IMAGE PROCESSING AND PATTERN RECOGNITION FOR INDUSTRIAL ROBOTIC VISION

by

Fereydoun Maali, MSc, DIC, CEng, MIEE

November 1983

A thesis submitted for the degree of Doctor of Philosophy of the University of London.

Department of Mechanical Engineering,
Imperial College of Science and Technology,
London, S.W.7.

ABSTRACT

A methodology for the design and development of industrial vision systems is presented. This rests on a nighly modular and interactive image analysis package which is microcomputer based and has been implemented in assembly language. The package is named CAVDEP (Computer Aided Vision DEvelopment Package). It provides a wide assortment of routines for detection, location, identification, and perception of orientation of multiple targets in a multi-class environment.

With its microcomputer based hardware, CAVDEP forms a low cost interactive image analyser that is distinct from similar systems by addressing almost all aspects of the robotic vision problem in a balanced manner. Classification, a less emphasised area in similar interactive systems, is an important aspect of CAVDEP.

Derivation of a compact and efficient parametric sequential classifier is reported. This theoretical contribution entails extension of the Wald's Sequential Probability Ratio Test without the restrictive assumptions of a similar algorithm appearing in standard texts.

The work then extends to the design of dedicated vision systems for specific industrial applications. A major area of investigation has been the design of a vision system for automated handling of certain engine components (i.e.

bearing caps) in the automotive industry. Here a microprocessor based vision system in conjunction with a 3-axes transfer gantry is employed for the transfer of components between various stations and component palletisation. The vision system in this case can also perform pallet positioning and effect automatic positional alignment of the gantry arm.

Later the hypothetical problem of recognition of the same engine components when presented on a conveyor is addressed. Here a new feature based on orthogonal projections of the image is derived and its adequacy demonstrated through simulation. Expressions are also derived from the projection domain for orientation of component sets, and a method is suggested for removal of the existent ambiguities, usually present in computation of orientation.

Finally, the theory of a new vision-based technique for use in CNC lathe has been developed. The technique primarily aims at automation of tool presetting and identification, and extends to certain tool wear measurements.

ACKNOWLEDGEMENTS

The author expesses his gratitude to his supervisor Dr. C.B. Besant for his guidance and advice throughout this research.

He thanks all his associates specially Dr. S.K. Khurmi and Dr. G. Kartsounis for their assistance in the initial phase of this project. He also thanks Dr. A. Pak and Dr. J.W. Mason for their constructive comments on the first draft of this thesis.

The part of R.D.Projects Ltd. in hardware implementation and installation of the vision system, presented in chapter 4 is acknowledged. Special thanks are extended to their able engineer Mr. C. Backhouse.

He thanks Dr. A. Pak and Mr. L. Daneshmend for making available their extensive knowledge of machine tool operation for the development reported in chapter 6

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

$A = \{a_{ij}\}$	an unprocessed image, at the input of an operation.
AA	area.
a	the grey level associated with a pixel.
a_{ij}	a pixel (a picture element)
CCTV	Closed Circuit TV.
CCU	Camera Control Unit.
CNC	Computer Numerical Control
CP	compactness.
CS	a column sum vector in orthogonal projections.
cs_j	a column sum element along the jth column in orthogonal projections.
$D(\underline{\phi})$	decision function

D_i	those values of $D(\underline{\phi})$ which assigns the incident pattern to class ω_i.
DAC	Digital to Analogue Convertor.
$E[\cdot]$	mean of (.).
FIFO	First IN FIRST OUT memory.
FOV	Field Of View.
f	focal length of a lens.
$f(x_o, y_o)$	irradiant flux for monochromatic stationary images
$f(x_o,y_o,t,\lambda)$	multidimensional irradiant flux
G	depth of the critical strip in tool wear measurement
g	width of the critical strip in the tool wear measurement
g(x,y)	image function
grad.	gradient (image), associated with a pixel.

H — neighbourhood operator.

h — an element of H, above.

$h(\cdot)$ — point spread function.

ICCC — Imperial College Computer Center.

$i(x_o,y_o)$ — illumination function.

K — variable, indicating the place (order) of a boundary point in the sequential boundary stack.

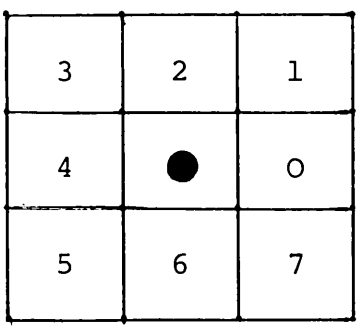
k — variable, indicating the associated with a feature prototype vector.

M — lens magnification; varible indicating the maximum classes of concern.

N — variable, indicating feature vector dimension.

Ne — element population in a linear array.

$N_r(a_{ij})|r = 1,\ldots,7$ — the neighbourhood pixels in accordance with the following mask.



$P = p_{ij}$	processed image, at the output of an operation.
$P(\omega_k)$	a priori probability of occurrence of
$P(\omega_k \underline{\phi})$	a posteriori probability of ω_k conditioned on feature $\underline{\phi}$
PS	a projection set.
pdf	probability density function
R	field of view.
$R(\omega_k \underline{\phi})$	conditional risk.
RS	row sum vector in projection set analysis.
r	image span on the transducer.
rsi	row sum element pertaining to row "i" in orhtogonal projection analysis.
S	storage requirement.
SPRT	Sequential Probability Ratio Test.

T	threshold.
T_L	integration time.
t	time.
w	weight vector.
Z	variable, indicating maximum number of targets in a scene.
α	relative aperture.
$B^{(k)}$	tool width associated with the prototype tool of class ω
β	width of the tool at a predefined distance from datum, used in tool wear measurement.
Θ	orientation angle.
λ	wavelength.
Λ	likelihood ratio.
ν_a	frequency of occurrence of grey level "a".
ρ	resolution.

Σ	number of points associated with a prototype pattern in the orthogonal projection analysis.
σ	number of points associated with a sample pattern in the orthogonal projection analysis.
σ^2	variance.
Ψ	the translation, orientation, size invariant feature extracted from the orthogonal projection set.
Ω	the entire feature space.
ω	a class of pattern.
$\Leftrightarrow$	corresponds to.
$\Rightarrow$	leads to.
$\hat{}$	as a "hat", approximate value.
$\forall$	for all.

CHAPTER 1

INTRODUCTION

1.1. FUNDAMENTAL ISSUES

Vision can prove an important link in automation of numerous industrial processes, typical of which are:

- handling of materials;
- sorting of components;
- inspection of manufactured items with a view to quality control; and
- guidance of manipulators to achieve finer positional control (with potential application in computer aided manufacturing and assembly work).

A vision system capable of automatic

detection;

location;

identification; and

perception of orientation

of targets*, is of sufficient scope to address the principal issues in all such machines that stand to have an application in any of the above typical areas.

The functional block diagram of a vision system with such capabilities is perceived to be as depicted in Fig.1.1. The

---

* Throughout this thesis, target refers to object of interest.

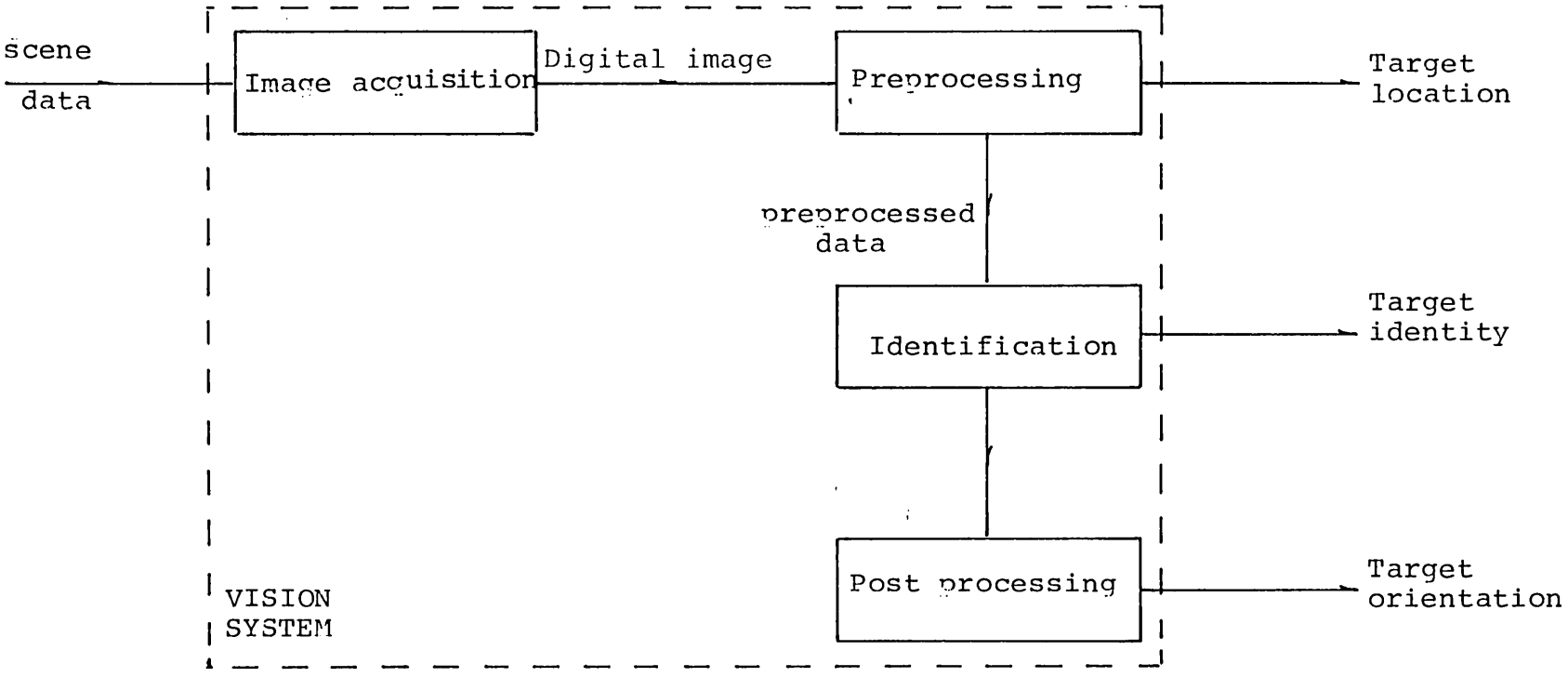


Fig 1.1. Functional block diagram of the perceived vision system

diagram is followed by a synoptic investigation of each block. The separation of the functions as shown in the diagram is not of theoretical consequence but of practical convenience.

