fixation points at each area of interest into a single cluster. The last two plates (d & e) are included to illustrate the ability of the clustering algorithm to suppress spurious points or quick eye excursions,



(a) Centres of Clusters
when R=5, T=2

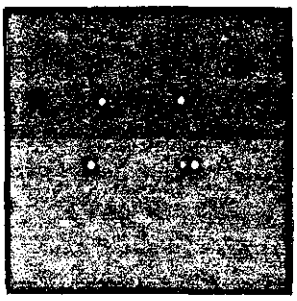


(b) Hardcopy display of eye track
scaling factor = 1



(c) Hardcopy display of eye track
scaling factor = 2

Fig. 9.41: Eye Track of a Subject Viewing Numbers Arranged at Corners of a Rectangle



(a) Centres of Clusters
when R=5,T=5

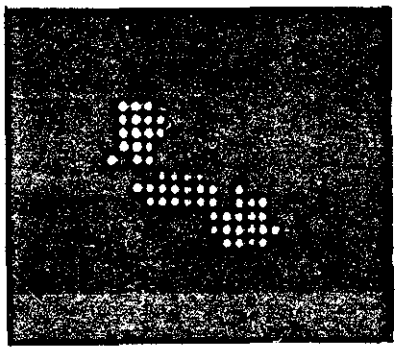
```
G5500

ENTER FIRST ADD.
49- 00-
ENTER LAST ADD.
49- 40-
0024:001E
0027:-00
0026:-
0025:-
0024:-
0023:-
0022:-
0021:-
0020:-
001F:-
0020:-
0021:-
0022:-
```

(b) Hardcopy display of eye track
Scaling factor = 1

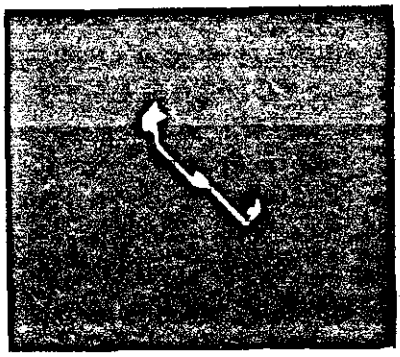
```
G5500

ENTER FIRST ADD.
49- 00-
ENTER LAST ADD.
49- 80-
0048:003C
0048:-
0048:-
0049:-
0048:-
0047:-
0046:-
0045:-
0044:-
0043:-
0042:-
0041:-
0040:-
003F:-
003E:-
003F:-
0040:-
0041:-
```

(c) Hardcopy display of eye track
Scaling factor = 2

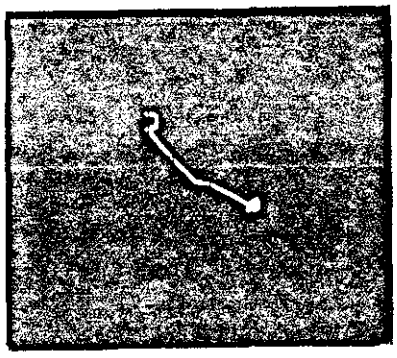
Fig. 9.42: Eye Track of a Subject Viewing Numbers Arranged at Corners of Rectangle



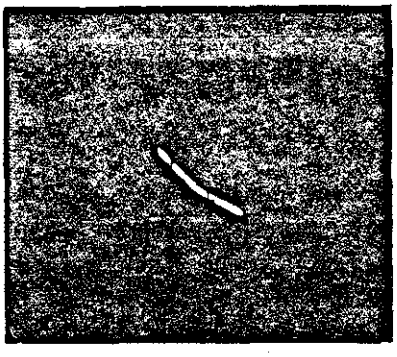
(a) Fixation Points Evaluated in Real-Time



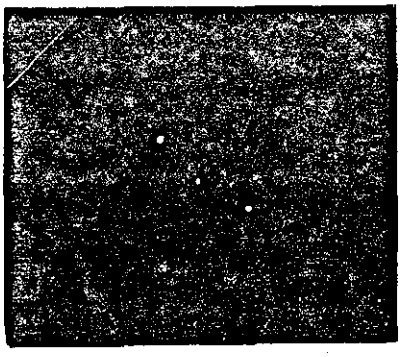
(b) Processed Eye Track (R=5, T=5)



(c) Processed Eye Track (R=7, T=7)



(d) Processed Eye Track (R=10, T=35)



(e) Centres of Clusters (R=10, T=35)

Fig. 9.43: Eye Track of a Subject Looking at 3 Points Arranged Diagonally

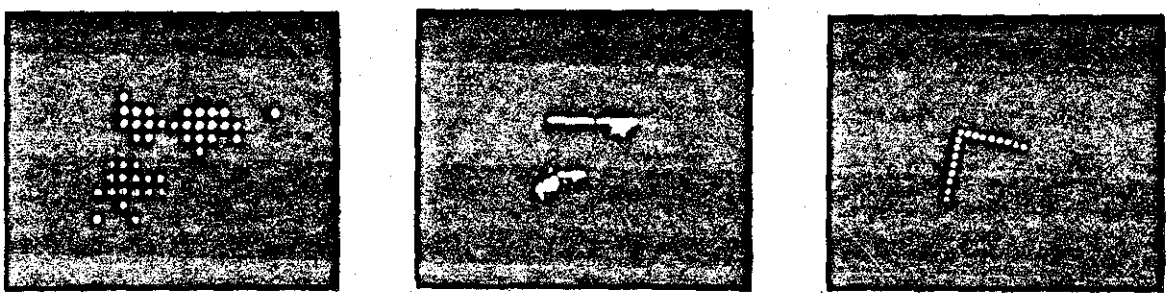
exhibited when the eye drifts away from the area of interest and is brought back quickly. In both plates the cross denotes the centre of the subsequent cluster. Plate (d) shows fixation points constituting the first cluster and their location relative to the centre of the second cluster. Plate (e) depicts eye track before moving to second area of interest. Here there are two excursions, one to the left and the other to the right of the main concentration of cluster points. By setting the minimum dwell time to 2 these excursions are eliminated as shown in Plate (b).

Fig. 9.45 shows records of eye movements when a subject was asked to trace visually the circumference of a circle drawn on a uniform background. Plate (a) illustrates the raw scan pattern and here the eye track does not invariably lie along the circumference, due to involuntary saccades arising during attempts to trace lines in general[78]. Subsequent plates show processed eye track. The subject did not fixate his gaze at any particular point of the stimulus pattern for a long time, but rather followed its outline at a fairly regular pace. Setting T=2 deletes most of the excursions shown in the raw eye track. To obtain a smooth scan pattern R is increased while T is kept equal to 2. Values of T>3 cause elimination of most of the cluster centres.

The sequence of plates shown in Fig. 9.46 - Fig. 9.48 shows records of eye movements of two subjects while reading the two lines;

THROUGHOUT HISTORY MOST OF THE CIVILIZATIONS

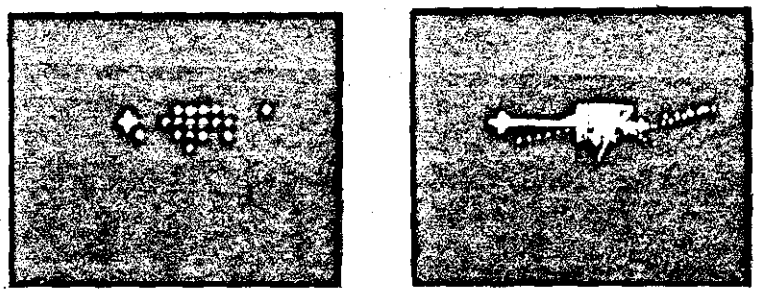
THAT DECLINED WERE VICTIMS OF STAGNATION



(a) Fixation Points evaluated in real-time

(b) Processed Eye Track (R=2, T=2)

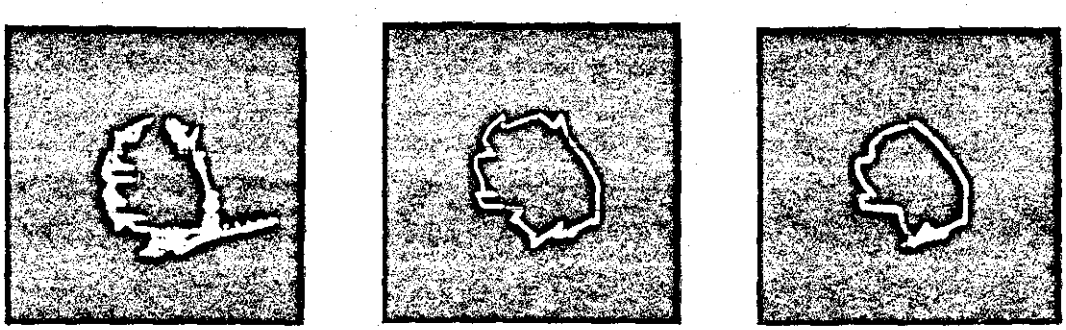
(c) Processed Eye Track (R=13, T=12)



(d) 1st Cluster of fixation points

(e) Raw Fixation Point scan pattern in 1st cluster

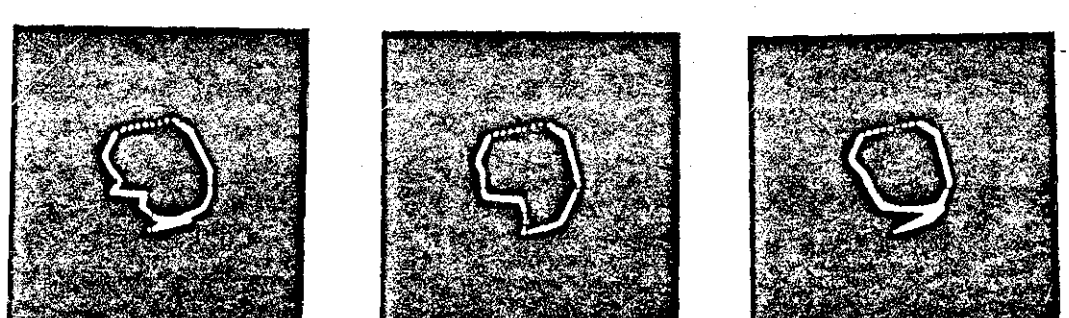
Fig. 9.44: Eye Track of a Subject Viewing Numbers Arranged at Three Corners of a Rectangle



(a) Raw Eye Track

(b) Processed Eye Track (R=2, T=2)

(c) Processed Eye Track (R=4, T=2)

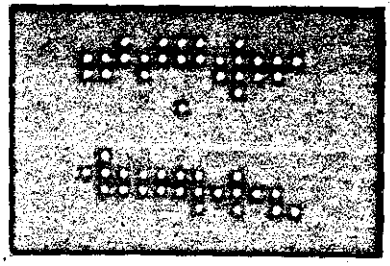


(d) Processed Eye Track (R=5, T=2)

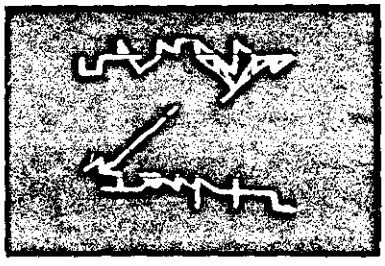
(e) Processed Eye Track (R=10, T=2)

(f) Processed Eye Track (R=16, T=2)

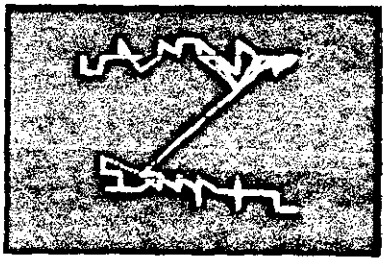
Fig. 9.45: Records of Eye Movements of a Subject Tracing Visually the Circumference of a Circle



(a) Fixation Points evaluated in real-time

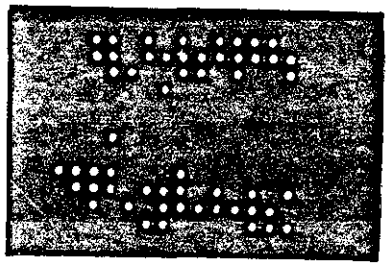


(b) Scan Pattern using original fixation points

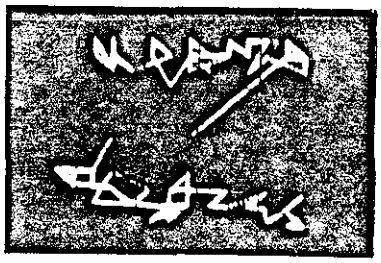


(c) Scan Pattern after running MODCEN

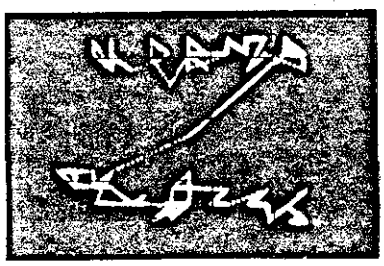
Fig. 9.46: Records of the Eye Movements of a Subject Reading Two Lines



(a) Fixation Points
evaluated in real-time

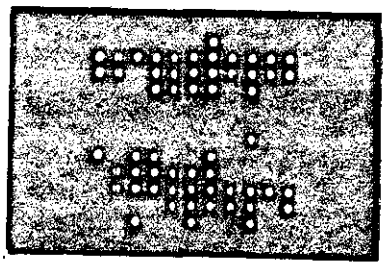


(b) Scan Pattern using
original fixation points

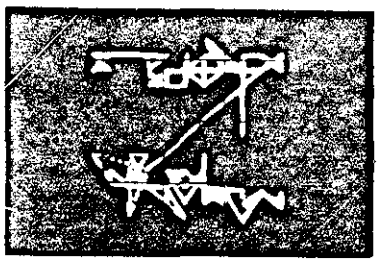


(c) Scan Pattern after running
MODCEN

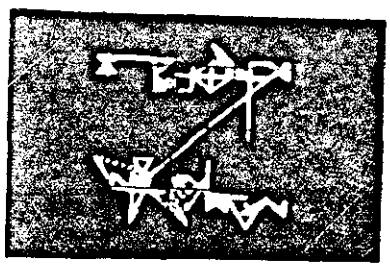
Fig. 9.47: Records of the Eye Movements of a Subject Reading Two Lines



(a) Fixation Points
evaluated in real-time



(b) Scan Pattern using
original fixation points



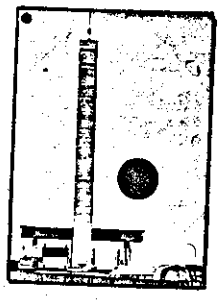
(c) Scan Pattern after running
MODCEN

Fig. 9.48: Records of the Eye Movements of a Subject Reading Two Lines

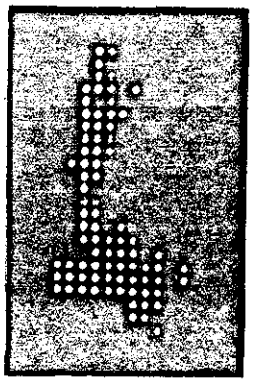
The records in Fig. 9.46 and 9.48 are for the same subject. Both subjects are post-graduates and have good reading ability. When reading the text, their eyes do not pause for a long time at any particular part of the two lines, but rather skim quickly over them. For this reason the clustering algorithm is not applied to achieve data reduction as points of fixation do not form natural clusters. The records shown represent original raw fixation points connected by straight lines to elucidate the manner in which the two lines were scanned. In the first two figures there is a fixation point lying between the two horizontal traces. The sequence of occurrence of this point indicates that it was captured while the subjects were transferring their point of gaze quickly to the beginning of the second line. As illustrated in the records, MODCEN hardly introduces any error in that part of the trace corresponding to the text, where fixation points are in close proximity. Errors are mainly incurred on the points located between the two lines.

Records of eye movements of subjects during free examination (without instruction) of complex stimulus patterns are presented in subsequent plots. Scan patterns obtained during experiments agree to a great extent with information given, later, by the subjects concerning the order at which they shifted their point of gaze from one area of the stimulus pattern to another.

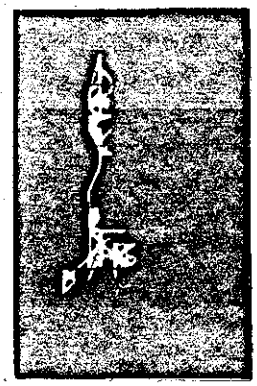
Fig. 9.49 includes records of eye movements of a subject during examination of a photograph of a tower shown in Plate (a). Due to the large number of fixation points collected (539) and limited store available for data representing scan patterns, the fixation points were divided into 3 parts, and raw eye track was displayed in turn



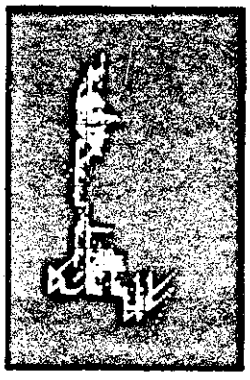
(a) Stimulus Scene



(b) Fixation Points evaluated in real-time



(c) 1st Part of Raw Eye Track



(d) Processed Eye Track (R=1, T=1)

(e) Processed Eye Track (R=1, T=2)

(f) Processed Eye Track (R=1, T=3)

(g) Processed Eye Track (R=2, T=2)

(h) Processed Eye Track (R=2, T=3)

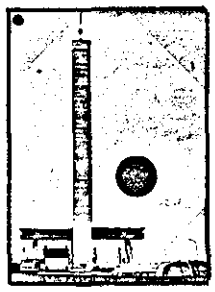
Fig. 9.49: Photograph of a Tower and Records of the Eye Movements During Examination of the Photograph

for each. It was difficult to combine the 3 tracks to obtain a single display on the storage oscilloscope, because it has low persistence and the pattern displayed first tends to fade before the arrival of the next one. However detailed representation of a raw eye track for the first 192 fixation points is shown in Plate (c).