1.1.1 Image Acquisition

The task at this stage is to transform the data due to the scene of interest into a matrix of numbers known as the "digital image".

This thesis will adhere to that definition of "image" which identifies it with intensity representation of a scene, with the latter being a field of view in the object space. Furthermore our interest will be confined to optical images which may be viewed as spatial distribution of luminous energy on some bounded region of a plane.

The scene data is conveyed via a multi dimensional irradiant flux $f(x_o, y_o, t, \lambda)$; where

x_o and y_o denote the spatial co-ordinates;

t is the time; and

λ is the spectral wavelength of the light.

As this work will predominantly be concerned with monochromatic stationary images the above function reduces to $f(x_o, y_o)$.

This function may in turn be considered in terms of its constituent components i.e

-illumination, $i(x_o, y_o)$; and

-reflectance, $r(x_o, y_o)$.

The following relation prevails between the light intensity function and its principal components [1]:

$$f(x_o, y_o) = i(x_o, y_o) . r(x_o, y_o) \qquad (1.1)$$

Eq.(1.1) reveals the significance of illumination which falls upon the scene as well as complexities that targets and backgrounds of uncompromising reflectivity can introduce. In other words these two factors attack the system perception of the physical world at the root, and deserve due attention.

The first step in the acquisition process is that of image formation. Throughout this work image formation is achieved via a lens assembly. In general, image forming devices, inclusive of lens assemblies, exhibit neighbourhood properties. That is, a point (x,y) at the image plane is the image not only of a corresponding point (x_o, y_o) in the object space but is also influenced by weighted contributions of the neighbourhood of (x_o, y_o). The function that governs the way in which neighbourhood contribution prevails is termed the point spread function (PSF) of the image forming device. Alternatively, the image of a hypothetical mathematical point source will not be a point but a finite area. This latter area defines the PSF of the lens assembly and may be attributed to effects such as the

aberration or diffraction in the lens [2].

The image formation process in its most general form may be expressed as [3]

$$g(x,y) = \int_{-\infty}^{\infty}\int h\left[x,y,x_o,y_o,f(x_o,y_o)\right]\ dx_o\ dy_o \qquad (1.2)$$

With the assumption of linearity this relation can be simplified to [4]

$$g(x,y) = \int_{-\infty}^{\infty}\int h(x,y,x_o,y_o)\ f(x_o,y_o)\ dx_o\ dy_o \qquad (1.3)$$

and space invariance assumption reduces the above equation to

$$g(x,y) = \int_{-\infty}^{\infty}\int h(x-x_o,y-y_o)\ f(x_o,y_o)\ dx_o\ dy_o \qquad (1.4)$$

Fig.1.2 depicts the elements of the image formation process.

By siting a photosensitive plate at the image plane, the

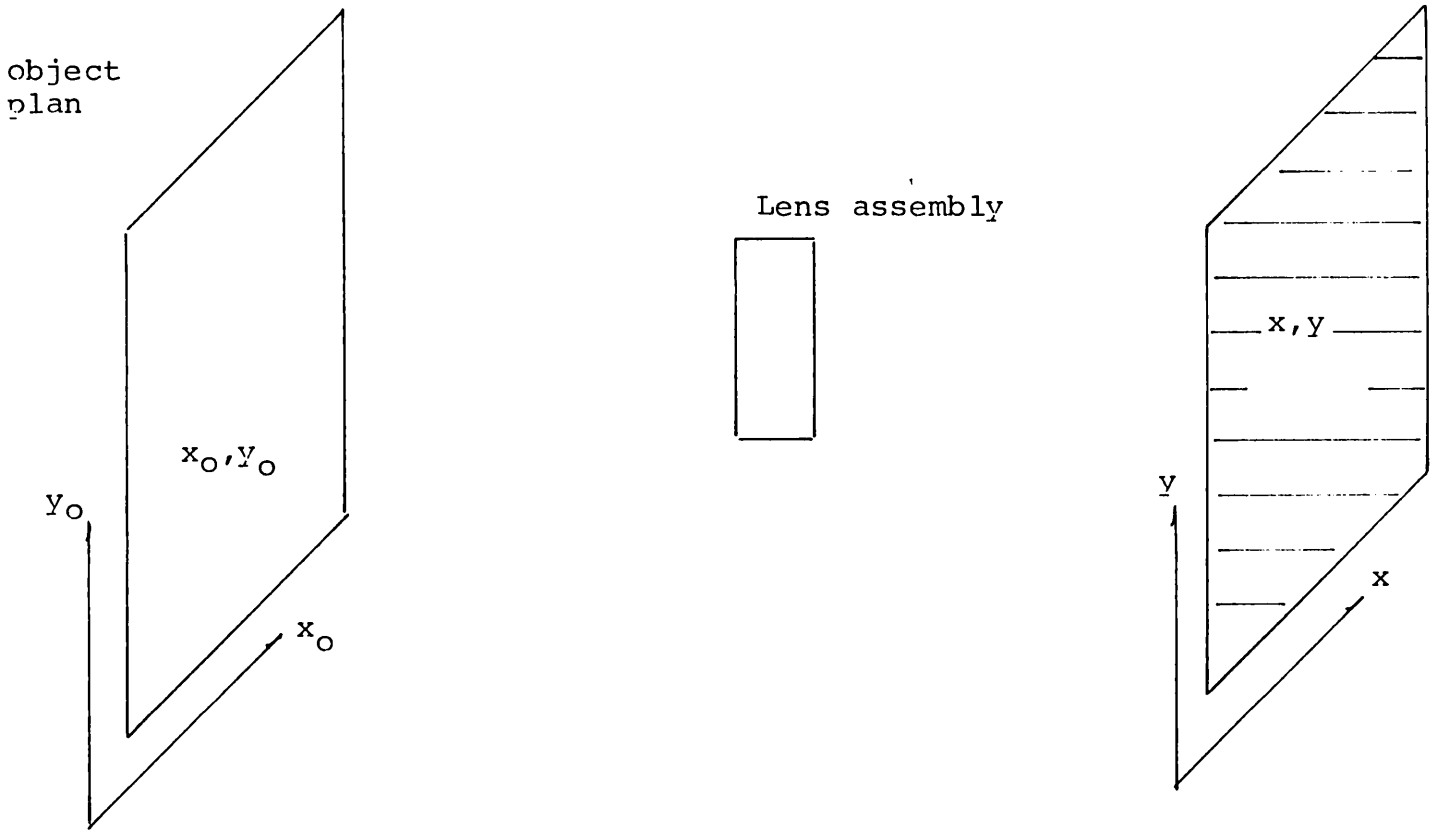


Fig 1.2. A depiction of image formation geometry

energy distribution containing the scene data can be detected and converted to a suitable signal.

Now if the continuous image is sampled spatially and is quantised, a digital representation of data known as "digital image" is obtained which may be viewed as a datum in the sequence of digital processes that follows.

1.1.2 Preprocessing

To appreciate the significance of this functional block it is expedient to digress and review the subsequent stage of "identification", or more appropriately termed "pattern recognition".

Pattern recognition, as conceived here, is performed on individual targets. This imposes the task of isolating targets within the scene, better known as "image segmentation" on the preprocessing stage. Location of targets within the scene can readily follow successful segmentation operation.

Raw images are highly redundant in their information content. Hence confronting the incident target information in its entirety to detect similarities between them and predefined templates can prove highly wasteful of processing resourses if not exhaustive. A more efficient approach would be to seek common properties or "features" to represent patterns of the same class. This would lead to the more efficient scheme of "feature matching".

Usually a single feature is inadequate to represent a pattern particularly when several classes are present. What

is needed is a set of features, usually referred to as a feature vector. A feature vector represents the incident pattern as a point in the feature space. The composition of the feature vectors should converge the representation of each class, in the feature space, to a bunch or "cluster". Furthermore the clusters of different classes must stand as wide apart as possible. Thus in clustering concept, such a feature vector is sought that results in different clusters for different classes of pattern involved. As an additional qualification the feature vector should yield clusters which stand as wide apart as possible for different pattern classes of concern.

The preprocessing stage may hence be summarised as that ..hen raw measurements are operated upon with a view to:

(i) segment the scene into disjoint regions;

(ii) infer the location of targets; and

(iii) facilitate subsequent extraction of features.

The realm of (iii) extends to such areas as encoding, enhancement, and restoration of images.

Patterns at the output of the preprocessing stage may be viewed as a vector $\underline{x}$ of dimensionality R such that

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_R \end{pmatrix}$$

1.1.3 Identification

This stage forms the crux of the problem in all vision systems. It is more often referred to as the "pattern recognition" part. A schematic presentation of this stage appears in Fig.1.3. This part may be decomposed into the following functional components:

- feature extraction; and
- classification.

1.1.3.1 Feature Extraction The term feature extraction, here, is intended to encompass the feature selection process as a prelude to the actual feature measurement.

In feature selection the question is which and how many features to employ as elements of the feature vector from the total set of attainable features. The approach to the problem of "Which features to select?" essentially remains heuristic [5]. The chosen feature vector elements should be those which enjoy the following attributes most:

(i) Assume significantly different values for different classes of pattern concerned.

(ii) Are highly uncorrelated.

(iii) Are reliable, i.e. show little variability for patterns of the same class.

(iv) Are independent of orientation and translation. (Two or more orientation/ translation dependent features may merge to produce an orientation/ translation independent feature.)

Having chosen a collection of viable features we move to

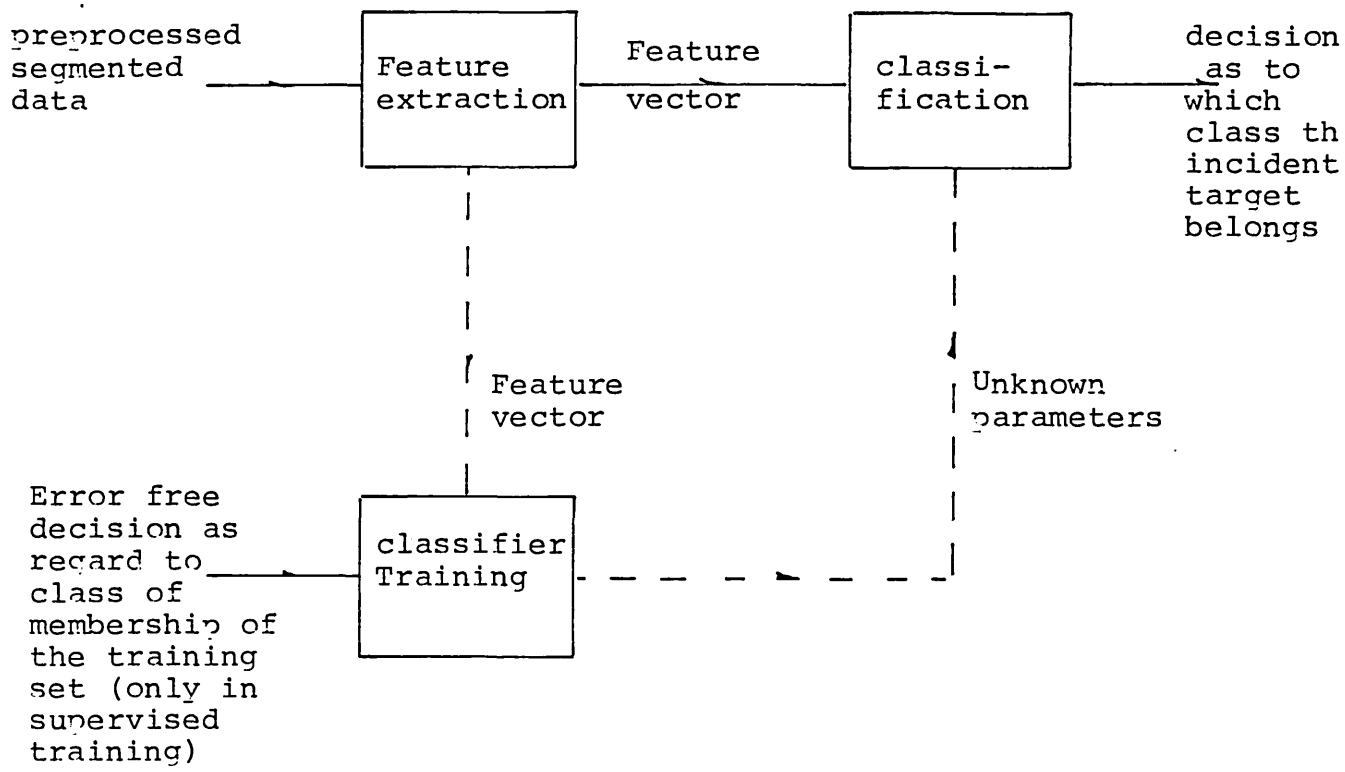


Fig 1.3. The constituent components of the identification stage

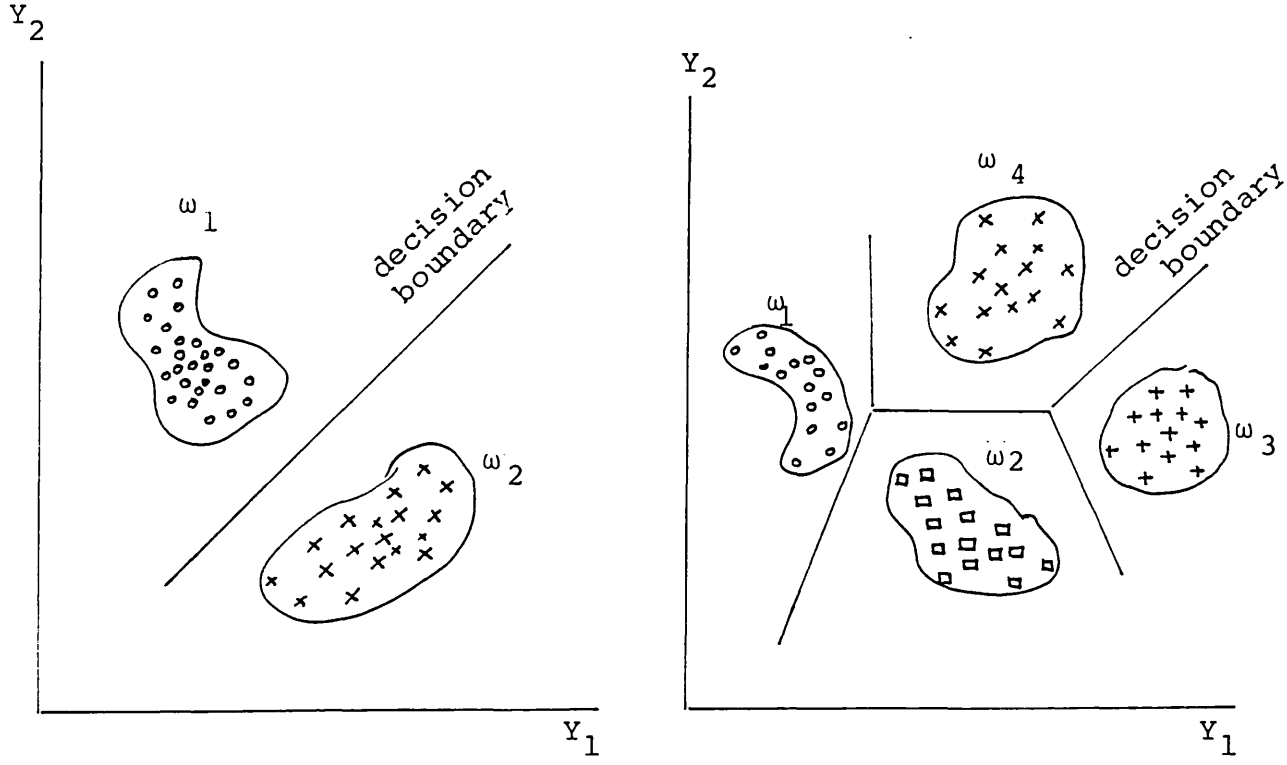


Fig 1.4. Decision surfaces partitioning the 2-dimensional feature space

confront the issue of how many features to select.

In general it might be said that a reduction in the representative features of a class degrades the classifier performance or alternatively overtaxes the classifier. On the other hand inclusion of every feature introduces additional processing effort. Hence a compromise (i.e. a manageable set of features leading to an acceptable classificattion error rate) must be reached.

Having decided about the number of features (i.e. dimension of the feature vector) the next step becomes one of choosing the desired subset of features, say K, from an available set of N features. The brute force approach examines certain criterion such as the classifier performance against every combination of subsets of K features that our total feature set N can accommodate. It then links the merit of every feature subset to the respective misclassification. This usually leads to an exhaustively high number of trials as indicated by

$$\text{No. of trials} = \binom{N}{K} = \frac{N!}{K!\,(N-K)!}$$

If it were desired to determine the best subset of 4 in a pool of 20 feature elements, the number of times that a classification based criterion function would need to be computed is 4845. This is by no means the end of the exercise as reliable classification requires a multitude of

training samples (to be discussed). Hence the number of trials needs to be increased many fold. This renders the exercise virtually impracticable. Thus the inference is that the feature selection process remains heuristic.

The process of feature measurement, as the term implies, may be summed up as the excercise of operating a tool (in software or hardware) which has been devised to compute a selected feature from the preprocessed data.

Each pattern at the output of the feature extraction stage is represented by a feature vector $\underline{\Phi}$ of dimensionality N such that.

$$\underline{\phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \vdots \\ \phi_N \end{pmatrix}$$

where $N << R$.

From the discussion so far it is evident that flexibility in vision systems in the sense of coping with extensive catagories of targets necessitates a feature vector of very high dimensionality (i.e. large number of components). Each element of such a vector is liable to dictate its own peculiar preprocessing. Hence a flexible vision system can

only be realised by resorting to excessively complicated systems, and is only justifiable when a high proportion of its resultant capability is utilised.

1.1.3.2 Classification In the classification phase a decision rule is implemented. By virtue of this action every incident pattern is assigned to its respective class. [Those patterns for which the classifier cannot reach a conclusive verdict can be viewed as constituting a class of their own.]

The decision rule can best be identified by the structure of the supporting classifier. A classifier may employ one of a variety of structures in sympathy with the nature of the pattern representation in the feature space. For instance, when the patterns give rise to a finite number of disjoint clusters, the structure may be a hypersurface separating the respective clusters. Fig.1.4 shows an example of such classifier structure. The decision surface in this instance acts as a separating (or a discriminant) function between different classes of pattern ω_i, k=1,2,...,M, by assuming distinctly different values for different classes of concern.

If the decision function is denoted by $D(\underline{\phi})$, then $D(\underline{\phi}) = D_k$, k=1,...,M assigns the pattern associated with the feature vector $\underline{\phi}$ (namely,the incident pattern $\underline{x}$) to the class of ω_k. In this example the information needed for the classification is fully contained in $D(\underline{\phi})$.

When the clusters (due to inappropriate choice of feature vector or the inherent variability of patterns) overlap, the concept of a pattern belonging to a class must give way to a probabilistic assignment of patterns to various categories. Here, attention is diverted to statistical properties of the feature vector that represents the incident pattern, namely:

(a) The conditional probability density function associated with the feature vector $\underline{\phi}$ for each class, i.e.

$$p(\underline{\phi} \ / \ \omega_k), \quad k=1,\ldots,M$$

[This is the probability that the feature vector will occur, given that the target belongs to class ω_k.]