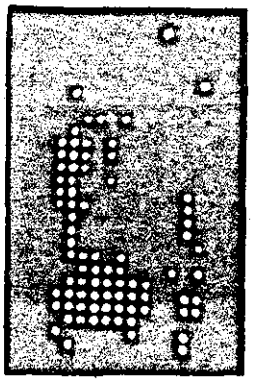
Analysis of the sequence of occurrence of raw fixation points shows that the subject started examination of the scene by looking at a point near the top of the tower, shifted his point of gaze upwards along the mask, and downwards towards the lower part of the tower; examined details to the right, then to the left and to the right again; and finally moved his eye up the tower and repeated the process again, but this time paid more attention to details at the bottom of the picture, next to the tower and below the bridge.

Processed eye track is illustrated in Plates (d-h), for different values of R and T. The scan patterns indicate that, the subject fixated his gaze more often at the top and bottom parts of the tower, the middle part attracting little attention from him. At the lower part of the picture he shifted his sight more towards the right and downwards to examine the top of the bus. These scan patterns conform to what might be expected a priori.

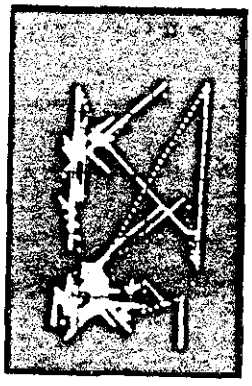
The same stimulus pattern was presented to another subject and records of his eye movements are shown in Fig. 9.50. Plate (b) illustrates fixation points collected in real time. Points on the right side of the plot are spurious and appear whenever brightness of the pupillary disc increases greatly as a result of an increase in the amount of infra-red radiation reflected from the retina, causing loss of eye detail. Such changes in retinal reflections are due to



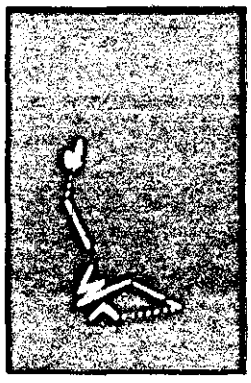
(a) Stimulus Scene



(b) Fixation Points evaluated in real-time

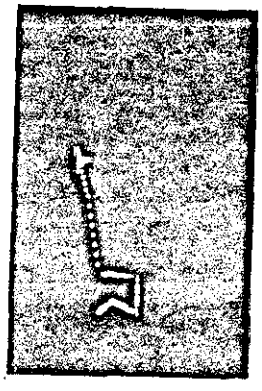


(c) Processed Eye Track (R=1, T=1)

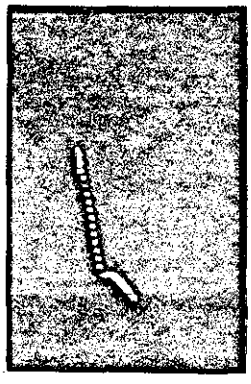


(d) Processed Eye Track (R=1, T=3)

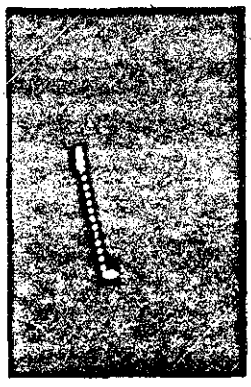
(e) Processed Eye Track (R=2,T=3)



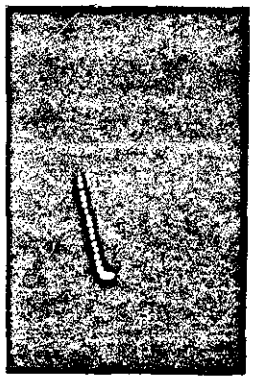
(f) Processed Eye Track (R=2, T=4)



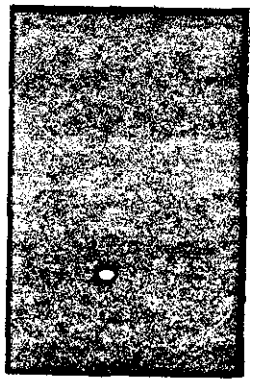
(g) Processed Eye Track (R=2, T=5)



(h) Processed Eye Track (R=2, T=6)



(i) Processed Eye Track (R=2, T=7)



(j) Processed Eye Track (R=2, T=8)

Fig. 9.50: Photograph of a Tower and Records of the Eye Movements During Examination of the Photograph. Here fixation Point Pattern ContainsMany Spurious Points.

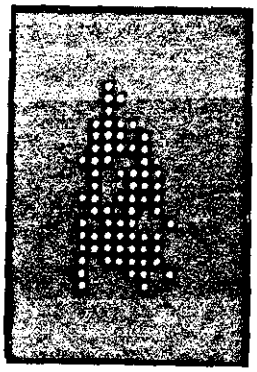
limitations inherent in the oculometer. These have been discussed elsewhere (Section 9.3).

When the brightness of pupillary disc increases drastically; the corneal reflection may disappear or become greatly distorted. Whenever it disappears, the program detects this, rejects the corresponding set of data and waits for a new one. Occasionally, the corneal reflection becomes distorted by an increase in its size and the emergence of spurious edge points within it, and between it and the pupillary disc. The program cannot detect this and as a consequence erroneous fixation points, located at a large distance from the original eye track, are given. Whenever brightness changes and image of eye reflection starts to deteriorate, the operator can adjust quickly the D.C. level of the video signal and sometimes has to alter the reference voltage connected to the ADC. Adjustments are made through two controls on the front panel of VAS, and they take a very short time.

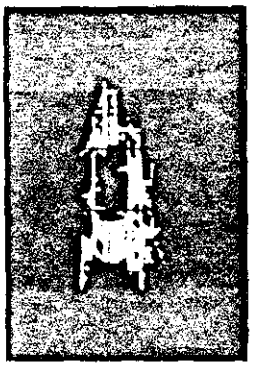
Returning to Fig. 9.50, Plate (c) shows processed eye track for R=1, T=1. For T≥3 all spurious points disappear as illustrated in subsequent plates. T is increased progressively to isolate areas where the subject has fixated his gaze for longer periods.

Records in Fig. 9.51 represent eye movements of a subject asked to examine a photograph of an ancient tomb, shown in Plate (a). The raw fixation point scan pattern (Plate (c)) indicates that the subject paid more attention to the periphery of the monument. According to the oscilloscope trace, at first he fixated his gaze at a point approximately at the middle of the left side of the tomb, then moved his eye upwards and examined the pillar at the top. After that his

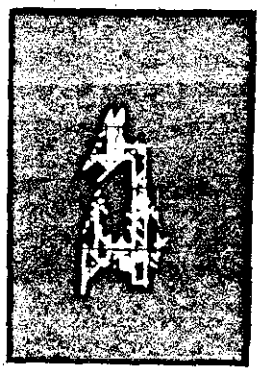
(a) Stimulus scene



(b) Fixation Points evaluated in real-time



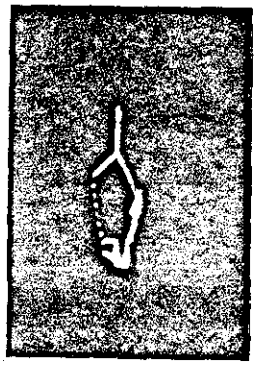
(c) Raw Eye Track



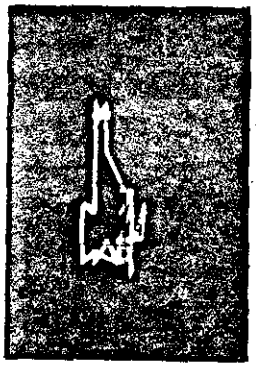
(d) Processed Eye Track (R=1, T=1)

(e) Processed Eye Track (R=1,T=2)

(f) Processed Eye Track (R=1, T=3)



(g) Processed Eye Track (R=1, T=4)



(h) Processed Eye Track (R=2, T=3)

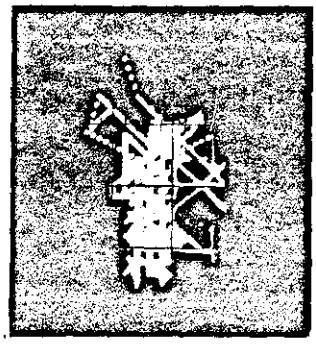
Fig. 9.51: Photograph of an Ancient Tomb and Records of the Eye Movements During Examination of the Photograph

eye travelled downwards along the right side of the tomb, around its lower part and up again towards the point where he had started. It seems that the palm tree, at the lower right-hand corner, attracted little of his attention. Plates (d-f) show samples of processed eye track for different values of R and T.

Fig. 9.52 contains a photograph of skiers and processed eye track of a subject during free examination of the photograph. There is a lot of smearing in Plate (b) due to the proximity of traces. It is much easier to interpret this record on the oscilloscope screen, where an operator can observe a bright spot moving slowly from one fixation point towards another. The subject started by fixating his gaze in the direction of the man's face, with excursions to the right and left. Then he moved his eye, mainly in upward and downward directions, to examine the glove, the stick supporting it and the skis and sticks between the man and the woman. Finally, he directed his gaze towards the woman's face, with excursions to the right in the direction of the man, and to the left towards the woman's head cover. In Plate (c) the dwell time threshold is increased to 5 and the record shows that the subject concentrated his attention mainly on the man's face, skis, sticks and the woman's face, which are the prominent features of the photograph . Plate (d) shows his eye track for T=9, to demonstrate the parts of the pictures to which he paid most of his attention, namely the faces of the man and woman and the lower part of the ski next to the man.

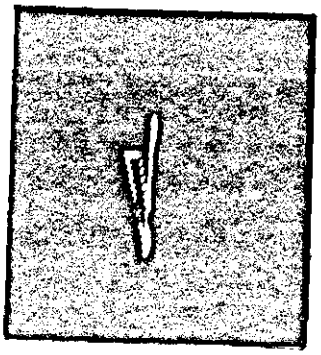
The sequence of photographs in Figures (9.53-9.56) represents eye tracks of a subject viewing the photograph of a sailing boat shown in Fig. 9.53, Plate (a). Plates (b) and (c), in the same figure, illustrate fixation points collected in real time and processed eye track for R=1, T=2.

(a) Stimulus Scene



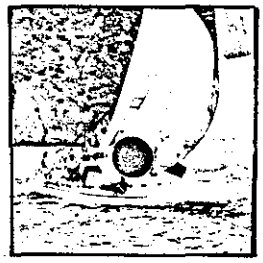
(b) Processed Eye Track
(R=1, T=1)

(c) Processed Eye Track
(R=3, T=5)

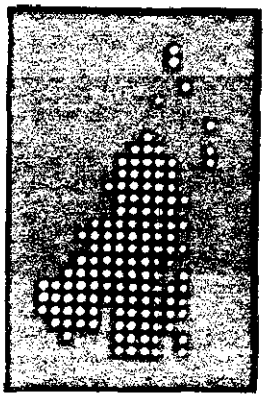


(d) Processed Eye Track
(R=3, T=9)

Fig. 9.52: Photograph of Skiers and Records of Eye Movements During Examination of the Photograph



(a) Stimulus Scene

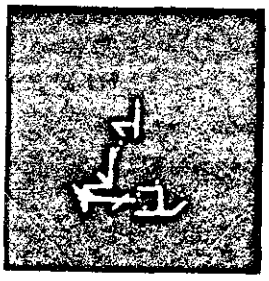


(b) Fixation points evaluated in real time

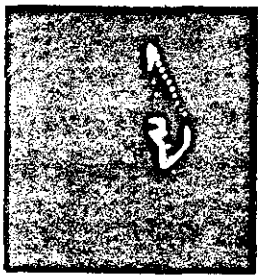
(c) Processed Eye track (R=1, T=2)

Fig. 9.53: Photograph of a Sailing Boat and Records of Eye Movements During Examination of the Photograph.

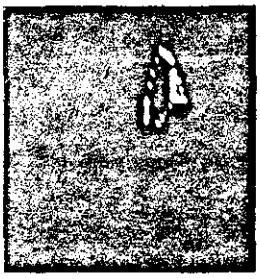
(a)



(b)



(c)



(d)

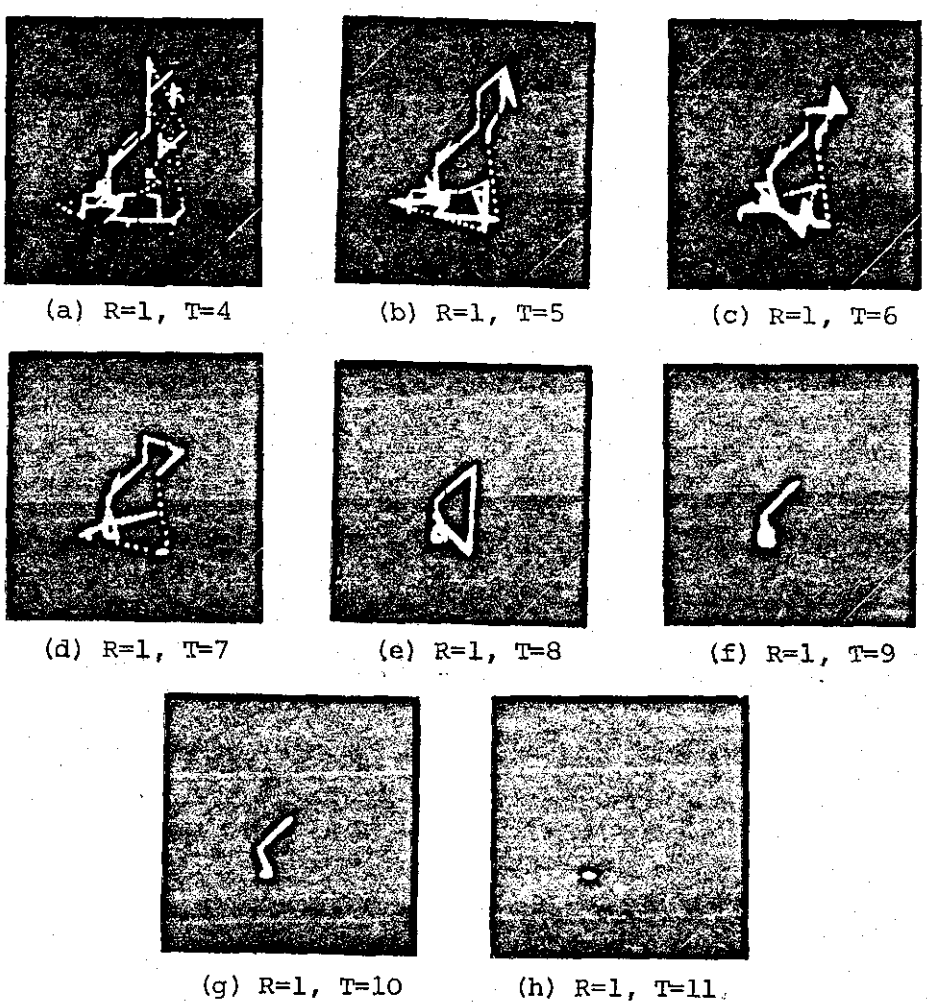
(e)

Fig. 9.54: Processed Eye Track (R=1,T=3) Based on Data Shown in Fig. 9.53(b). MODCEN is Executed Before Interpolating Straight Lines

Fig. 9.54, Plate (a), shows eye track for R=1, T=3. This track is divided into 4 consecutive parts, shown in Plates (b-e) to demonstrate the sequence of the eye movements. Direct observation of the oscilloscope trace indicated that the subject at first examined the inner side of the big sail, proceeded downwards along the part of the boat where most of the men are sitting, and directed his gaze across the boat and started to shift his vision upwards (Plate (b)). Next he examined the sails (Plates (c)and(d)), and again he directed his attention downwards, towards the people sitting on the front of the boat (Plate (e)), and then across the whole length of the boat. Finally, he shifted his gaze upwards, examined part of the sails and returned his line of sight back to the front of the boat.

Records of Fig. 9.55 show processed eye track for increasing values of T, while R is kept equal to 1, to extract salient features of the scan pattern. According to Plate (h), the place where the subject fixated his gaze for the longest time is somewhere in the front of the boat where most of the men are sitting. All previous records of the subject were traced after implementing MODCEN. Similar records, without running MODCEN were obtained and when compared to corresponding ones shown in Fig. 9.55, some were found to be identical. Records which are not identical are included in Fig. 9.56. The difference is insignificant and is hardly noticeable, except in the last record depicted in Plate (e). Comparing this with the corresponding record in Plate (f) of Fig. 9.55, MODCEN caused the two separate traces to merge into one.