(b) The prior knowlege (a priori probability) of occurrence of patterns of class

$$P(\omega_k), \quad k=1,\ldots,M$$

Bayes Theorem allows to improve our initial knowledge through estimation of an a posteriori probability upon measurement of $\underline{\phi}$, according to

$$P(\omega_i|\underline{\phi}) = \frac{p(\underline{\phi}|\omega_i)P(\omega_i)}{p(\underline{\phi})} \tag{1.5}$$

where $P(\underline{\phi}) = \sum_{k=1}^{M} p(\underline{\phi}|\omega_k) P(\omega_k)$ is the unconditional probability density function.

If the classifier errors were to cost equally, the classifier task would be computation of the a posteriori probabilities for different classes of interest and

determination of their maximum. The class corresponding to this maximum would reveal the class membership of the incident pattern.i.e. class assignment will proceed according to

$$D(\underline{\phi}) = D_i \quad \text{if}$$

$$P(\omega_i|\underline{\phi}) = \max_{k=1,\ldots,M} [P(\omega_k|\underline{\phi})]$$

However, the losses incurred in the event of misclassification in a multiclass problem are liable to be different. By introduction of a loss function, $\lambda\ (\omega_i|\omega_k)$, the cost of misassignment of a pattern of class ω_k to class ω_i can be quantified. Thus with every decision there will be associated a minimum risk, $R\ (\omega_i|\underline{\phi})$, where

$$R(\omega_i|\underline{\phi}) = \sum_{k=1}^{M} \lambda\ (\omega_i|\omega_k)\ P(\omega_k|\underline{\phi})$$

which in turn leads to average risk over the entire feature space, Ω, as

$$R = \int_{\Omega} R(\underline{\phi})\ p(\underline{\phi})\ d\underline{\phi}$$

The optimal decision rule, known as Bayes decision rule, is based on minimisation of conditional risk and hence the average risk "R". Thus class assignment with the latter scheme proceeds according to

$$D(\underline{\phi}) = D_i \text{ if}$$

$$R(\omega_i|\underline{\phi}\ ) = \min_{k=1,\ldots,M} [R(\omega_k|\underline{\phi}\ )]$$

Classifiers have been catagorised as follows:

(i) Parametric when the functional forms of the probability density functions associated with the feature vector of each class is known. The unknowns in this case are confined to such parameters as mean, variance or a priori probabilities.

(ii) Nonparametric when the functional form of some or all of the probability density functions are not known.

The obvious inference from the above discussion of classification is that, when design of a vision system capable of coping with unspecified catagories of targets is undertaken the choice of an optimum classifier (from the cost-performance point of view) is denied.

1.1.3.3 Classifier Training The aim here is to estimate those parameters in the classifier's structure which are not fully defined, from a collection of pattern samples, generally referred to as the 'training set'. The parametres may range from estimation of pdf's* to their means, variance or covariences. In the case of the discriminant function based classifiers it may be known that a linear or a polynomial discriminant function governs the classifier's structure. Then during the training the decision boundaries are adjusted while preserving the general shapes.

* pdf : probability density function

Nonetheless there are those classifiers whose structural shapes are also concluded in the course of training, e.g. potential function based classifiers [7].

Training sequences are termed "supervised" when the training set is assumed to have been classified by some error free entity; they are considered unsupervised when such a source is not available. This latter scheme will not be pursued in this work.

1.1.4 Post processing

This refers to a sequence of operations whose purpose is to establish the orientation of the targets after their recognition has been achieved.

The processing at this stage may further be extended to refinement of target location or detection of specific points on the recognised pattern.

1.1.5 Scope of Work

Before concluding the discussion of fundamental issues it is deemed proper to justify the choice of the title of this thesis, and thence shed light on the scope of the present work.

Interest in image processing and pattern recognition has grown drastically particularly in the last decade. Rosenfeld

[8] quotes the number of papers published currently in these fields at over 1000 annually. Such a high level of interest can explain the dynamics with which the boundaries of these fields are pushed farther.

Image processing is concerned with manipulation and analysis of images by computers, and Rosenfeld (8) assigns the following as its major subareas:

(a) Digitisation;

(b) Enhancement,restoration, and reconstruction;

(c) Matching, description, and recognition.

On the other hand pattern recognition has been defined [9] as "the study of artificial as well as natural mechanisms that analyse, detect, recognise and describe patterns in sensory and numerical data."

Evidently the border between image processing and pattern recognition is fuzzy, and for this reason we shall attempt to elucidate the intended meanings of these terms.

The term "image processing" in this work will encompass all operations in the following functional blocks of our vision system discussed earlier:

- image acquisition;
- preprocessing; and
- post processing.

The functional block of "identification" will fall under the auspices of pattern recognition. The emphasis throughout

this work will be on the classification aspect of pattern recognition rather than the structural description of the patterns.

In this work an industrial robot is viewed as a programmable manipulator, a definition advocated by many investigators [10-11]. The constituent components of such a machine are the manipulator and the controller. As far as the vision system is concerned it has to communicate its findings of the scene of interest to the robot controller via a standard or a non-standard interface. Thus it is immaterial from the vision system point of view whether the robot emulates a human arm to the same degree of freedom or is constrained to two or three degrees of freedom.

1.2 APPRECIATION OF THE OBJECTIVE

The preceeding section has discussed the variable nature of the processing power in vision systems and have also led to the inference that the following factors are principally responsible for definition of such a requirement.

(i) The extent of preprocessing required - This factor is scene dependent and naturally can be relaxed through appropriate adoption of the image acquisition subsystem, inclusive of illumination;

(ii) Dimensionality of the feature vector - target dependent;

(iii) Acceptable classification error rate - application dependent; and

(iv) Operation cycle time - application dependent.

Examination of the above factors leads to adoption of a suitable scenario and thence design of the hardware necessary to support various operations to be performed.

Thus far it has been established that the complexity of a vision system grows with those of targets and scenes that it must confront. Furthermore realisation of a vision system capable of coping with a wide assortment of targets in a variety of supporting scenes can render a vision system prohibitively expensive.

Thus in the absence of a specific requirement instead of aiming at the design of flexible but industrially or commercially non-viable vision system, it was deemed best to pursue a goal within the latitude of the objective which follows:

OBJECTIVE:

(i) Within the constraints of a low cost system, design and develop an interactive image analysis computer package that can aid formulation and expedite implementation of dedicated algorithms for

(a) detection;

(b) location;

(c) identificaton; and

(d) perception of orientation

of specified classes of industrial components in scenes of varying degrees of complexity.

(ii) With the aid of above, design and develop "stand-alone" vision systems for specific industrial applications.

1.3 ORGANISATION OF THE REPORT

There are two principal areas of development reported in this thesis. The first is an interactive image analyser for prototyping industrial vision systems. To this end, a microcomputer based interactive image analysis package has been developed. It is named CAVDEP (Computer Aided Vision DEvelopment Package). It resides on a M6809 (8-bit) microcomputer and has been developed from scratch totally in assembly. Its description is the subject of chapters 2 and 3.

In addition, chapter 3 reports the theoretical extension of the Wald's Sequential Probability Ratio Test to a compact and efficient parametric classification scheme.

The second is the development of a dedicated vision system for recognition of engine bearing caps. The components are non-planar with varying reflectance, and the

background is equally uncomprimising. Here,the vision system provides real time information to a 3-axes transfer gantry for automated handling of these components and their palletisation. Furthermore, it can engage in pallet location, automatic positional alignment of the loader arm and calibration of its movements. The system has been installed in an engine assembly plant. This is presented in chapter 4.

The work then extends to solution of a hypothetical problem in chapter 5. This serves the purpose of handling of the same engine components in a different way. Thence based on decomposition of the image into an orthogonal projection set and derivation of a new translation, orientation and size invariant feature, the recognition of the same component sets is pursued. The analysis then proceeds to perception of orientation. Expressions are derived from the projection domain for computation of orientation, and a method is introduced for removal of existent ambiguities, usually present in orientation computation.

Finally, an application of practical merit to manufacturing technology is reported in chapter 6. The theory of a new vision based technique for tool presetting, and identification is developed. This also addresses certain types of tool wear measurement. Possible configurations of the supporting vision system are also investigated. The investigation does not extend to implementation of the proposed system in a CNC lathe;it is confined to assesment

of viability via simulation.

The subsequent chapter presents the conclusion, followed by two appendices. Appendix A discusses how image files are transported to the ICCC mainframe complex, and the facility for processing image files. Appendix B provides a brief account of the distance metrics used in this work, for pattern classification.

CHAPTER 2

SYSTEM CONFIGURATION FOR SUPPORTING THE INTERACTIVE IMAGE ANALYSIS PACKAGE

There are two aspects in realising the aims set out in chapter 1: hardware, and software. These aspects for the interactive image analysis facility (addressed in the initial part of the objective) are considered in this and the next chapters, while those of the specific applications will be dealt with in their respective chapters.

2.1. TEST BED FOR EVALUATION OF NEW ALGORITHMS

The equipment utilised in the appraising algorithms and for the demonstration of their viability comprised:

- A closed circuit TV (CCTV) system consisting of a Link Electronics 1Ø9B TV camera, and a monitor;

- A column digitiser;

- A MC68Ø9 based microcomputer system including an HP2648A graphic/alphanumeric display, a floppy disc drive and a printer;

- A PDP-11/45 mini computer system;

- The College multi-mainframe complex (comprising a tightly coupled CDC Cyber 174 and a Cyber 720).

The equipment is configured in accordance with Fig.2.1.

In the ensuing pages the above apparatus will be reviewed in perspective to their alternatives.

2.1.1 The Image Acquisition Subsystem This subsystem consists of the following components:

(a) the ilumination

(b) the optics

(c) the sensor

(d) the digitiser

They will be the subject of the forthcoming discussions.

The photograph of Fig.2.2 shows ,among other items, the image acquisition subsystem employed in this work.

2.1.1.1 Illumination The importance of this component in vision systems cannot be overrated. Illumination must be compatible with the sensor. Furthermore through adoption of an appropriate viewing technique, the illumination should emphasise the contrast between the targets and the background. In brief, illumination should be optimised with a view to ease the subsequent image segmentation and feature extraction operations.

Sources of illumination are diverse, examples of which

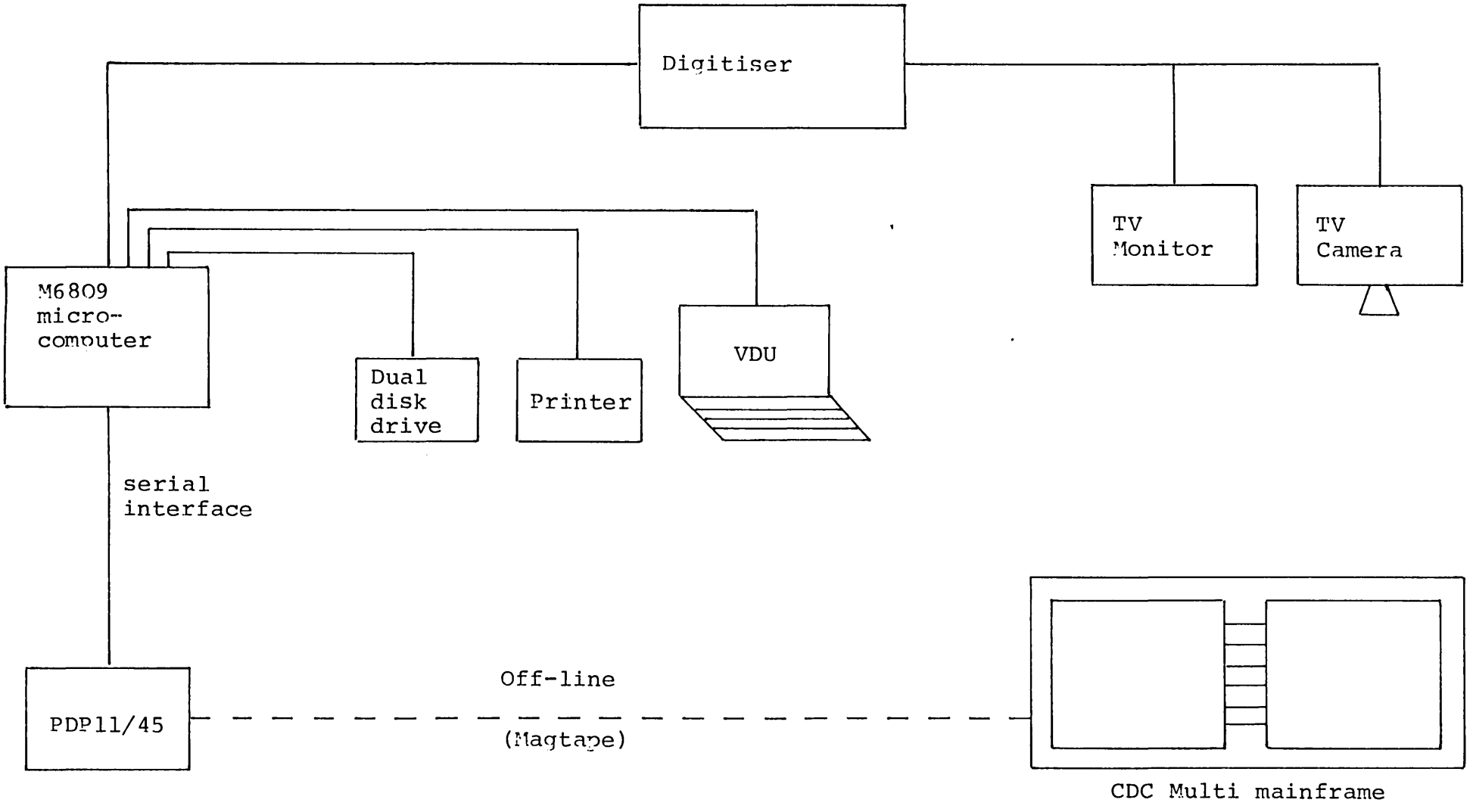


Fig 2.1. Configuration of the test bed

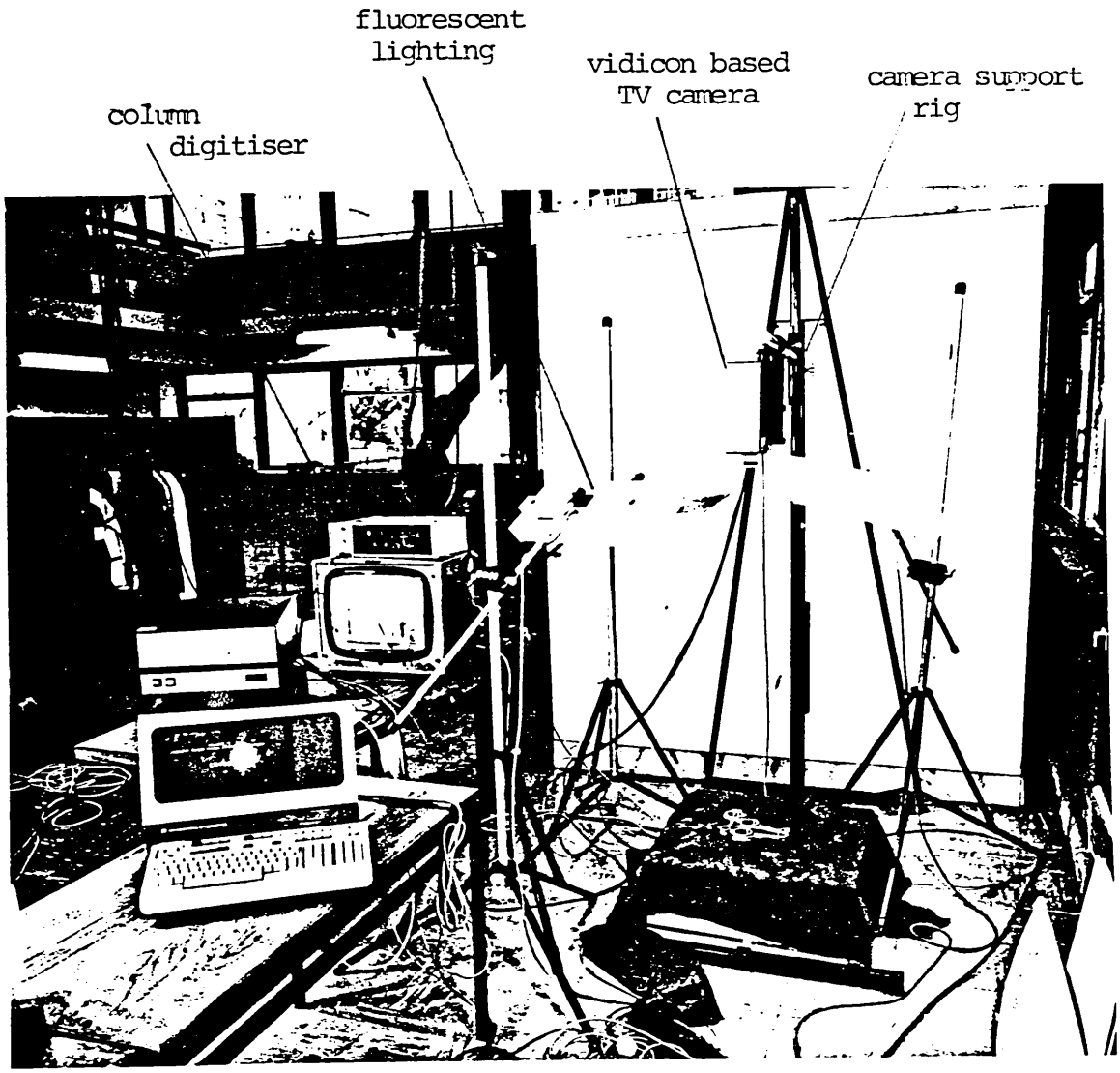


Fig 2.2 Constituent components of the image acquisition subsystem

are: fluorescent, tungsten, tungsten-halogen, sodium, quartz-halogen, gallium arsenide, mercury or xenon arc lamps, stroboscopes, and lasers. This list may be extended to such sources as ultra violet, x-ray or infrared sources whose usefulness in a general purpose development system is debatable [12].

Viewing techniques range from front to rear and structured lighting. Spectral illumination, varied use of mirrors, shadows and collimated light provide further examples of viewing techniques.

In the case of the employed test bed (i.e. the development system) the illumination is provided by fluorescent lighting. The light is further diffused by insertion of diffuser grids before each source. Both front and rear lighting options are available.

2.1.1.2 Optics This refers to an assortment of components such as mirrors, prisms, lenses, optical filters, etc. Their task is to focus the light due to the scene of interest on the image plane in the desired form.

A comprehensive method of reporting the image-forming properties of any optical component is through the Modulation Transfer Function (MTF). MTF may be defined as the modulus of the Fourier transform of the line spread function of the component concerned. For a compound optical

assembly comprising several components, the overall MTF would simply be the product of the MTFs of the successive componets of that assembly. To be conclusive MTFs should be available for various regions from the central axis of the optical assembly.

The pre-eminent component in the optics of an image acquisition subsystem is the lens, whose detail investigation is a discipline in its own right. When the image acquisition stage of a vision system is being designed, it is seldom that one goes through the expense of having a custom lens designed, as a wide assortment of lenses are usually available. For such a purpose the resolving power of the desired lens is more than adequate for specifying its spatial image-forming characteristics. [The subject of desired resolution is addressed in details in the ensuing pages.]

Selection of a lens is, obviously, contingent upon its distinct specification, such that it can be identified in the vast collection of commercially available lenses. Hence the selection process must proceed after the following factors have been given due considerations.