The last set of records, shown in Fig. 9.57 and 9.58, illustrate eye movements of a subject examining a human face. When looking at a human face, an observer usually pays most attention to the eyes, lips

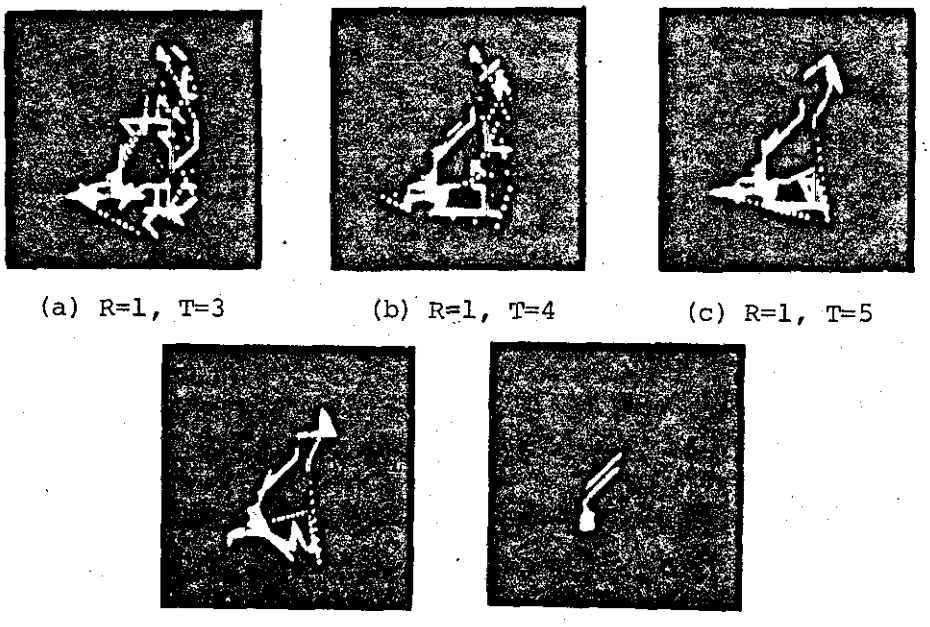


(a) R=1, T=4 (b) R=1, T=5 (c) R=1, T=6

(d) R=1, T=7 (e) R=1, T=8 (f) R=1, T=9

(g) R=1, T=10 (h) R=1, T=11

Fig. 9.55: Processed Eye Track Based on Data Shown in Fig. 9.53(b) MODCEN is Executed Before Interpolating Straight Lines

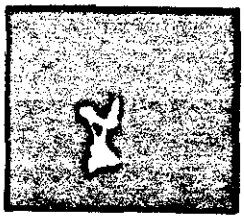


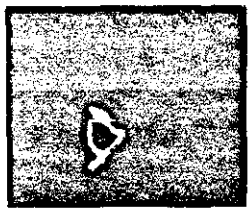
(a) R=1, T=3 (b) R=1, T=4 (c) R=1, T=5

(d) R=1, T=6 (e) R=1, T=9

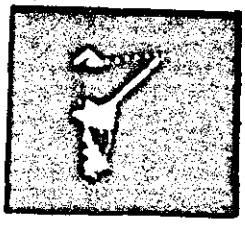
Fig. 9.56: Processed Eye Track Based on Data Shown in Fig. 9.53(b) MODCEN is not Executed.

(a) Stimulus scene

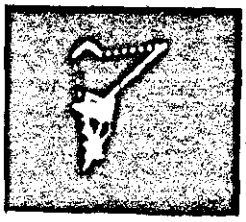




(b) Processed eye track (R=4, T=9)

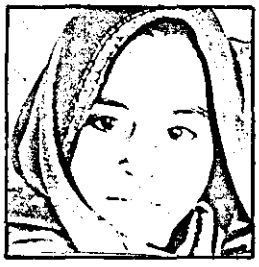


(c) Processed eye track (R=4, T=4)

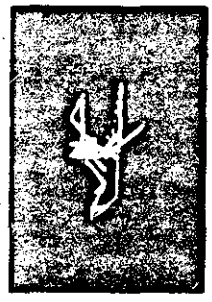


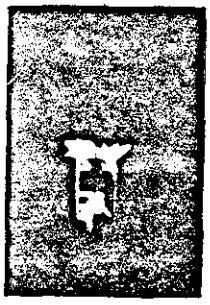
(d) Processed eye track (R=4, T=5)

Fig. 9.57: Photograph of a Girl's Face and Records of Eye Movements during examination of the photograph



(a) Stimulus scene





(b) (c) (d)

Fig. 9.58: Photograph of a Girl's Face and Processed Eye Track During Examination of the Photograph (R=3,T=3)

and nose. In Fig. 9.57(b), the trace is divided between two plates to give a better understanding of the scan pattern. Starting with the plate to the left, and in accordance with the oscilloscope plot, the subject examined the right eye, the left eye and then the lips. After that he shifted his gaze around the face (second plate), in an anticlockwise direction, i.e. the same direction as in the first trace. More information about how he examined the face can be obtained by lowering the dwell time threshold, as illustrated in records of Plates (c) and (d). These records show that the subject moved his eye upwards to examine the hair and then returned it quickly to the face. The records also show repeated alternate fixations between the two eyes.

Processed eye track in Fig. 9.58 are based on data collected during examination of a girl's photograph shown in the first plate. Plates (b) and (c) exhibit the first and second parts of the eye track. Both parts are combined together in Plate (d). Again, here the main areas of interest are the distinctive features of the human face, i.e. eyes, nose and lips.

9.12 Discussion

Eye movements records obtained during experiments in which stationary stimulus patterns were presented, show "jump and rest" fixation movements, the characteristic features of the saccadic eye movements. When asked to examine the circle, the subject was instructed to follow its circumference smoothly. For him the tracking movements seemed uninterrupted, but in fact, as records illustrate, they are not smooth and moreover, are not aligned with the circumference. This is due to involuntary saccadic movements of which he was unaware. It should be

noted that it is not only small saccades which are involuntary; large saccades are also involuntary[78].

Number of fixation points, their distribution and the order at which they occur are the indicators which give an idea about the perception of complex objects. The fixation-point scan patterns obtained, show that the subjects fixated on elements that give information essential for interpretation of the picture. Very little or no attention is paid to the background. This is shown by the distribution of fixation points during examination of the tower and the tomb, where the background is uniform. In the photograph of the sailing boat, although the cliff in the background is composed of many details, it did not attract the attention of the subject.

Concerning elements of the pictures examined, some of them receive most attention, while others are given only cursory consideration. Previous research has indicated that elements attracting attention, may contain information, in the observer's opinion, useful and essential for perception[78,104]. For example, the human eyes and lips indicate the mood of a person, his attitudes and may give cues to his next behavioural patterns. Therefore it is understandable that they attract the attention more than any other part of the human face.

When asked to examine the pictures, the subjects at first fixated on all elements which, for them, were useful for perception. Then they returned to the same elements examined before and spent more time on those which are important. This is exemplified by the records obtained when the subject was shown the picture of the tower (Fig. 9.49). After examining the tower and other details at the bottom, he moved his eye up and down again to re-examine elements he had fixated before.

The same can be seen in Fig. 9.54. After examining the whole boat, the subject repeated the process, but in the second cycle he ignored the upper part of the sails and spent more time on the area of the boat where most of the people are sitting. The glow of the trace in that area (Plate (e)) is indicative of the concentration of scan lines. Re-examination of elements previously fixated were also noticed when pictures of the two girls were used as stimulus patterns. The subject's fixation alternated between the girl's eyes in both experiments.

9.13 Conclusion

VAS has successfully performed all functions essential for eye motion analysis. Its edge detector is very efficient in extracting pupillary disc and corneal flux from a rather noisy video signal. Unlike some other real time systems, blinking or sneezing does not result in momentary loss of the continuity in the eye image. The microprocessor can track the human eye and determine its point-of-regard in real time. Besides this, it is an adequate tool for further statistical analysis of stored data, and presentation of results in a format that is easy to interpret.

Difficulties encountered were mainly due to the oculometer and the environment under which the experiments were undertaken. During these experiments, controls on the front panel of VAS were adjusted from time to time to offset wide variations in brightness of the eye image. With a better designed and constructed oculometer,there is no need to change setting of the controls performed at the onset of an experiment. The edge detection technique is tolerant to reasonable variations of brightness,as it is based on evaluation of differentials.

Concerning environment, the very poor lighting condition, small space and the means by which stimulus patterns were presented, created many problems. Compounding these problems;is the low resolution of the oculometer and its small range.

At present 8K words of memory are allocated for the programs dealing with eye motion analysis and tracking.
Because of the limited store, the whole program is divided into two parts, one for the real time tracking of the eye and the other for processing and displaying stored data. After execution of the first part, the second part is loaded on the same area, which is about 2 K words. The other 6 K words is allocated for storage of data, intermediate and final results.

9.14 Recommendations

VAS has worked well to establish the feasibility of its implementation in different research and industrial applications pertaining to eye movements. Improvements can be made to its speed and processing power, storage capacity, display unit, and FIFO's buffer store to enhance its capabilities. Suggested improvements are summarised below:-

1. Since the emergence of the TMS9900, faster and more powerful microprocessors have been introduced to the market. The TMS9900 can be replaced with one of the new microprocessors to make it possible to obtain fixation points at the rate of 50 samples per second, and to perform a more sophisticated statistical processing of the data.

2. To allow for the acquisition of more eye point-of-regard data, and generation of numerous interpolated points the store must be expanded. Extra space may also be required for user programs. Storage capacity of 16 K words is quite adequate for most purposes.

3. To improve the display, the telequipment DM64 storage oscilloscope must be replaced by a high resolution storage oscilloscope. A character generator may be incorporated in the display system to label the eye scan path with its temporal components, indicate blink rates and supply information about subjects and types of experiments. Another possible refinement is to use a real time clock to indicate duration of experiments on the display screen. A hardware vector generator, which allows the drawing of lines with minimum amount of data,can be included.

The major drawback of a system based on a storage tube display is the fact that there is no direct way for superimposing fixation scan patterns automatically over a stimulus scene. So, another option for improving the display facility, is to acquire a raster scan computer graphics display terminal which generates alphanumerics as well as graphics. Such terminals are much more expensive than storage tubes, but they are advantageous in that it is easy to combine eye fixation scan patterns and stimulus scenes in one display.In addition,they offer the facility of providing interaction between eye-motion

signals and computer-generated displays, a technique useful for simulation experiments and for eye-control applications.

With the storage or raster scan tubes, provision has to be made for obtaining a hardcopy of the display. Displays by the Silent 700 are of low resolution and are inadequate for most practical applications.

4. To achieve high angular resolution in eye movement measurements, it is envisaged that in an improved system the pupillary disc may cover about 1/3rd of the monitor screen, i.e, approximately 90 scan lines per field. To cope with the corresponding increase in the number of edge points, the present size of FIFO (64 words) must be increased. A new oculometer has to be built to provide larger eye images and greater range of operating angles.

5. To widen the field of application of VAS, pupil diameter can be made available, for recording on a strip chart recorder, through the addition of a third digital-to-analogue converter, or the pupil diameter can be printed on the edge of the screen of an improved display system. This allows easy correlation of pupil diameter with fixation points. If desired, strip chart recorders can also be connected to the outputs of the other two digital-to-analogue converters to obtain analogue outputs of x- and y- eye directions.

6. Subroutines can be added to provide procedures for the calibration and linearisation of the oculometer output. Calibration is performed to allow for the slight displacement between the corneal reflection and the centre of the pupil, observed when an eye is looking straight ahead. This is due to the non-coincidence of the foveal and geometrical axes of the eye. Linearisation compensates for non-linearities in the relationship between eye-direction and relative corneal reflection position within the pupil, for eye angle-of-rotation greater than 10^{o} or 20^{o} [83]. The nonlinearities are due, in part, to the way in which the position of the centre of the pupil is inferred from the partial boundary of an obliquely viewed,and thus elliptical pupil, and in part, to the geometry of the eyeball.

   Calibration and linearisation must be performed for each subject, because of differences in eye characteristics. In fact, the output signals for the left and right eyes of any given subject may differ significantly because of a difference in the individual's eye geometry.

9.15 Applications

VAS, as a real time processing unit of the eye movements, has many advantages over its minicomputer-based counterparts. It is much cheaper, smaller, more compact, portable and its power consumption is low. Its potential applications by far outstrip those of existing systems.

Two eye motion analysis systems can be built around an improved version of VAS (IVAS): one is a general purpose laboratory system and the other is a miniaturised system. The improvements in VAS are based on the recommendation cited in the previous section.

9.15.1 Laboratory Eye-Motion Analysis System

In addition to IVAS and an oculometer, the laboratory system includes a subject's compartment, in which stimulus patterns are presented on a TV monitor or a rear-projection screen. Any projection device can be used, but a random access projector is more convenient, because it allows automatic microprocessor selection of various stimuli as a function of the subject's prior responses. A unit for displaying results can either be a high resolution storage oscilloscope or a raster scan computer graphics terminal. The latter is more convenient, for it can be used to superimpose eye tracks on stimulus patterns. If a computer graphics terminal is used a scan converter may be needed to transform the stimulus image in a form amenable for display on the terminal. A proposed layout of a laboratory system is shown in Fig. 9.59.

General purpose laboratory systems can be utilised in studies of vision, perception, television viewing, image scanning, novice training, and effects of fatigue, drugs or impaired performance over time. Clinical and psychological applications include: measurement of pursuit and saccadic eye movements, nystagmus; measurement of vergence and muscular imbalance (if position of both eyes are recorded by including extra equipment)[79,80]; and correlation of voluntary and involuntary eye movements to other physiological and psychological variables[82]. Many diseases, such as schizophrenia, apparently

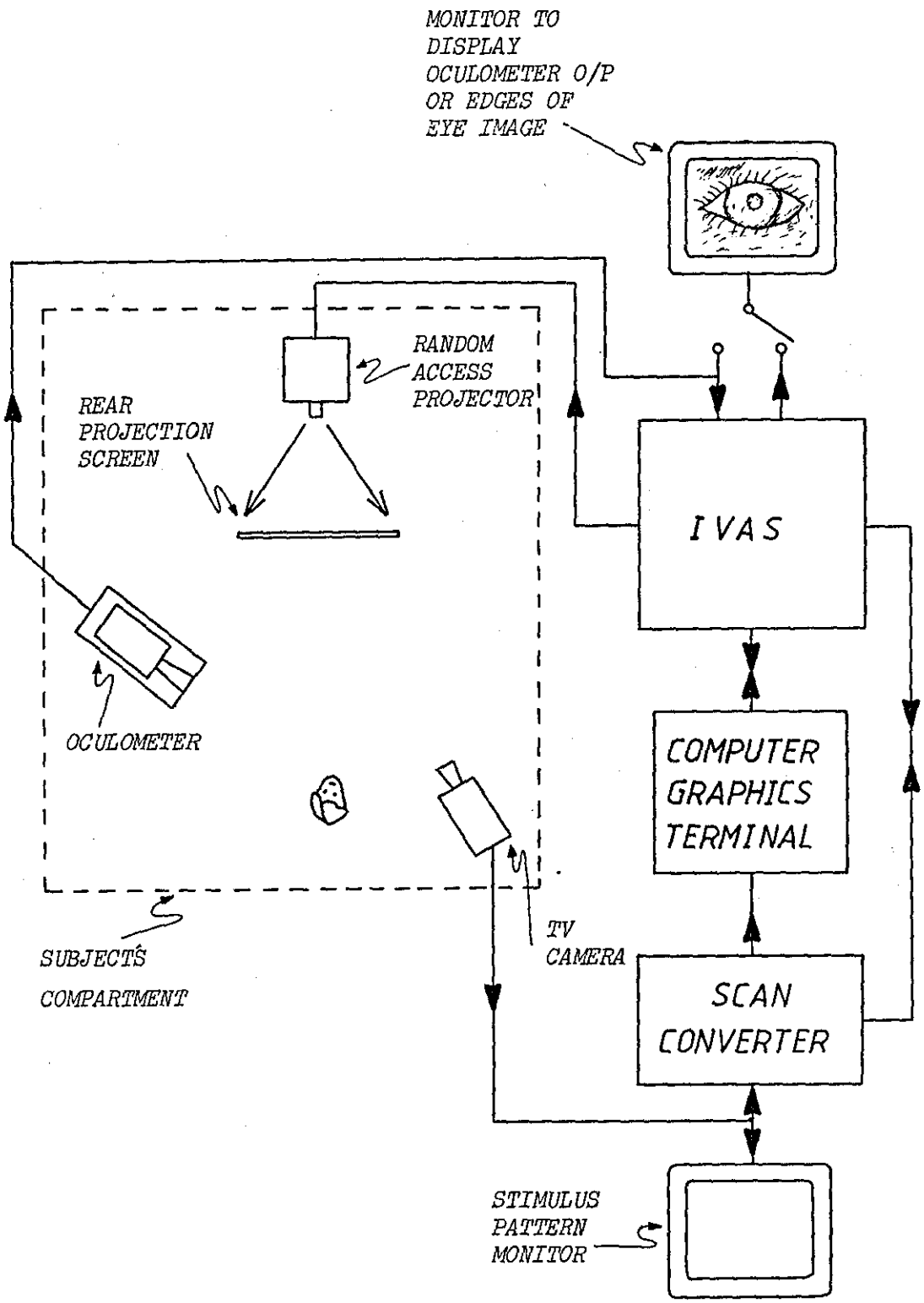


Fig. 9.59: General Purpose Eye Movements Analysis System

affect scanning patterns, and accordingly, eye movements as well as pupillometry (measurement of pupil diameter) are increasingly used as tools in the study of this and various other neurological disorders[79]. The on-line capability of the system can easily be used, simultaneously, to collect data on physiological parameters such as heart rate, electroencephalogram (EEG)[81], etc, as well as data on motor responses such as reaction time. Other potential application is reading analysis and diagnosis of learning disabilities, for instance dyslexia, testing of visual acuity, etc.