(i) The focal length- which is related to the FOV and the span of the photosensitive transducer (see Fig.2.3), through the following relations:

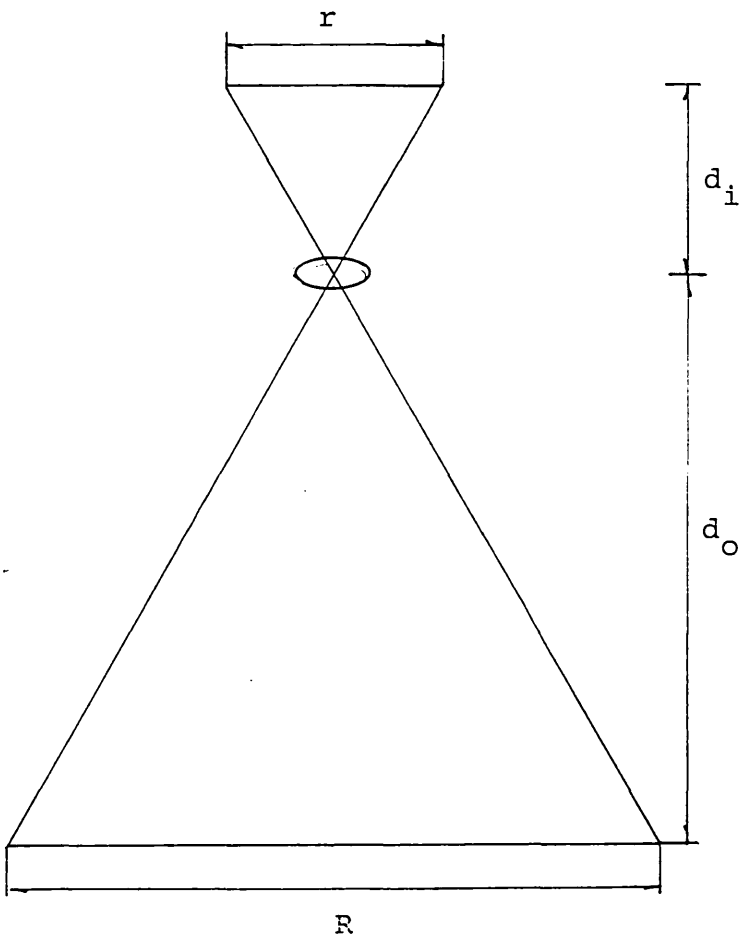


Fig 2.3. Simplified schematic of a lens system

R = Field of view

r = image span on the photo sensitive plane

$$M = \frac{r}{R} \qquad (2.1)$$

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f} \qquad (2.2)$$

where M denotes the magnification and f is the focal length.

Manipulation of the above expression which leads to the following relations describes the lens's focal legth:

$$f = \frac{d_i d_o}{d_i + d_o}$$

$$= \frac{Md_o}{M+1} \qquad (2.3)$$

(ii) Relative aperture (α)- determins the ability of a lens to carry energy into the image plane and is defined as the ratio

$$\alpha = \frac{\text{focal length}}{\text{diameter of entrance pupil}} \qquad (2.4)$$

It affects the depth of field (i.e. the maximum variation in the distance of the scene from the face of the lens without the image becoming unacceptably blurred), and the resolving power of the lens.

2. System supporting interactive package

If we denote the diameter of circle of confusion by "c" (this circle defines the limit of acceptibility in loss of sharpness of the image), then the depth of field "D_f." can be expressed as [13]

$$D_f = \frac{c.\alpha.(M+1)}{M^2} \qquad (2.5)$$

As we can move the scene either way of the optimised plane without violating the acceptibility criterion, then

$$\text{Total Depth} = 2 \cdot D_f \qquad (2.6)$$

The mechanism for effecting an aperture is known as the "aperture stop". In the case of a simple lens the aperture stop is either fixed or has one or two settings, while a compound lens is fitted with an iris, which provides a continuum of adjustments between an upper and a lower limit.

(iii) The variety of existing lenses- they extend from normal to wide-angle, telephoto, long focus, mirror, zoom, relay and macro lenses.

(iv) Type of mount- as the lens must be compatible with the camera housing in this respect. Examples of

mounts are C-mount, K-mount, L-mount, T-mount, U-mount, bayonet-mount, and the microscopic-mount.

The significance of using a correct lens mount is not confined to physical compatibility of the mechanical coupling (e.g. the screw fit), but also extends to the back focal length of the lens. The type of lens mount takes into account where the detector surface sits with respect to the mechanical coupling of the camera housing, and can affect the focussing capability for specified working distances.

(v) Aberration- These are certain lens errors which affect their suitibility for different applications. For example, a lens with aberrations corrected to allow wide angular field may not give good image quality at or near central axis, compared with a lens for a narrower field. An aberration of significant importance in vision work is "distortion", which is attributed to non-uniform magnification across the field of view.

The development system uses a Cosmicar 25 mm lens with an iris that effects a continuum of relative apertures between 16 and 1.9 (usually expressed as f/16 - f/1.9).

The camera support rig, shown in Fig.2.2.. provides the necessary facility for adjusting the working distance.

2.1.1.3 Sensors Sensors assume different forms in vision systems, viz

(i) **Conventional TV cameras** - TV cameras are generally signified by their transduction element. Production of electrical image in such transducers is either based on photoemission or photoconduction. In the first case the incident photon flux impinging on the photosensitive element causes electrons to be emitted, while in the second case the conductivity of the photosensitive element is varied in accordance with the incident luminous energy distribution.

Conventional TV cameras fall under the general catagory of electron-beam imaging sensors. Essentially they may be conceived as consisting of two separate stages of photodetection and readout. At the first stage an image of the scene of interest is focussed on a photosensitive surface. The readout stage is coupled to the initial stage via an evacuated tube. External components are used to focus and scan an electron beam across the photosensitive element, in accordance with a prespecified convention, and hence convert the irradiant energy distribution to a corresponding electrical variation.

If given some sense of spatial identity, the electrical signal at this stage constitutes the electrical image. This is conventionally achieved by

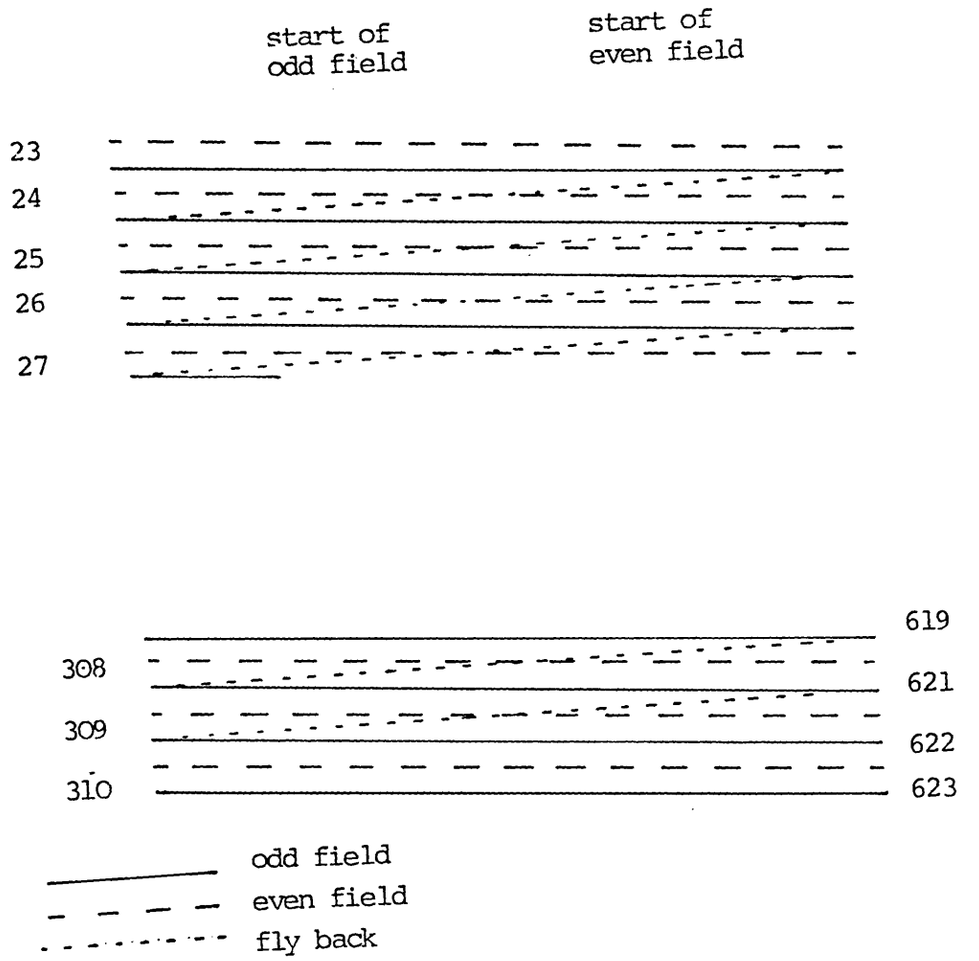


Fig 2.4 The 625 line TV raster scan

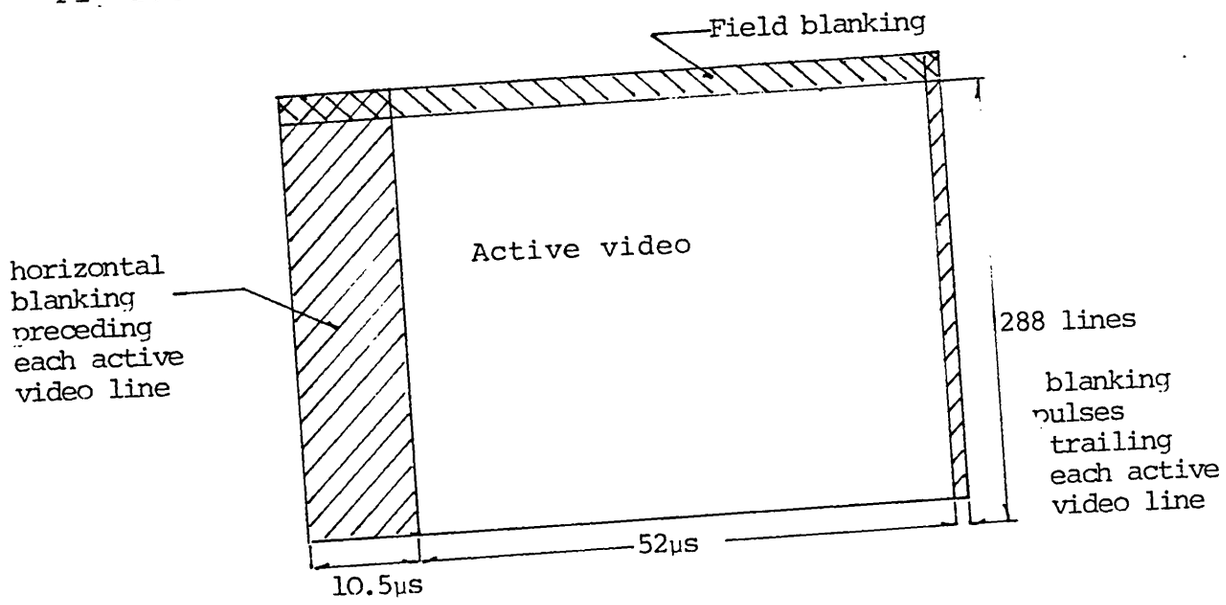


Fig 2.5 Active video in a field

addition of synchronising pulses (horizontal and vertical) to the video information. The mixed signal in this form is referred to as the "composite video".