9.15.2 Miniaturised System

In some applications, subject variability coupled with an uncontrolled measurement environment makes it extremely difficult to simulate experimental conditions within a laboratory, as for instance in studies of human factors relevant to ground and air transportation. IVAS in conjunction with an oculometer built around a miniature TV camera and small display units offers, for the first time, an eye movements analysis system which can be carried to sites to perform visual performance measurements under conditions unattainable in a laboratory. This system could be very useful in human factors studies, dual-resolution display systems, air-to-air combat and control applications[105]

From a psychological and a human factors perspective, problems related to an operator's ability to acquire, process and apply information can be investigated[82]. This includes: the study of sensory mechanisms, motor skill performance, emotional reactivity, physical capabilities, indices of general health, reaction to stress

and physiological indices of psychological stress; which are important for determining normal operator-performance capabilities as well as for understanding the effects and magnitude of performance decrements due to fatigue, alcohol, drugs, emotional stress, etc. Study of the man/machine interaction can be performed in a vehicle, aeroplane, factory etc., to obtain results under natural conditions.

In a dual-resolution display system,a high resolution, narrow-field-of-view scene is displayed wherever the eye is directed within a low-resolution, wide-field-of-view display. Thus wherever an observer directs his eye, within the wide-field-of-view, he fixates on a high resolution display. A potential application of this system is in such areas as reconnaissance, weapon delivery and remote control.

Concerning air-to-air combat, the use of the eyes as a control device for an aircraft fire control system,would give an off-boresight capability, provide for head-out-of-the-cockpit operation and allow quick target acquisition. Generally,eye-control systems, in addition to freeing the hands for other tasks, will offer faster reaction than conventional manual control.

Also, small portable instruments have other potential clinical and psychological applications. For instance,they could be used by doctors to diagnose neurological disorders, and by educational psychologists to diagnose reading disabilities.

CHAPTER 10

Conclusions and Further Remarks

10.1 Conclusions

The objectives of the research reported in this thesis (Section 1.2) and the stipulated features and facilities of a real time microprocessor-based video analysis system have been fully realised through the design and construction of VAS. The implementation of the novel hard-wired edge detector as a preprocessing unit to extract the edge points of a video image and assign coordinates to each point enables the microprocessor to accomplish goal-guided tasks within the context of real time processing.

An outstanding feature of the edge detector in contrast to existing parallel edge detection schemes, is the elimination of the requirement to store the video image and the ease of its hardware realisation. The ability of the edge detector to extract essential edge information from different types of input images has been verified. The accessibility of the three parameters which determine its output (viz, the level of the video signal, the reference voltage of the ADC and the threshold for the central differential) provides the facility of interaction with the operator. This allows him to obtain the desirable or the best edge representation by manipulating controls on the front panel of VAS. In the absence of any formal quantitative measure to indicate the accuracy of edge detectors and the seriousness of missed or spurious edges, the operator has to rely on his own subjective visual evaluation to determine the relevance of the extracted edge formation to the solution of the

problem under investigation. Since the edge detection procedure involves spatial differentiation it is insensitive to reasonable variations in the brightness of the video signal. In addition, it performs a thinning operation and some noise cleaning in the vicinity of true edges by taking into consideration irregularities superimposed by noise on an edge profile.

The video signal relayed to the edge detector is represented by 4-bit samples and as a result the digital video image exhibits the undesirable contouring effect due to coherent quantising errors. Elimination or mitigation of the contouring artifacts can be achieved by restricting the edge extraction process to the part of the video signal which contains the relevant edge information, through proper adjustment of the parameters cited above so as to reduce the quantising errors by decreasing the increments between successive quantising levels in the ADC. In practice, it is not possible to eliminate the contouring artifacts when the whole dynamic range of the 4-bit signal is processed. If normal subjective standards of broadcast quality are to be maintained, then 8 bits per sample are required to produce a picture sufficiently free of quantising noise. The requirement was difficult to satisfy because the hardware needed to implement parallel ADC, using ordinary TTL logic circuitry, is considerable and doubles for each additional bit. The choice of a 4-bit ADC was dictated by economy of hardware consideration.

The hardware realisation of the overall system is cost effective, portable, expandable and reliable. The wide applicability of edge detection as an effective preprocessing function and the inclusion of a general purpose processing unit render the system capable of solving a variety of image processing problems. Once the problem

domain is clearly defined, then the operator can utilise the general purpose programs for storage and display of edge patterns to obtain samples of the input data. This enables him to gain more knowledge about the structure of the image to facilitate the formulation of the solution and optimisation of goal-guided real time programs. The constraints of real time and the limited computational capabilities of the microprocessor are the predominant factors which affect the design of the tracking algorithms.

The problem of tracking rigid circular discs exercising planar motion has been addressed within the context of a low level of understanding by the system and the introduction of simplifying assumptions to render the problem solvable. The plausibility of this approach is justified by the unavailability of general interpretation procedures, complications imposed by the constant change in the appearance of tracked objects and unstationary background/foreground situations.

The tracking experiments were performed using a single or two targets moving in a circular path. The tracking programs have successfully located the position of the centroids of the moving targets and presented a visual display of their tracks in real time. Vector fields representing velocities of the targets are computed using stored motion history data and displayed on the storage oscilloscope. Another tracking program is able to identify and lock to a moving target in a scene which, in addition, contains a stationary target.

VAS has successfully performed all functions essential for eye motion analysis. The ability of the edge detector to extract the edges of the pupillary disc and corneal reflex from the output of

the oculometer has been demonstrated. The software package for on-line acquisition of the relevant optical features of the eye, calculation of its point-of-regard and its display accomplishes its task in real time. The interactive program for statistical data reduction offers the facility of assigning appropriate values to a distance metric and temporal thresholds to compress the raw fixation point data and obtain its salient features. The operator can communicate interactively with a set of subroutine to generate a graphic data base and obtain a pictorial representation of the eye fixation-point scan path using raw or compressed point-of-regard data.

A major contribution of this research is the establishment of VAS as a commercially viable eye motion analysis system. This view has been confirmed through discussions with people working in the fields of psychology and physiology. Two commercial versions of a microprocessor-based eye motion analysis system could be built: one is a general purpose laboratory system and the other is a compact portable system.

The general purpose laboratory system is envisaged to include a subject's compartment in which stimulus patterns are presented, a high resolution storage oscilloscope or a computer graphics terminal with provisions to superimpose eye fixation point tracks on the stimulus pattern. It is recommended that the system should be supported with a repertoire of software packages for calibration and linearisation of the oculometer's output and extensive analysis of different aspects of the eye movement. This system could be used to perform research on eye movements as they pertain to psychological and physiological processes.

The compact portable system could be assembled using miniature

optical and display components. Different types of instruments could be manufactured, each with specially configured hardware and software designed to serve particular purposes. Such specialised instruments have many potential clinical, psychological and human factors studies applications. For instance, it could be used in diagnosis of neurological disorders and reading disabilities, testing of visual acuity, human factors studies relevant to ground and air transportations, etc.

10.2 Suggestions for Further Research

In addition to real time analysis of eye motion, VAS could be used to quantify blood cell motion employing procedures designed by Levine, et al [106]. To facilitate the study of cell movements, they have introduced an automatic picture processing computer-based system for tracking and quantifying the dynamics of blood cell motion, which is of importance to the understanding of the role these cells play in host defence mechanisms. To prove that VAS is capable of performing the same task a review of their technique is given below.

Cells are filmed for approximately 24 hours at the rate of one frame exposure every 20sec. The cells appear on the film as white blobs on a black background (Fig. 10.1). The film is mounted on a computer-controlled film transport unit and each frame is scanned by a TV camera. The video signal is digitised and thresholded to convert the grey-scale image to a binary image of white objects (1 bits) against a black background (0 bits) [107]. The value of the threshold is

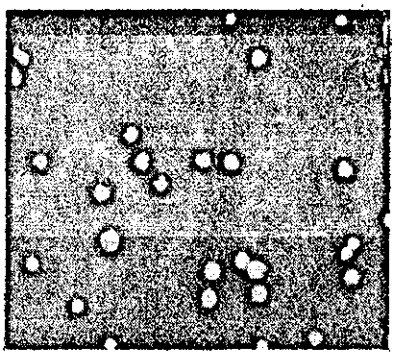


Fig. 10.1: A Typical Frame From the Cine Film Showing Several Cells[106]

derived from a histogram of the grey level image ; then each frame of the binary image is scanned line by line from top to bottom and each line from left to right until reaching the first pixel having binary 1 value. A window is prescribed around this pixel and its boundaries are adjusted such that most of the cell elements are located inside it. After labelling the cell its elements are set to zero and the scanning is resumed.

The design of the tracking program is based on the fact that the maximum displacement of a cell between any two successive frames is from one cell diameter to 1.5 cell diameters. Consequently, for each cell tracked in a frame an appropriate window may be postulated in the next frame. Only objects within this window are examined to match objects in two successive images in order to determine the paths of cells. The global directional movement

tendencies of groups of cells are characterised by a Markov chain model * to determine the probability that the cell population is moving in a particular direction. This information might be of interest in the study of the effect of substances on cell movements, defects in white blood cells migration and the general interaction among cells.

Using the automatic technique, the processing time for tracking and analysing 15 to 25 cells over 300 frames is less than three hours. In contrast, the manual approach based on visual tracking of cell locations takes about two days to record and track the paths of 5 to 15 cells over 200 frames[107].

The system could be improved if the intermediate stage of film recording is replaced by a TV camera observing the cell behaviour directly via a microscope and tracking is performed in real time. A practical laboratory device employing a VAX11/780 minicomputer and a GMR-27 Graphic Television Display System is under development[107].

In the Author's opinion VAS offers a cheap alternative to the proposed computer-based real time system and has the following advantages:-

1) Since the image processed consists of white objects against a black background the hard-wired edge detector can easily extract the relevant edge images thus eliminating the tedious histogram computation.

---

* *In the Markov chain model S, the state spaces, is a set of angles with respect to the X-axis. Angle 2π is partitioned into n parts such that the angles between θ_{i-1} and θ_i constitute state i of the Markov chain. A cell is in state i if it is heading in a direction whose angle with the X-axis is between θ_{i-1} and θ_i.*

2) No need to store the full frame of imagery. VAS can superimpose a window on the input image to isolate single cells and accomplish the matching procedure employing their edges.

3) Being small and portable VAS could be developed into a simple on-bench laboratory apparatus.

Throughout this research emphasis has been laid on real time applications. Another field of application in which VAS could be useful is automatic scene analysis when procedures employed are based on edge detection, contour following and algorithms utilising structural features of objects under investigation. The Author hopes to make enquiries about the applicability of VAS, or an upgraded version of it, in analysis of chromosomes, bacteria and nerve-fibre cross-section.

10.3 Closing Remarks

Since the emergence of TMS 9900, there has been tremendous advancement in the speed, architecture and sophistication of microprocessors. It is generally appreciated that the recently introduced fourth generation of microprocessors (iAPX 432) is in some configurations as powerful as a contemporary midrange mainframe[108]. This suggests upgrading VAS by replacing the TMS 9900 with a more powerful microprocessor to achieve a faster rate of processing and implement more complex pattern recognition techniques. In addition, to lighten the burden of the programmer a better programming method may be introduced. The performance of the edge detector could be enhanced by increasing the number of bits representing the digitised

video signal to 8 by incorporating in the system one of the now commercially available ADC's.

10.4 Notes on Publications

Two papers describing some aspects of this research have been published. The first paper is entitled, "Microprocessor-based Real Time System for Eye Motion Analysis", by O. M. Abdel Gadir and D. J. Quarmby. This paper gives a brief description of the system and has been published in Electronics Letters, IEE, Vol. 17, No. 5, 1981.

The second paper is by the same authors and is entitled, "A Microprocessor-based Real Time Video Analysis System". It includes a brief description of the edge detection technique and a cursory review of the applications of the system. The paper has been published in IERE, Proc. No.49, April 1981.

REFERENCES

1. Skalansky, J., ed. Pattern Recognition, Dowden Hutchington and Ross, 1973.

2. Duda, R. and Hart, P., Pattern Classification and Scene Analysis, Wiley, 1973.

3. Tou, T. and Gonzaley, R., Pattern Recognition Principles, Addison-Wesley, 1974.

4. Andrews, H.C., Introduction to Mathematical Techniques in Pattern Recognition, Wiley, 1972.

5. Watanabe, S., ed., Frontiers of Pattern Recognition, Academic Press, 1972.

6. Cheng, G.C., et al., ed., Pictorial Pattern Recognition, Thompson, 1968.

7. Fu, K.S. and Rosenfeld, A., "Pattern Recognition and Image Processing", IEEE Trans. Comput., Vol. C-25, No.12, pp 1336-1346, Dec. 1976.

8. Bullock, B.L., "Finding Structure in Outdoor Scenes", in "Pattern Recognition and Artificial Intelligence", Chen, C., ed., Academic, 1976.

9. Pratt, W.K., Digital Image Processing, Wiley, 1978.

10. Dwyer, S.J., McLaren, R.W. and Harlow, C.A., "Computer Aided Diagnosis of Breast Cancer from Thermography", in "Pattern Recognition and Artificial Intelligence", Chen, C., ed., Academic, 1976.

11. Tenenbaum, J.M. and Barrow, H.G., "A Paradigm for Integrating Image Segmentation and Interpretation", in "Pattern Recognition and Artificial Intelligence", Chen, C., ed., Academic, 1976.

12. Philbrick, O., "Shape Description with the Medial Axis Transformation", in "Pictorial Pattern Recognition", Cheng, G.C., et al., ed., Thompson, 1968.

13. Connor, D.J. and Limb, J.O., "Properties of Frame-Difference Signals Generated by Moving Images", IEEE Trans. on Comm., Vol.COM22, pp 1564-1575, Oct. 1974.

14. Smith, E.A. and Phillips, D.R., "Automated Cloud Tracking Using Precisely Aligned Digital ATS Pictures", IEEE Trans. on Comp., Vol.C-21, pp 715-729, July, 1972.

15. Thompson, W.E. and Flacks, G.M., "A Structure and Dynamic Mathematical Model of a Real Time Video Tracking System", Proceed. IEEE, Nat. Aerospace and Electronics Conf., May, 1976.

16. Monty, R.A. and Senders, J.W., Eye Movements amd Psychological Processes, Wiley, 1976.

17. Rocca, F. and Zanoletti, S., "Television Bandwidth Compression via Movement Compensation", IEEE Trans. on Comm., Vol.Com-20, pp 960-965, Oct. 1972.

18. Martin, W.N. and Aggarwal, J.K., "Survey: Dynamic Scene Analysis", Computer Graphics and Image Processing, Vol.7, pp 356-374, 1978.

19. Ulstad, M.S., "An Algorithm for Estimating Small Scale Differences Between Two Digital Images", Pattern Recognition, Vol.5, pp 323-333.

20. Limb, J.O. and Murphy, J.A., "Estimating the Velocity of Moving Images in Television Signals", Computer Graphics and Image Processing, Vol.4, pp 311-327, 1975.

21. Gilbert, A.L. et al., "A Real-Time Video Tracking System", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-2, No.1, pp 47-56, Jan. 1980.

22. Aggarwal, J.K. and Duda, R.O., "Computer Analysis of Moving Polygon Images", IEEE Trans. Computers, Vol.C-24, pp 966-976, Oct. 1975.

23. Chomsky, N., "Three Models for the Description of Language", IRE Trans. Inform. Theory, Vol.IT-2, No.3, pp 113-124, Sept. 1956.

24. Unger, S.H., "Pattern Recognition and Detection", Proc. IRE, pp 1737-1752, Oct. 1959.

25. Kruse, B., "A Parallel Picture Processing Machine", IEEE Trans. on Comp., Vol.C-22, No.12, pp 1075-1087, Dec. 1973.

26. Hawkins, J.K., "A Parallel Computer Organisation and Mechanisation". IEEE Trans., Electron. Comp., Vol.EC-12, No.3, pp 251-262, 1973.

27. Floating Point Systems, Inc., The Age of Array Processing is Here, Product Information, 1978.

28. CSP Inc., An Introduction to Array Processing with Snap - 11, Technical Report, 1975.

29. CPS Inc., An Introduction to MAP Series Models 100, 200 and 300, Preliminary Technical Report, 1975.

30. Stockman, G.C. and Kanal, L.N., "Interactive Screening of Reconnaissance Imagery", in "Pattern Recognition and Artificial Intelligence", Chen, C., ed., Academic, 1976.

31. Vernon, M.D., The Psychology of Perception, Penguin, 1962.

32. Schmidt, R.F., Fundamentals of Neurophysiology, Springer-Verlag, 1978.

33. Noback, C.R. and Demarset, R.J., The Nervous System: Introduction and Review, McGraw-Hill, 1972.

34. Wooldridge, D.E., The Machinery of the Brain, McGraw-Hill, 1963.

35. Fukushima, K., et al., "An Electronic Model of the Retina", Proc. IEEE, Vol.58, No.12 December 1970.

36. Fukushima, K., "A Model of Associative Memory in the Brain", Kybernetik 12, pp 58-63, Springer-Verlag, 1973.

37. Fukushima, K., "Cognitron: A Self-Organising Multilayered Neural Network", Biol. Cybernetics 20, pp 121-136, Springer-Verlag, 1975.

38. Hubel, D.H. and Wiesel, T.N., "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex", J.Physiol. (1962), 160, pp 106-154.

39. Bullock, B.L., "The Performance of Edge Operators on Images with Texture", Hughes Aircraft Company Technical Report, October 1974.

40. Davis, L.S., "A Survey of Edge Detection Techniques", Computer Graphics and Image Processing, Vol.4, pp 248-270, 1975.

41. Fries, R.W. and Modestino, J.W., "An Empirical Study of Selected Approaches to the Detection of Edges in Noisy Digitised Images", Report by the Communication and Information Processing Group, Dept. of Electrical and System Engineering, R.P.I.Troy, New York, March 1977.

42. Sakai, T. and Nagao, M., "Processing of Multilevel Pictures by Computer - The Case of Photographs of Human Face", Systems, Computers and Control, Vol.2, No.3, pp 47-54, 1971.

43. Persoon, E., "A New Edge Detection Algorithm and its Application in Picture Processing", Computer, Graphics and Image Processing, Vol.5, No.4, pp 425-447, 1976.

44. Wechsler, H. and Kidode, M., "A New Edge Detection Technique and its Implementation", IEEE Trans. Systems, Man and Cybernetics, Vol.SMC-7, No.12, pp 827-831, 1977.

45. Rosenfeld, A. and Thurston, M., "Edge and Curve Detection for Visual Scene Analysis", IEEE Trans. Comput., Vol.C-20, pp 562-569, 1971.

46. Rosenfeld, A., Thurston, M. and Lee, Y.H., "Edge and Curve Detection: Further Experiments", IEEE Trans. Comput.(Special Issue on Two-Dimensional Digital Signal Processing), Vol.C-21, p 677-715, 1972.

47. Rosenfeld, A., "A Note on Automatic Detection of Texture Gradients", IEEE Trans. on Comput., Vol.C-24, pp 988-991, 1975.

48. Fram, J.R. and Deutsh, E.S., "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance", IEEE Trans. on Comput., Vol.C-24, pp 616-628, 1975.

49. Nudd, G.R.and Nygaard, P.A., "Demonstration of a C.C.D. Image Processor for Two-Dimensional Edge Detection", Electronics Letters, Vol.14, No.4, 1978.

50. "Technical Reference Book", IBA Technical Review, IBA, 1977.

51. Pulse Code Modulation of Video Signals: Subjective Study of Coding Parameters, BBC Research Dept., Report No. 1971/40.

52. Moore, T.A., "Digital Video: Number of Bits per Sample Required for Reference Coding of Luminance and Colour-Difference Signals", BBC Research Dept., Report No. 1974/42.

53. Moore, T.A., "Digital Video: A Theoretical Assessment of the Quantising Noise Spectra Arising from a Change in the Sampling Frequency of PAL Signals", BBC Research Dept., Report No.1976/10.

54. Downing, O.J. and Thorpe, P.L.M., "Proposed Technique for Reduction of Contouring in Quantised (Monochrome) Image Signals", Electronics Letters, Vol.15, no.1, 1979.

55. Lenk, J.D., "Handbook of Microprocessors, Microcomputers and Minicomputers", Prentice-Hall, 1979.

56. Kraft, G.D. and Tog, N.W., Mini/Microcomputer Hardware Design, Prentice-Hall, 1979.

57. Klingman, E.E.,"Microprocessor System Design", Prentice Hall, 1977.

58. Technico Inc., "Technico Super Starter Manual", Technico Inc., Colombia, U.S.A., 1976.

59. Texas Instruments Inc., "TMS9900 Microprocessor Data Manual", (Texas Instruments, Houston, 1976).

60. Texas Instruments Inc., "TMS9900, Assembly Language Programmer's Guide", (Texas Instruments, Houston, 1976).

61. Texas Instruments Inc., "The Memory and Microprocessor Data Book", (Texas Instruments, Houston, 1976).

62. Texas Instruments Inc., "9900 System Handbook", (Texas Instruments, Houston, 1976).

63. Artwick, B.A., "Microcomputer Interfacing", Prentice-Hall Inc., 1980.

64. Temes, G.C. and Mittra, S.K., "Modern Filter Theory Design", John Wiley and Son, 1973.

65. Temes, G.C. and Gyi, M., "Design of Filters with Arbitrary Passband and Chebyshev Stopband Attenuation", IEEE International Convention Record, Vol.15, Pt.5.

66. Guillemin, E.A., "Synthesis of Passive Network", John Wiley, New York, 1957.

67. Hoeschele, D.F., "Analog to Digital/Digital to Analog Conversion Techniques", John Wiley, 1968.

68. Peatman, J.B., "Microcomputer-Based Design", McGraw-Hill, 1977.

69. O'Rourke, J. and Badler, N.J., "Model-Based Image Analysis of Human Motion Using Constraint Propagation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-2, No.6, pp 522-536, Nov. 1980.

70. Roach, J.W. and Aggarwal, J.K., "Determining the Movement of Objects from Sequence of Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. AMI-2,No.6, pp 554-562, Nov. 1980.

71. Tsuji, S., Osada, M. and Yachida, M., "Tracking and Segmentation of Moving Objects in Dynamic Line Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. AMI-2, No.6, pp 516-522, Nov. 1980.

72. Potter, J.L., "Motion as a Cue to Segmentation", Milwaukee Symposium on Automatic Control, pp 100-104, 1974.

73. Roach,J.W. and Aggarwal, J.K., "Computer Tracking of Objects Moving in Space", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-1, No.2, pp 127-134, April, 1979.

74. Badler, N., "Three-Dimensional Motion from Two-Dimensional Picture Sequences", Proc. 2nd Int. Joint Conf. on Pattern Recognition, pp 157-161, August 1974.

75. Wallace, T.P. and Mitchel, O.R., "Analysis of Three-Dimensional Movement Using Fourier Descriptors", IEEE Trans. on Pattern Analysis and Machine Intelligence", Vol.PAMI-2, No.6, pp 583-588, Nov. 1980.

76. Thompson, W.B., "Combining Motion and Contrast for Segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No.6, pp 543-549, Nov. 1980.

77. Jacobus, C.J., Chien, R.T. and Selander, J.M., "Motion Detection and Analysis by Matching Graphs of Intermediate-Level Primitives", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No.6, pp 495-510, Nov. 1980.

78. Yarbus, A.L., Eye Movements and Vision, Plenum Press, New York, 1967.

79. Gulf & Western Applied Science Laboratories, "Eye-Trac", (G & W Applied Science Laboratories, Massachusetts, 1978).

80. Whittaker Corporation, "Eye View Monitor and TV Pupillometer System", (Whittaker Corporation, Massachusetts, 1974).

81. Lambert, R.H., Monty, R.A. and Hall, R.J., "High-speed Data Processing and Unobtrusive Monitoring of Eye Movements", Behaviour Research Methods and Instrumentation, Vol. 6(6), 1974, pp 525-530.

82. Davies, P.W., Lutz, J.S., Warner, A. and Fannini, A.A., "Design and Construction of a Portable Oculometer for use in Transportation Oriented Human Factors Studies", U.S. Dept. of Transportation, Report No. DOT-TSC-OST-71-13, August 1971.

83. Merchant, J., Morrissette, R. and Porterfield, J.L., "Remote Measurement of Eye Direction Allowing Subject Motion Over One Cubic Foot of Space", IEEE Trans. on Biomedical Eng., Vol. BME-21, No.4, July 1974, pp 309-317.

84. Ville, C.A., Biology, W.P.Saunders Co., 1959.

85. Weisz, P.B., The Science of Biology, McGraw-Hill, 1967.

86. James, P.C., "Introduction to Physiological Optics", Southall Dover Publications Inc., New York, 1961.

87. Hubel, D.H. and Wiesel, T.N., "Receptive Fields of Single Neurones in the Cat's Striate Cortex,J.Physiol. (1959),148, pp 574-591.

88. Hubel, D.H. and Wiesel, T.N., "Receptive Fields and Functional Architecture of Monkey Striate Cortex", J. Physiol. (1968), 195, pp 215-243.

89. Blakemore, C. and Campell, F.W.,"On the Existence of Neurones in the Human Visual System Selectively Sensitive to the Orientation and Size of Retinal Images", J. Physiol . (1969), 203, pp 237-260.

90. Campel, C.J., et al, Physiological Optics, Harper and Row, 1974.

91 Best, C. and Taylor, N., The Living Body, A Text in Human Physiology, Chapman and Hall Ltd., 1960.

92. Young, L.R. and Sheena, D., "Survey of Eye Movement Recording Methods", Behaviour Research Methods and Instrumentation, Vol. 7(5), 1975, pp 397-429.

93. Carpenter, R.H.S., Movements of the Eye, Pion, London 1977.

94. Cornsweet, T.N. "The Purkinje-Image Method of Recording Eye Position", ed. Monty, R.A. and Senders,J.W., Eye Movements and Psychological Processes, Lawrence Erlbaum Associates, Publishers, New Jersey, 1976.

95. Gale, A.G., "An Inexpensive Oculometer for Human Factors Research", Departmental Report, Dept. of Human Sciences, Loughborough University of Technology, 1976.

96. Lambert, R.H., "Recent Developments in High-Speed Data Processing and Unobtrusive Monitoring of the Eye", ed., Monty, R.A. and Senders, J.W., Eye Movements and Psychological Processes Lawrence Erlbaum Associates, Publishers, New Jersey, 1976.

97. Anliker, J., "Eye Movements: On-Line Measurement, Analysis and Control", ed., Monty, R.A. and Senders, J.W., Eye Movements and Psychological Processes, Lawrence Erlbaum Associates, Publishers, New Jersey, 1976.

98. Monty, R.A. and Senders, J.W., "Eye Movements and Psychological Processes", Lawrence Erlbaum Associates, New Jersey, 1976.

99. Albutt, M.G., et al., "Eye Movement Recording by On-Line Analysis of Television", Technical Report,Department of Psychology, University of Oxford.

100. Corby, N. R., "The Design and Development of a Real-Time, On-Line, Interactive, Minicomputer Based Eye Motion Measurement and Analysis System", Doctoral Thesis, R.P.I., Troy, New York, 1976.

101. Morse, S.D., et al., "Intel Microprocessors-8008 to 8086", Computer, IEEE Computer Society, October 1980.

102. Johnson, R.C., "Z9000 Chip to Aim at Easing the Use of Virtual Memory", Electronics International, McGraw-Hill, 23rd October 1980.

103. Rogers, D.F. and Adams, J.A., "Mathematical Elements for Computer Graphics", McGraw-Hill, 1976.

104. Mackworth, N.H. and Morandi, A.J., "The Gaze Selects Information Detail Within Pictures", Perception and Psychophysics, Vol. 2., 1967.

105. Merchant, J., Morrissette, R. and Porterfield, J.L., "Aerospace Medical Research Laboratory/Honeywell Remote Oculometer", Xeroxed type-script received on request from Honeywell Inc., Radiation Center, 2 Forbes Road, Lexington, Massachusetts 02173, Undated.

106. Levine, M.D., et al., "The Quantification of Blood Cell Motion by a Method of Automatic Digital Picture Processing", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 5., pp.444-450, 1980.

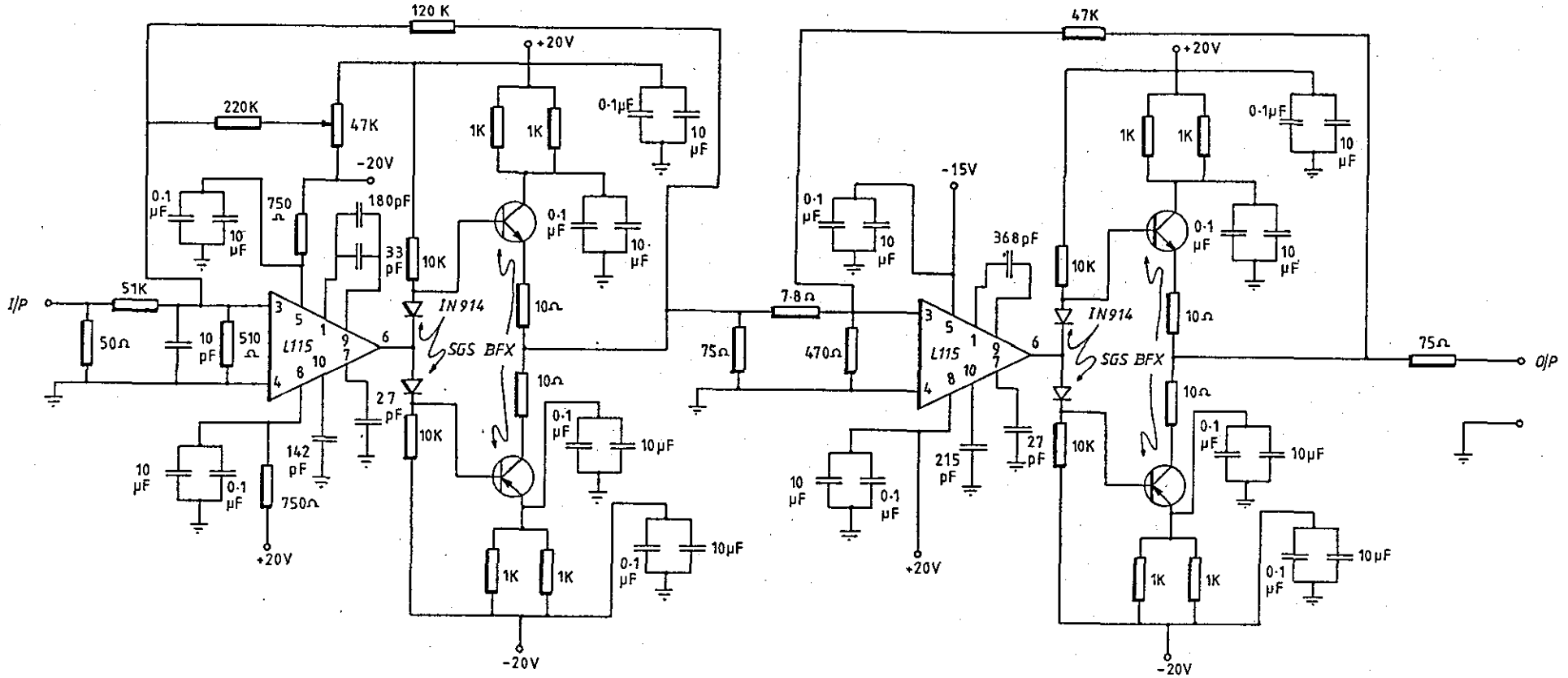
107. Knoll, A., "A Real Time System for Tracking and Quantifying Blood Cell Motion", Dept. of Electrical Engr., McGill University Montreal, P.Q., Canada, Report No.79-12R, June 28, 1979.

108. Morabe, D., "Microprocessor Architectures: Ten Years of Development", Electronics and Power, IEE, Vol.27, No.3, pp. 214-221, 1981.