The convention that governs the operation of the adopted CCTV is that of the 625 line interlaced television. Here a beam scans the entire area of photo sensitive element in 625 lines 25 times each second. Each frame comprises two interlaced fields of 312 ½ lines. The scanning format is depicted in Fig.2.4. Each horizontal line scan requires 63.5 μs of which, approximately, of which 52 μs are active. This is due to the fact that every active video line is preceeded and trailed by 1Ø.5 μs and 1.5 μs of blanking pulses respectively. The blanking pulse sequence allows for flyback as well as the settling of the time-base amplifiers (responsible for scan generation). Of the 625, only 575 line will carry active video and the rest are lost in the vertical retrace, as indicated in Fig.2.4 and Fig.2.5.

(ii) **Solid state arrays-** These sensors are finding increasingly wider applications in vision work. Photodiode CCD and CID arrays are variations of transducers used in these sensors.

A linear photodiode array-based sensor has been used in the application reported in chapter 4. It is also envisaged that a line scan camera can be used

for another application considered in chapter 6. A detailed description of the principles of operation of solid state arrays appears in [14]. Linear photodiode arrays, commonly, operate in "integration mode" [15]. Here, the successive elements of the array are regularly charged to a nominal potential to compensate for the decay caused by flow of a current, predominantly due to the incident photon flux. The interval between the successive charging up is known as the integration time, T_L. On the other hand, data read out in these arrays is either effected digitally or in an analogue fashion. In the application reported in chapter 4, data read out is performed digitally. Here, the elements are multiplexed sequentially at a rate governed by the integration time. The alternative analogue read out is effected by activation of a common transfer line which reflects the reading of each element to a corresponding stage of an analogue shift register, simultaneously. The video information can then be clocked at a rate independent of T_L. The latter configuration is referred as CCPD.

(iii) **Flying spot scanners-** Large size and structural complexity have limited the use of these sensors whose main advantage is high resolution and uniformity of response.

2.1.1.4 Digitisers- Conversion of a continuous image into a digital image (i.e. digitisation) is accomplished through implemention of the following sequence:

(i) Division of the image into pixels; thence measuring and quantising the grey levels at each pixel;

(ii) Storing the resultant grey levels on a storage device (for on-line processing this step may be skipped);

The entity responsible for realising this conversion is known as a "digitiser". Image digitisation is achieved through harmonious operation of the sensor, the digitiser, and the medium that provides storage for the resultant digital image.

To accomplish digitisation, a digitiser must be equipped with the following facilities (not necessarily within its bounds):

(a) a sampling aperture to access the pixels individually;

(b) an aperture scanning mechanism;

(c) a quantiser (i.e. an analogue to digital convertor);

(d) buffer for retaining the digitised image.

In the development system employed herein, (a) and (b) are provided by the internal features of the vidicon based

camera as well as that of the digitiser, while the quantiser is solely due to the digitiser. The storage medium for retaining the digital image (i.e. the image buffer) is provided by the microcomputer.

The number of times that each active video line is sampled determines the spatial horizontal resolution. The frequency with which successive horizontal lines in a frame are accessed constitutes the vertical resolution. Resolution is the degree of discernible details in an image. The resolution is either selected with a view to subsequent reconstruction of a replica of the original image or merely detecting specific details within a Field Of View (FOV). In the former case Shanon's sampling Theorem [16] provides the necessary guidance by prescribing the number of samples to be greater than twice that of the highest frequency component of significance present in the signal. (The monochromatic stationary image of concern constitutes a two dimensional signal). Violation of this criterion would lead to unwanted distortion in the reconstructed image and is referred to as aliasing giving rise to Moire's effect [8] in imaging systems.

As evident, the above theorem requires the original signal to be band limited to half the sampling frequency. To satisfy such a requiremment it would be necessary to pass the continuous signal primarily through a low pass filter where those frequency components beyond that of a threshold are suppressed; the threshold being the highest significant

frequency component present in the original signal. Such a filter is commonly referred to as an "anti-aliasing" filter. The optics or the photosensitive element in the sensor may provide this low pass filtering provided the highest frequency component of interest is well within the latter components pass band.

An analogous situation exists with the quantisation of the signal at successive sampling points. Here another resolution, also known as the radiometric resolution, governs the process. With a coarse resolution the number of grey tones between the white and black will be limited, and in certain scenes, information due to intensity variation is liable to be lost.

In pattern recognition the objective is to represent the patterns by a handful of features, i.e. achieve drastic data reduction far beyond the point where the original data can be reconstructed. As a result, it could be said that the data reduction process could start from the very beginning, namely the image acquisition stage. Almost in all industrial vision systems this notion is followed. In other words sampling with a view to signal recovery is abondoned, and the sampling frequency is governed by the span of field of view and the finest detail of interest within that FOV. The desired spatial resolution may then be obtained through adjustment of the working distance or adaptation of the optics and selection of a suitable sampling frequency.

The above argument should not be extended to the ideal test bed case, where the precise effect of each operation

should be known and unaccounted information loss should be avoided.

In digitising a frame, the obvious solution would be to access successive active video lines in commensurate with the desired sampling frequency. Accessing the video so often although at one time considered a taxing operation, may now be readily achieved by commercially available flash digitiser chips (e.g. TRW's TDC1007J). Storage of the data particularly when sampling at high rates, say 1024 samples per active line, is not as easily achieved as in the former case of video accession.

In the case of stationary imagery the video access rate can be substantially reduced by recourse to a column digitiser. The operational significance of a column digitiser is that in each frame one vertical column is sampled. The digitising column is then moved from left to right in increments on successive frames, until the whole image is converted.

This project has inherited its sensor and digitiser from an earlier project, concerned with digitisation of line drawing. The digitiser, being a column digitiser, accesses every active video line (i.e. samples 575 lines) in each frame and moves the digitising columns from left to right in 1023 increments. The total frame is converted in 8 blocks, each of which consists of 128 columns, in approximately 45 seconds. The digitiser in its current configuration is

interfaced via a PIA port to the microcomputer.

2.1.2 THE PROCESSING NETWORK

A M68Ø9 microcomputer coupled to the image acquisition system constitutes the primary processing network. Provision has also been made to process imagery on the Imperial College multi mainframe complex. In the latter case image data is obtained via the same image acquisition system, via CAVDEP, and is transmitted to the PDP minicomputer (met in 2.1) which is supported by a magtape unit. Thus data acquisition for the second processing medium is achieved via an off-line data link.

A description of this facility appears in Appendix A.

2.1.2.1 The microcomputer as the processing medium

A M68Ø9 microcomputer supported by a array of peripherals forms the first processing medium. Its characteristics are:

- CPU: MC 68Ø9 microprocessor

- Clock cycle: 2 MHz

- Data bus width: 8 bits

- Address bus width: 16 bits

- Memory: 8 Kbytes of EPROM

2. System supporting interactive package

56 Kbytes of CMOS dynamic RAM with a cycle time of 450 ns. Of the 56 Kbytes of RAM only 48 Kbytes are available to the user. The remainder are occupied by the operating system.

- I/O ports. 9 I/O ports one of which is a dedicated parallel port and the rest are capable of accommodating either a dual serial or a dual parallel port, bringing the total facility to 1 parallel port plus 16 serial/parallel ports.
- Cost. under £ 2000.

The microcomputer is supported by the following peripherals:

- A dual, double-sided. double-density, floppy disk drive capable of providing 2.5 Mbytes of mass storage;

- An HP2648A alphanumeric/graphic terminal;

- A printer

The system hardware is illustrated in Fig.2.6.

This processing medium although severely constrained for many applications, does provide a cheap and easy to implement option for numerous other applications that yet remain to benefit from the use of vision systems.