THE APPENDIX

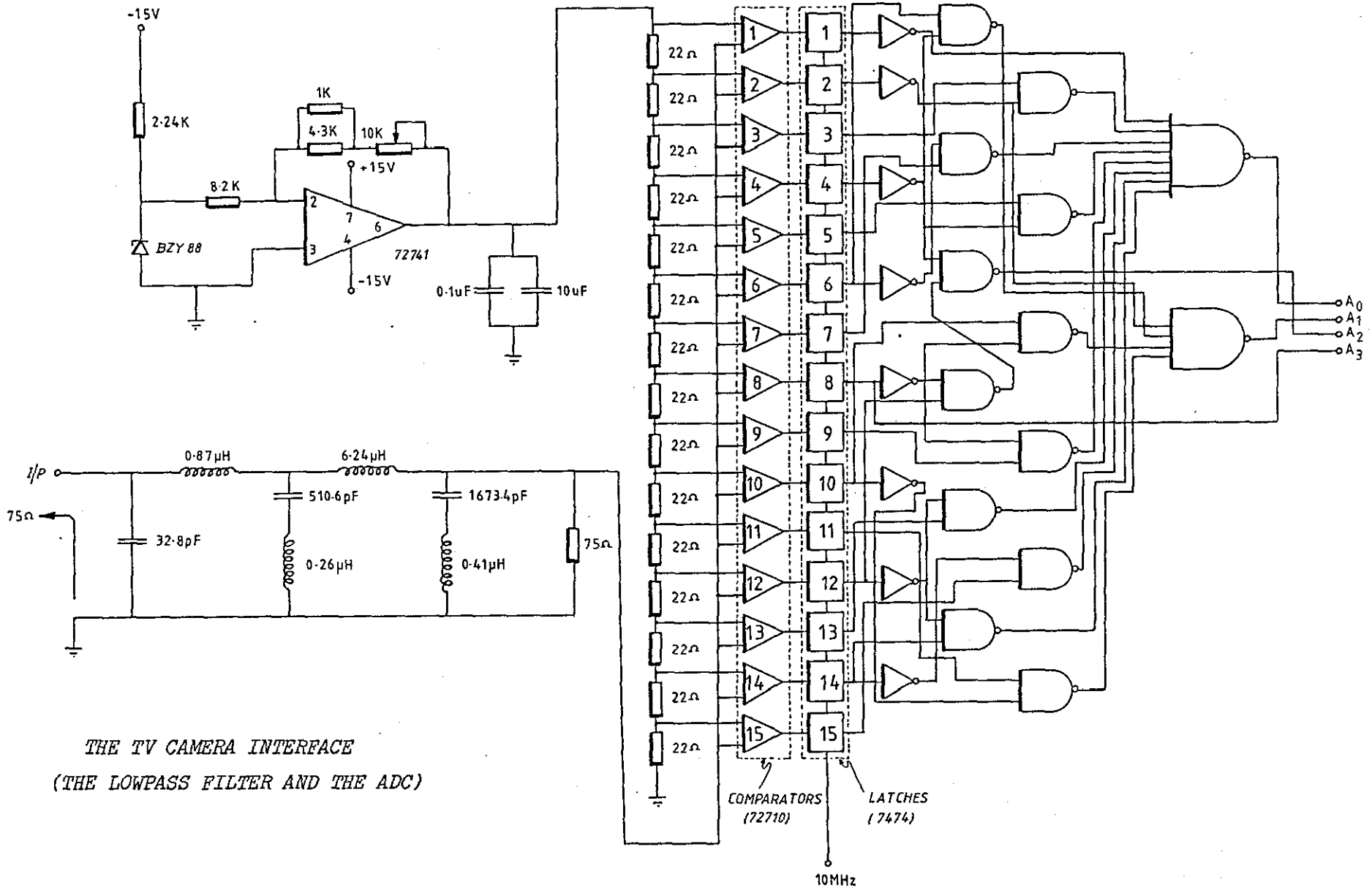
FULL CIRCUIT DIAGRAMS

OF VAS



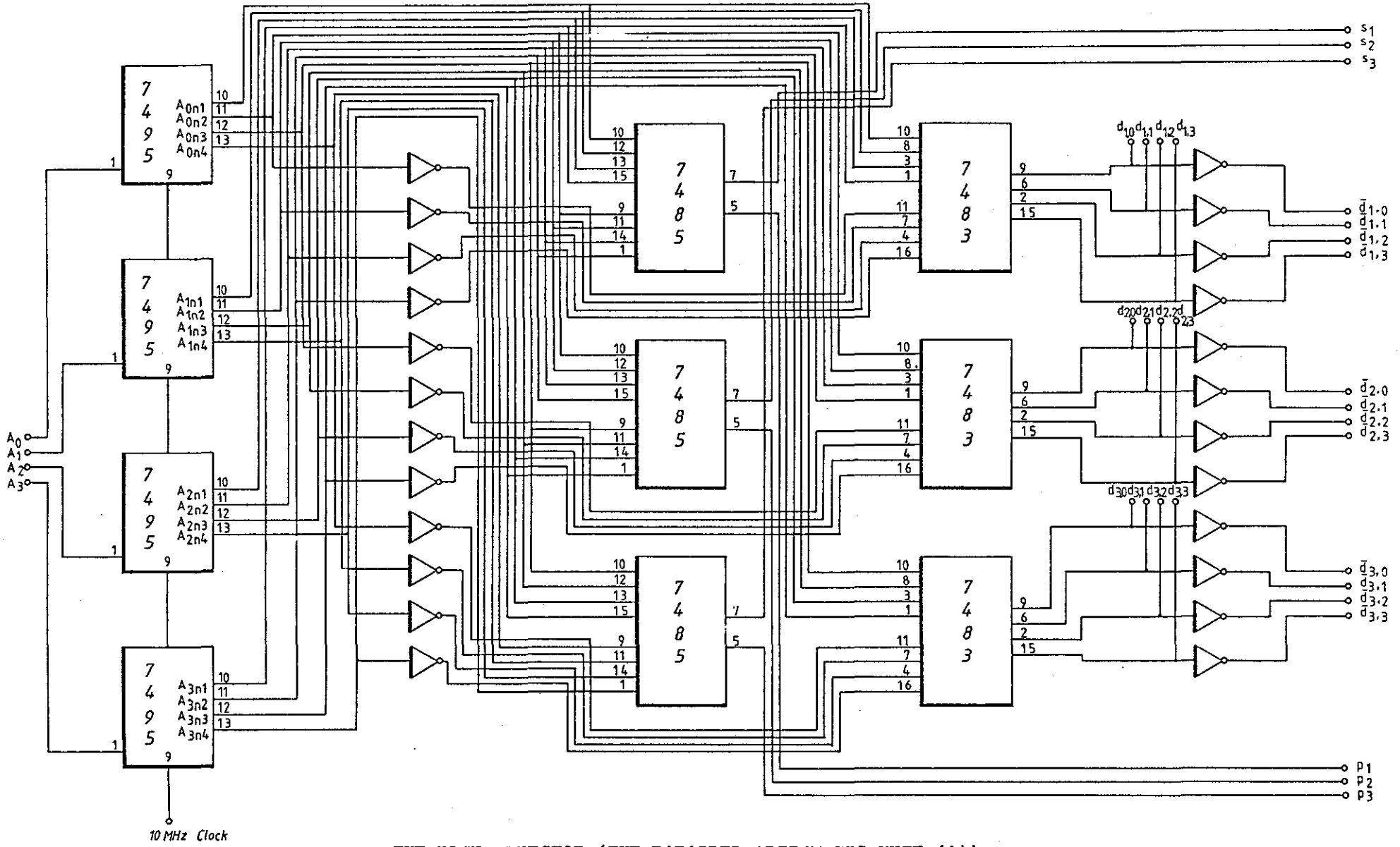
THE TV CAMERA INTERFACE

(THE VIDEO AMPLIFIER)

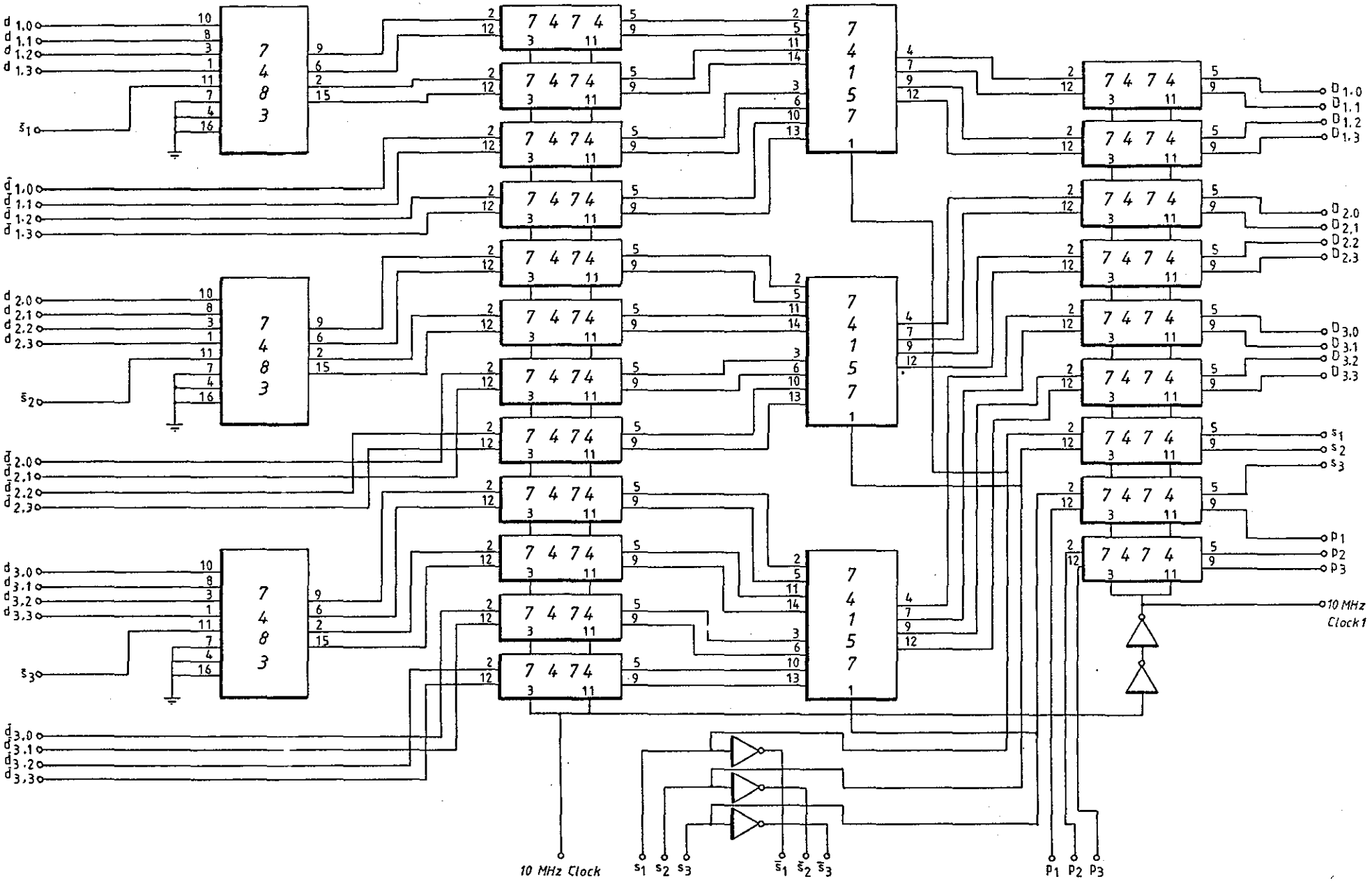


THE TV CAMERA INTERFACE

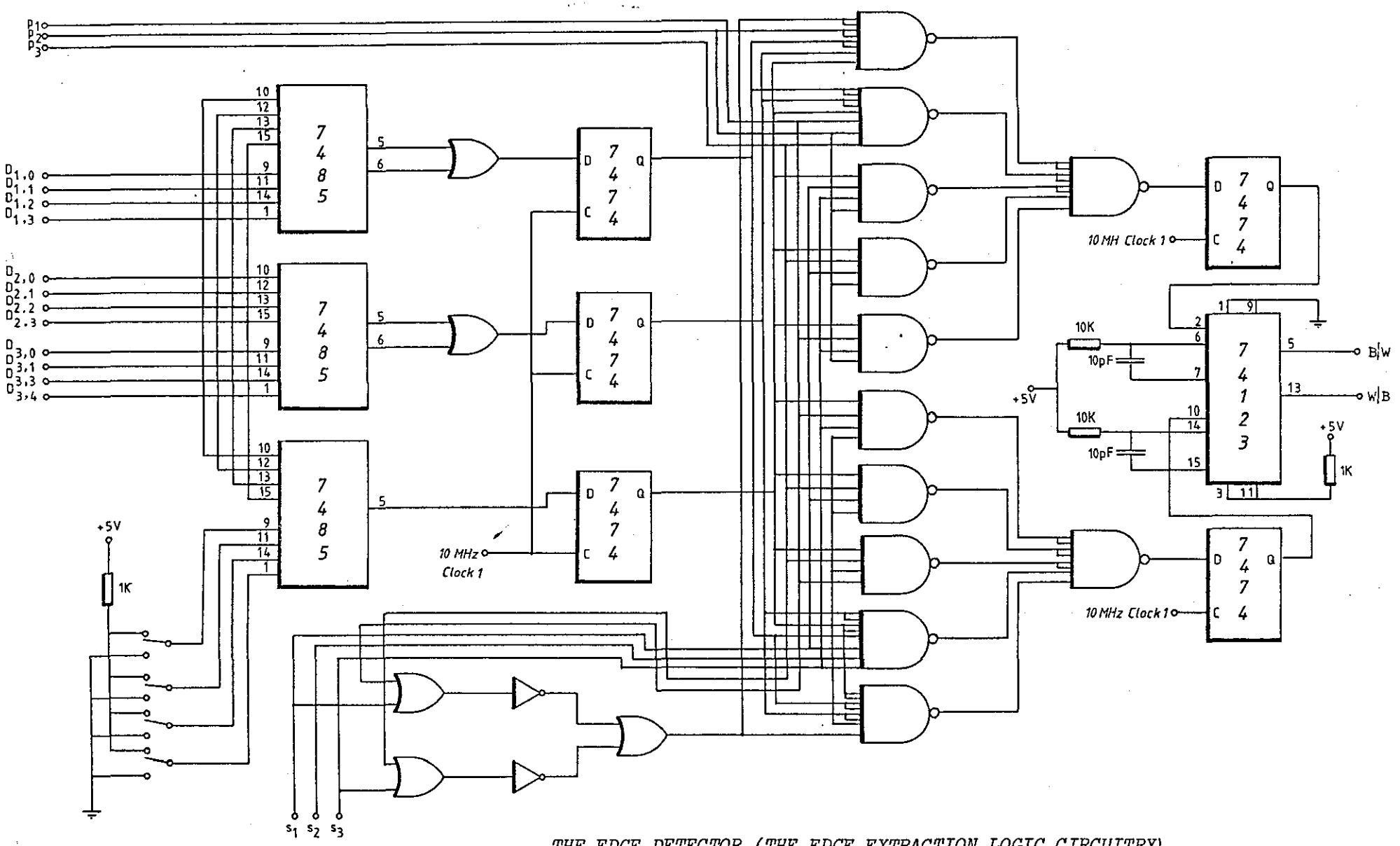
(THE LOWPASS FILTER AND THE ADC)



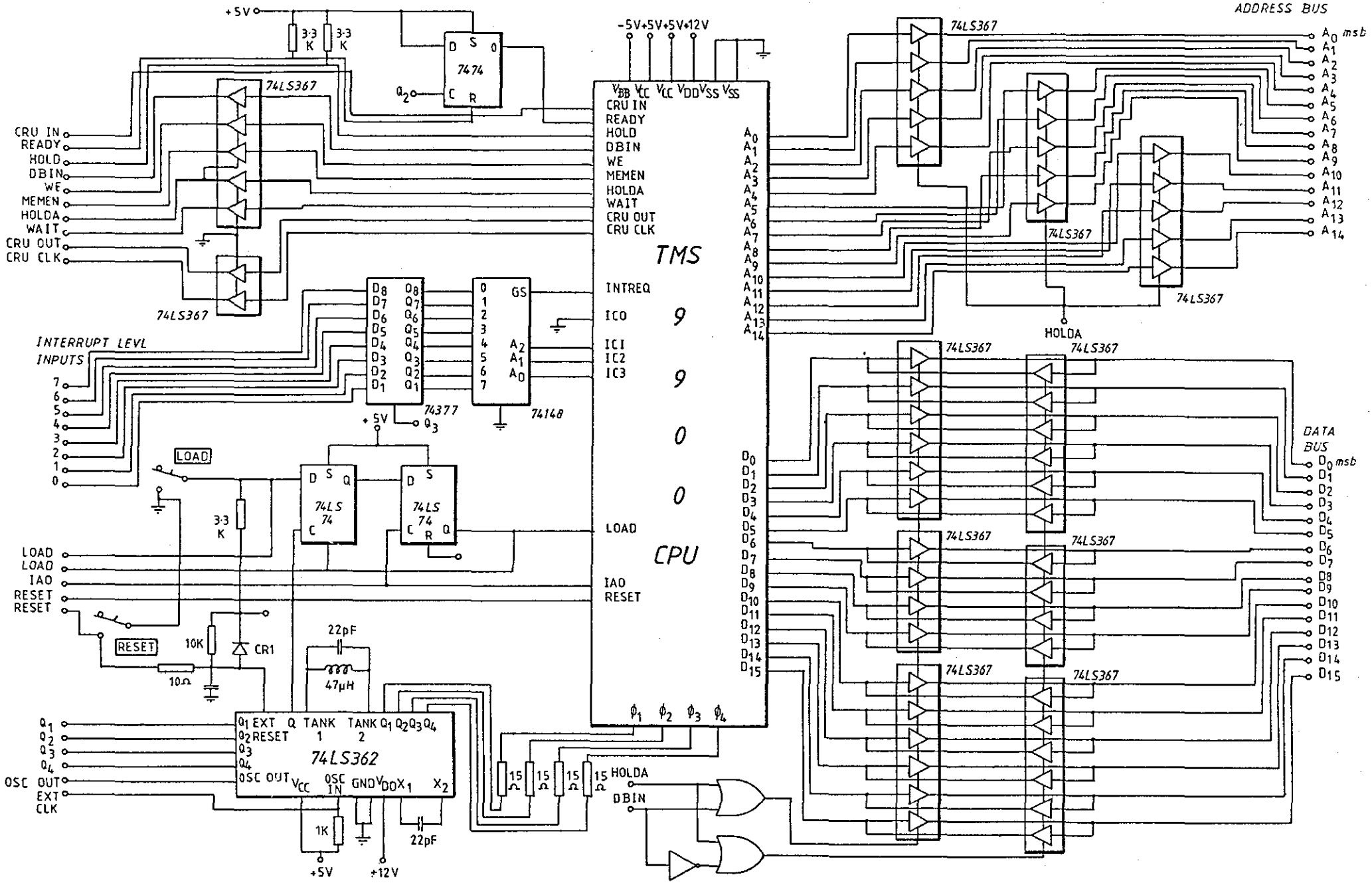
THE EDGE DETECTOR (THE PARALLEL ARITHMETIC UNIT (A))



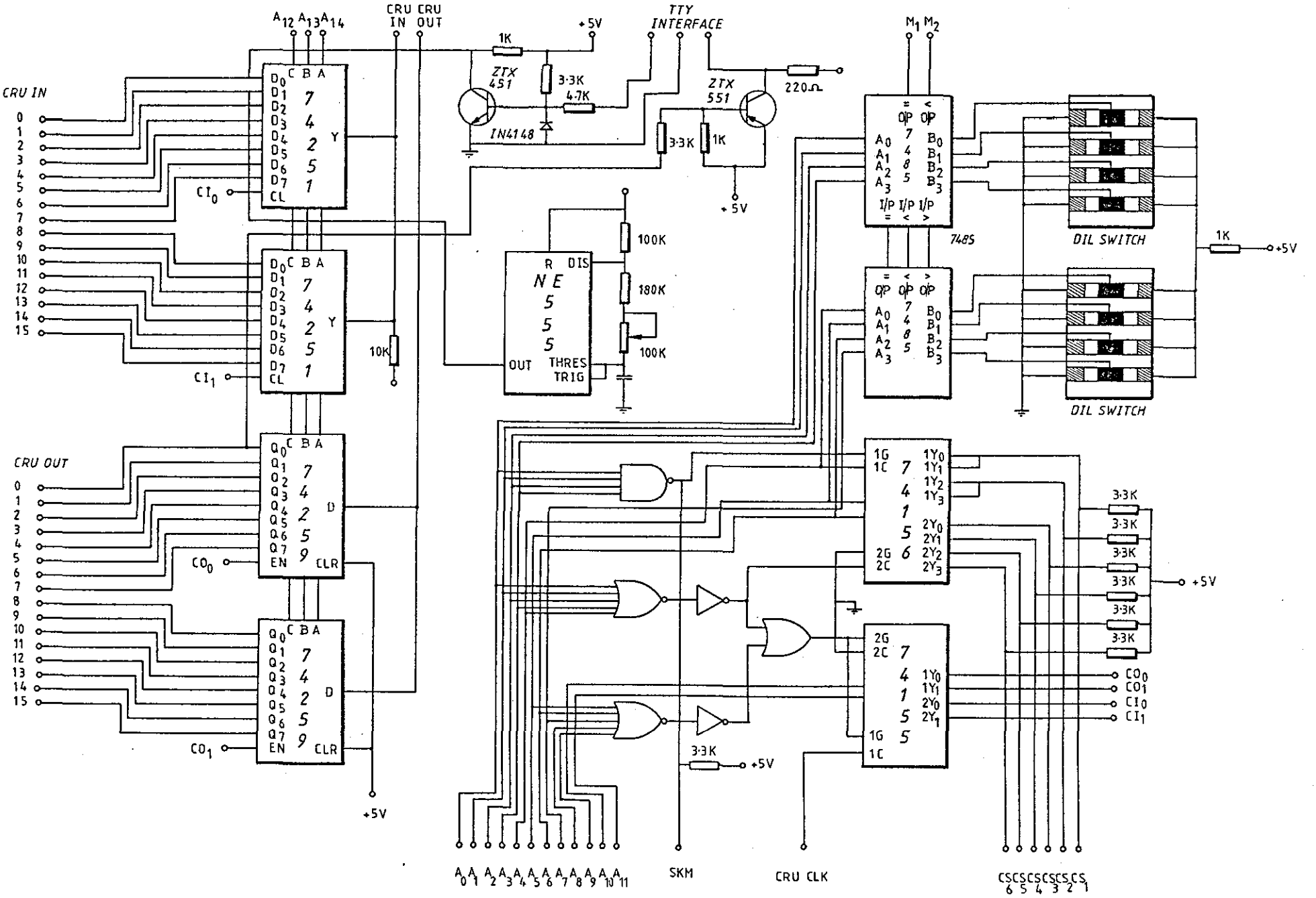
THE EDGE DETECTOR (THE PARALLEL ARITHMETIC UNIT (B))



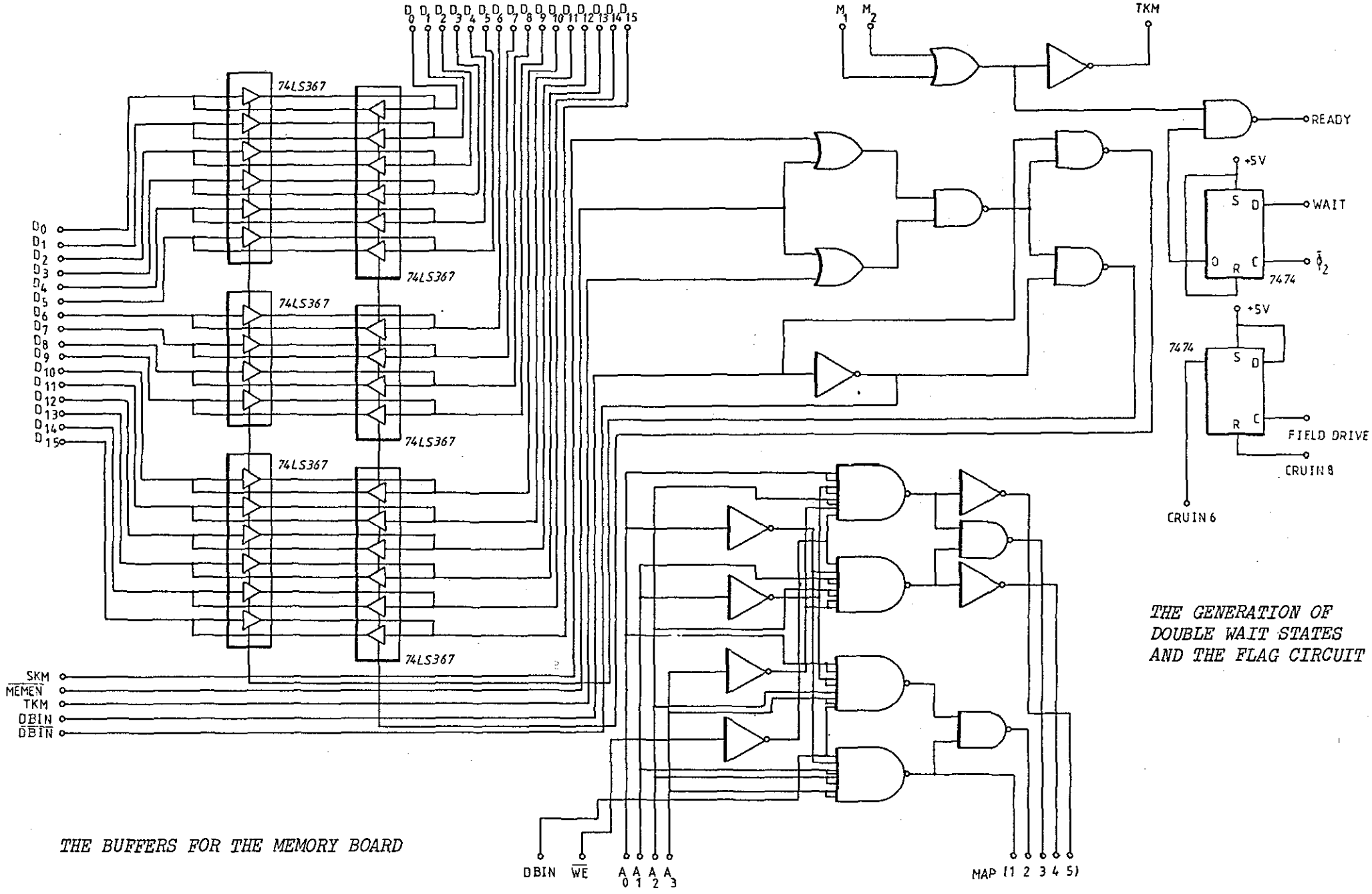
THE EDGE DETECTOR (THE EDGE EXTRACTION LOGIC CIRCUITRY)



THE BUFFERED TMS 9900 CPU AND CONTROL



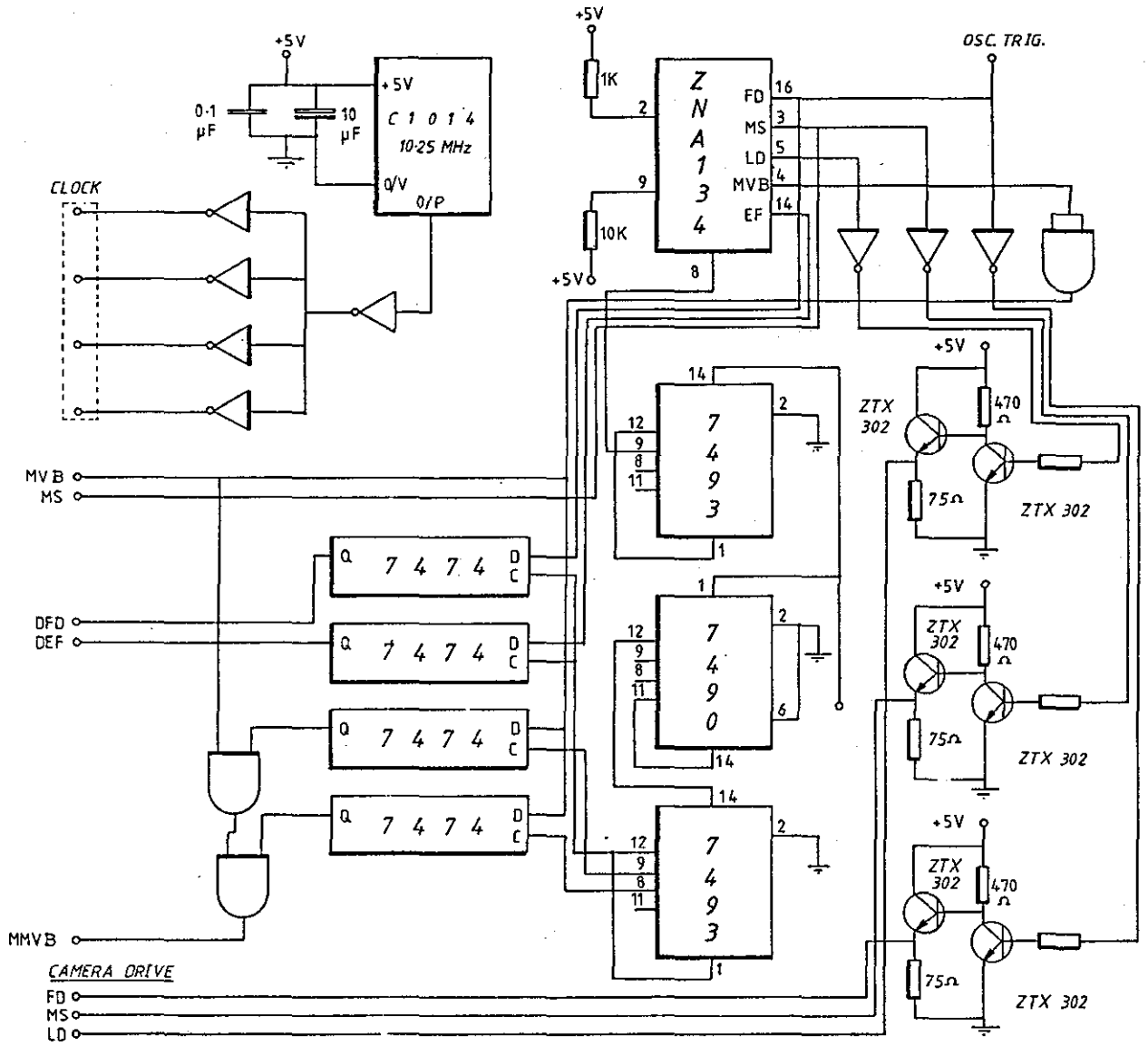
THE CRU INTERFACE AND ADDRESS DECODERS



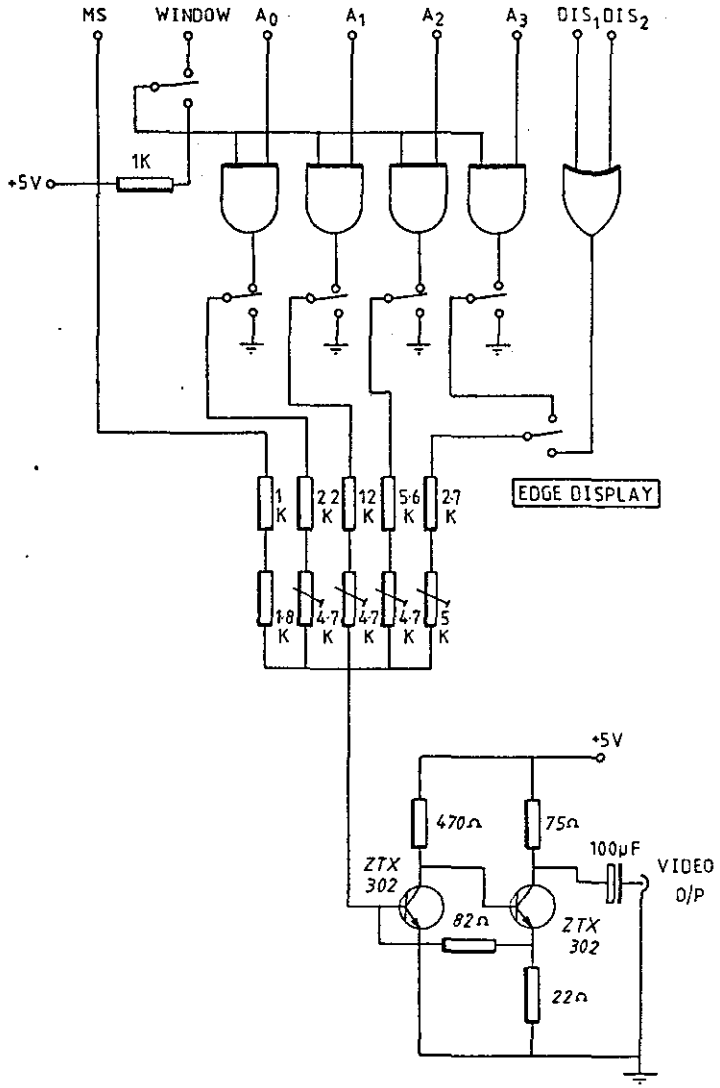
THE BUFFERS FOR THE MEMORY BOARD

THE GENERATION OF DOUBLE WAIT STATES AND THE FLAG CIRCUIT

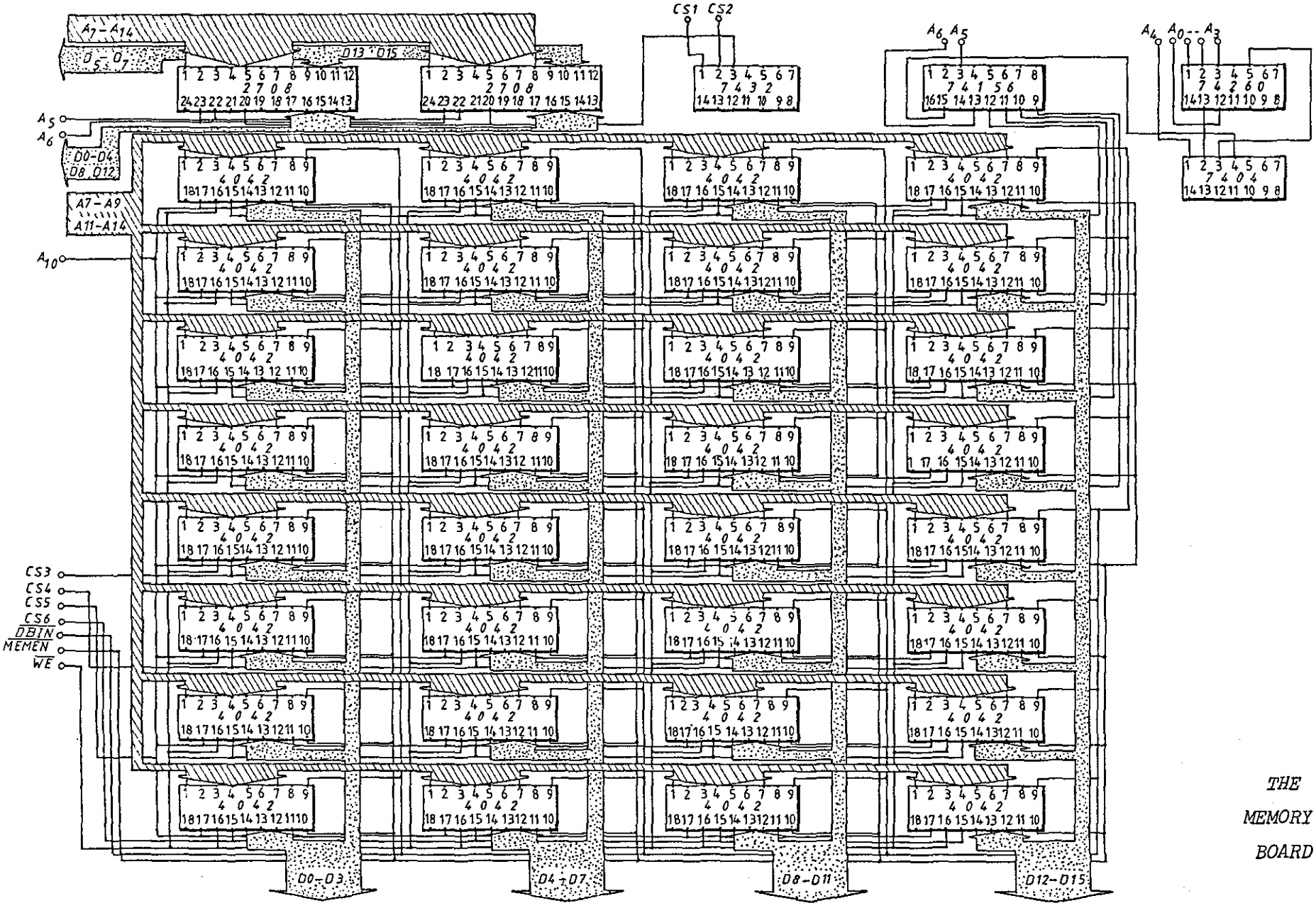
THE DECODERS FOR MEMORY MAPPED I/O



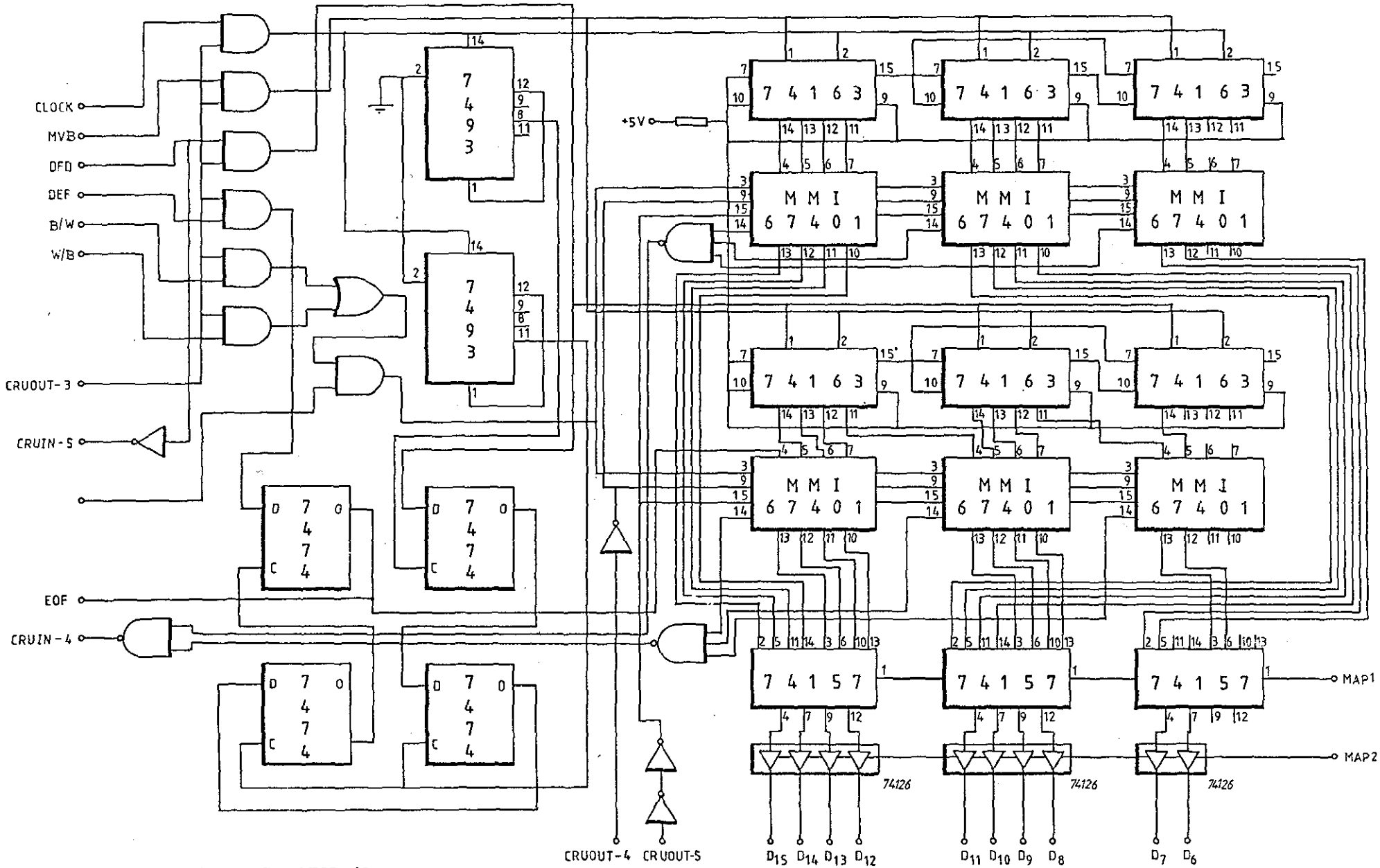
THE CLOCK GENERATOR AND CAMERA DRIVE



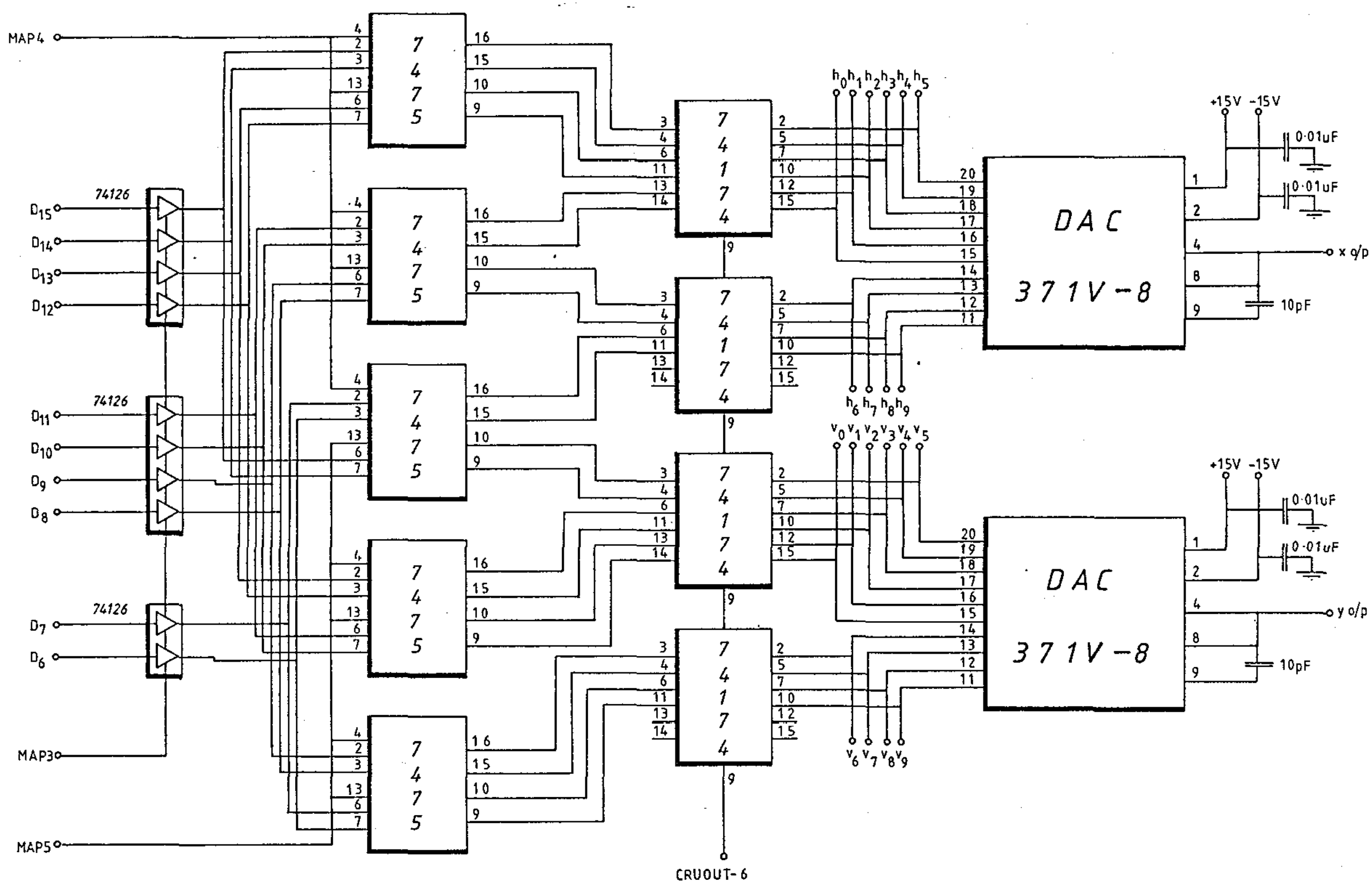
THE DISPLAY UNIT-1 INTERFACE



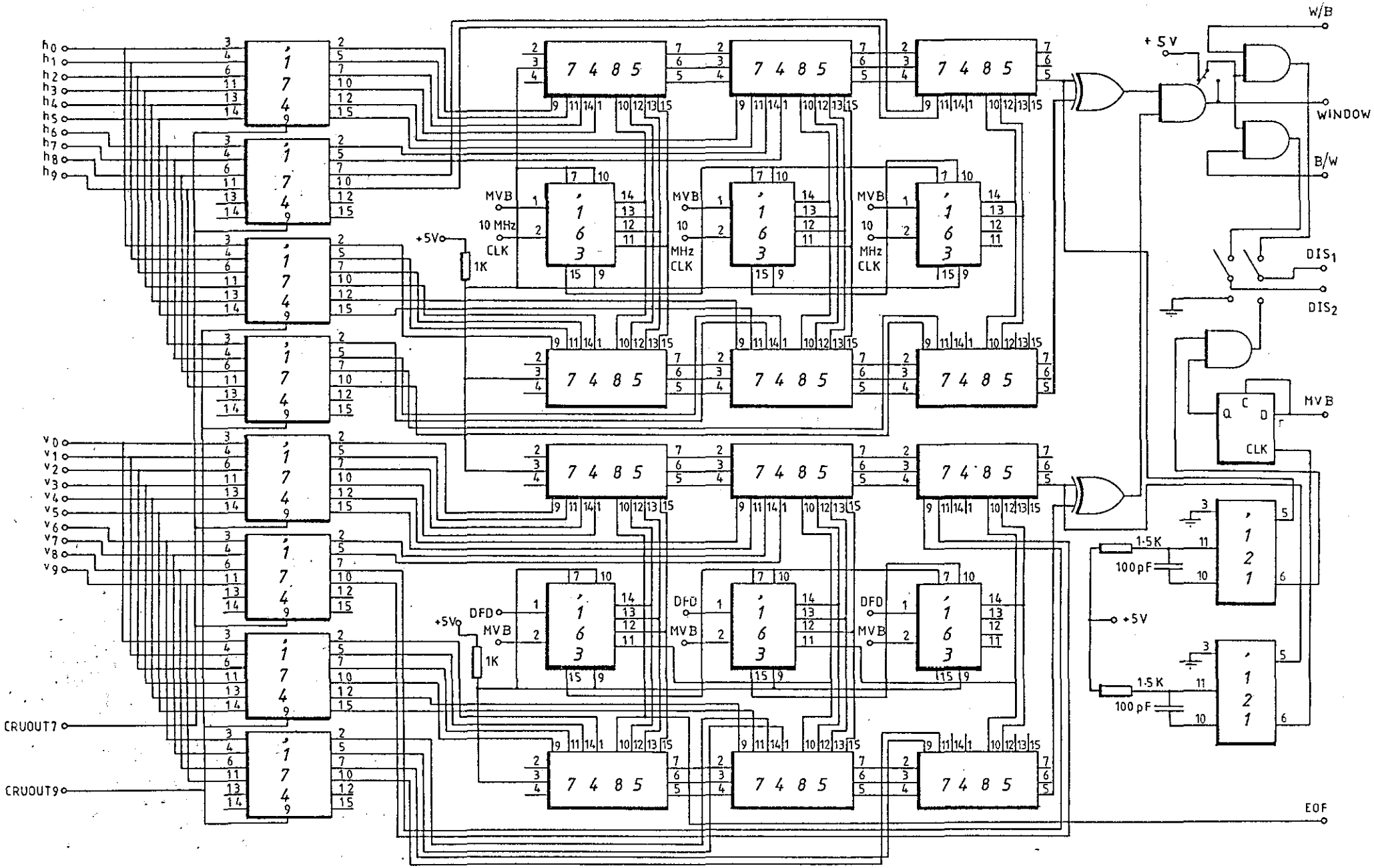
THE MEMORY BOARD



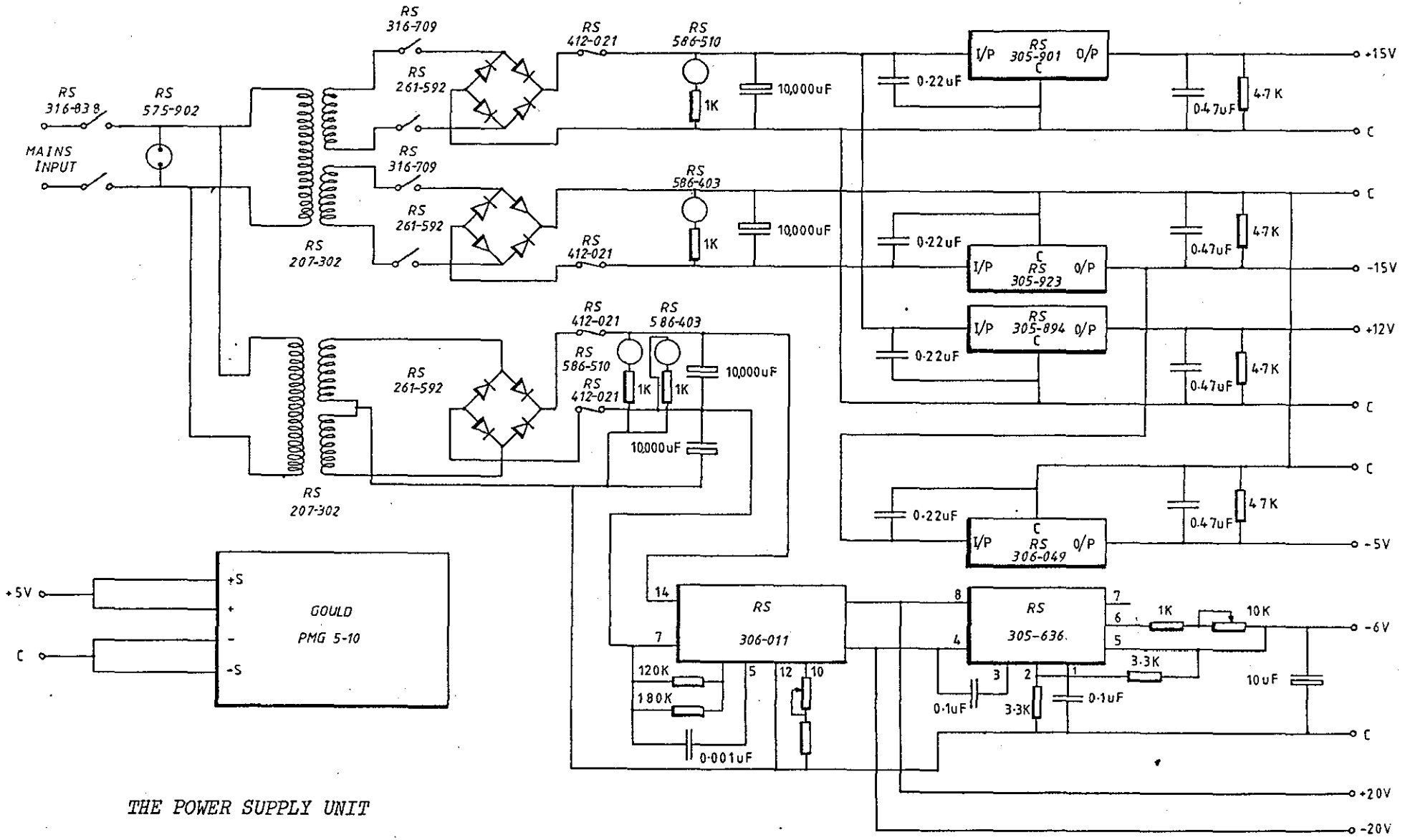
THE COORDINATE GENERATOR AND INTERFACE



THE DISPLAY UNIT-2 INTERFACE



THE WINDOW GENERATOR



THE POWER SUPPLY UNIT