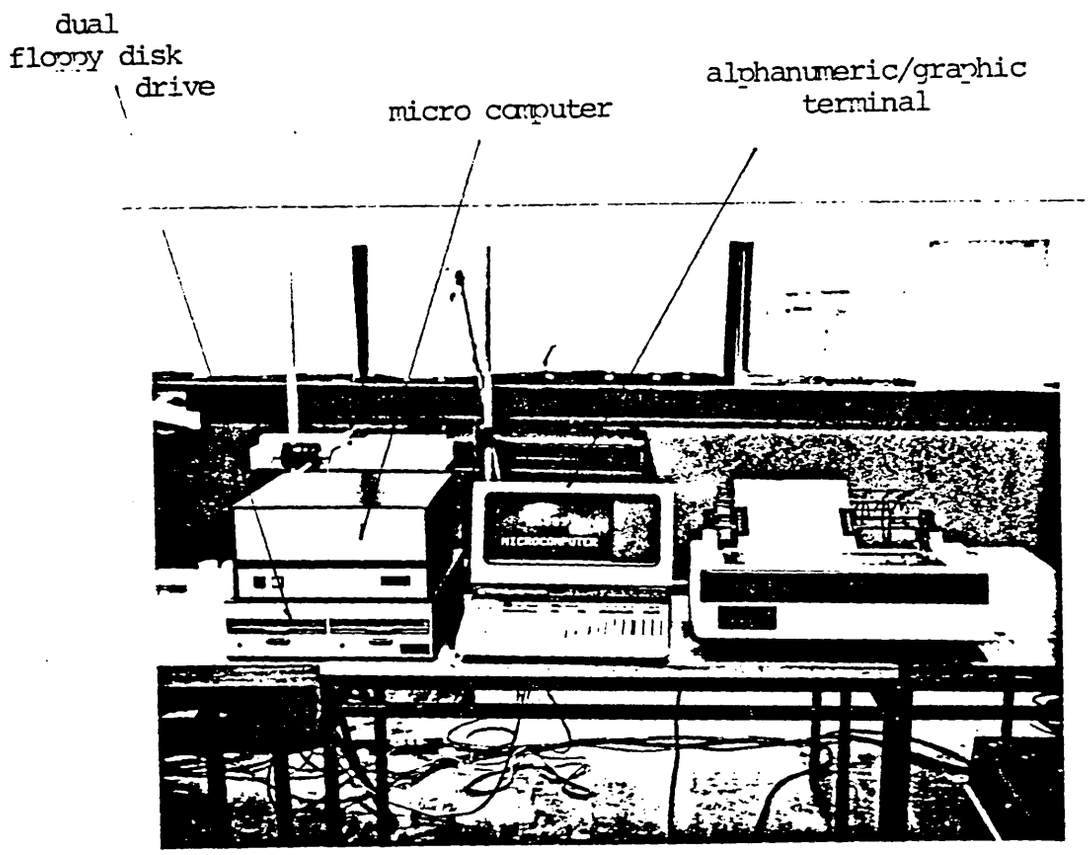


Fig 2.6 The microcomputer as the first processing medium

The attraction is not manifest in the development system but in the relatively cheap and unsophisticated microcontrollers supporting the stand-alone-vision systems for specific applications.

2.2 SOFTWARE ASPECT

Ease of implementation makes the microprocessor based circuit an attractive proposition in implementation of stand-alone vision systems.

The maximum attainable speed of operation with microcontrollers can only be secured through low level (assembly) programming. The drawback of assembly programming is that it is tedious and that an asembly routine is not portable to other classes of machines. Nonetheless, to attain the necessary potential for expeditious design and development of stand-alone vision systems, a collection of efficiently coded microcomputer based routines would prove an invaluable asset. Such routines, of course, should form independent entities so that they could be applied to imagery data in any desired sequence.

If these routines were arranged in an operationally meaningful fashion, such as a package structure, which could easily interact with the user, then it could facilitate not only the upkeep of the routines themselves, but also the introduction and evaluation of new algorithms. Routines representing the new algorithms would ultimately lead to the

package expansion and versatility.

As a result it was decided to develop a microcomputer based package thoroughly in assembly in such a way that its routines would almost be ready for immediate implementation on stand-alone vision systems.

The designation of this package is "CAVDEP" for Computer Aided Vision DEvelopment Package.

It should be emphasised here that, it is not envisaged all stand-alone vision systems will be solely supported by microprocessor based controllers. In many instances the operation cycle time of the application concerned may demand higher speeds only attainable by augmentation of the microcontroller by dedicated logic circuitry, which means implementation of selected routines in hardware form.

In the course of this investigation provision was made to transport image files to the college mainframe computer. The principal advantage of the particular mainframe facility was the extensive support of graphical and statistical library packages available to it. This latter feature is not necessarily shared by all mainframe computers. [Appendix A]

Interactive image analysis systems are not without precedent, but those which focus on industrial applications are rather limited [17]. SUSIE(2) is a prominent example of such a system which was developed as a prototyping tool for

industrial inspections. The original SUSIE is centred around a PDP-12 minicomputer with the software developed in assembly code. The subsequent generation of this system, i.e. SUSIE2(2), is built around an LSI-11 and programmed in FORTRAN [18].

British Robotic Systems (BRS) has also launched an interactive image analysis system which is also based on an LSI-11 and its software is developed in RTL/2. The software for this system has been developed by SPL international and is largely based on the work of Batchelor in UWIST, as reported in [19]. The strategy adopted in SUSIE(2) and by BRS appears to be based on development and evaluation of algorithms in a high level form, and only rewriting them in assembler or implementing them in hardware once implimentation of the design becomes imminent.

Both of the above systems are primarily concerned with inspection rather than robotic vision.

CHAPTER 3

Computer Aided Vision DEvelopment Package (CAVDEP)

CAVDEP is the interactive image analysis package developed in support of the objective (discussed in chapter 1). It was developed from scratch, and now has grown into a 60-operator package. To attain the maximum execution speed, it has been totally implemented in assembly language. Despite constraints of its low cost supporting hardware, CAVDEP addresses nearly all facets of the robotic vision problem. Classification, a less emphasised area in similar packages, is an important aspect of CAVDEP.

A point worthy of note is, that CAVDEP is not based on a fixed sequence of operators, but offers a multitude of alternatives to the user. The user applies its operators on a trial and error basis with the aim of determining the best sequence, for the problem in hand. This attribute makes its description in terms of a sequence of events arising from application of its operators, as is customary in dedicated vision systems or those based on a fixed sequence of operation, almost impossible. Hence when it comes to description of the operations supported by CAVDEP, recourse is made to mere description of its constituents; how the overall operation takes a row image to a state where its

contents are recognised is implicit in the organisation of the package.

This chapter commences with the description of CAVDEP's archetecture, its constituents, their inter-relation, the interface with the user, and then presents the operations they perform. Furthermore, in this chapter, the theoretical extension of the Wald's Sequential Probability Ratio Test (SPRT) to a compact and efficient parametric classification scheme is undertaken (section 3.5.5.1). The derived classifier is not based on the restrictive assumptions of a similar algorithm that appears in standard texts. CAVDEP offers a multi-target, multi-class interactive image analysis facility, and its implementation has entailed formulation of new and modification of some traditional algorithms in order to solve a particular problem, minimise the on-line computational effort, or meet the constraints of the supporting hardware.

3.1. ORGANISATION

Fig.3.1 presents the structure of CAVDEP. CAVDEP is intended to remain highly interactive and flexible to allow execution of any sequence of operation desired upon entering the package. It should further retain a highly modular structure to facilitate subsequent introduction of new operators. It must also provide extensive information to guide the user through the package.

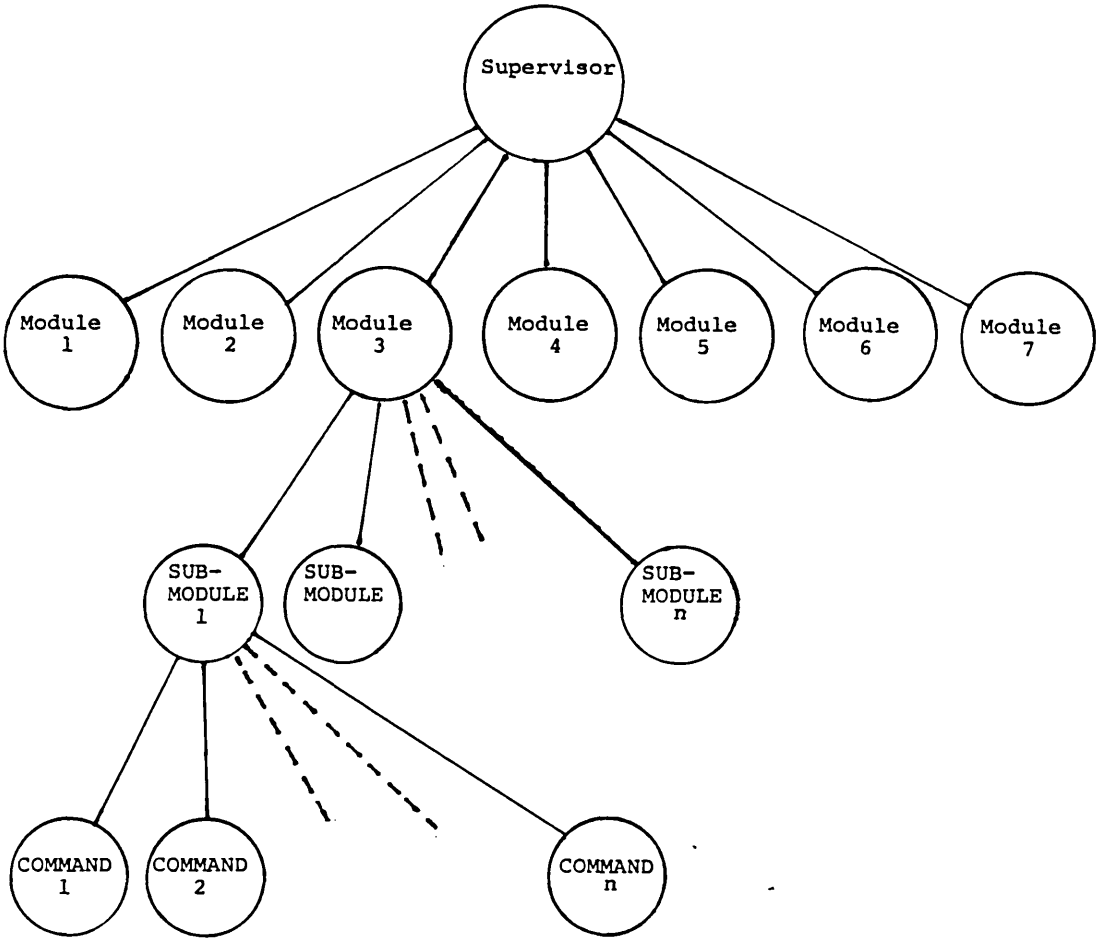


Fig 3.1. The hierarchy of CAVDEP

The levels of hierarchy in CAVDEP are:

- SUPERVISOR: A body of program which presents the module directory and directs the user to the desired module;

- MODULE: A body of program which encompasses a number of submodules and selects the desired submodule;

- SUBMODULE. A body of program which encompasses a number of commands and overlays the desired command;

- COMMAND. An integral entity which performs a specific operation on data.

When the number of commands in a submodule does not exceed one, the submodule level is ommitted and the command is directly catered by the parent module.

3.2 MISSION OF THE MODULES:

The package consists of seven modules, as in Fig.3.2, whose functions are described below.

(i) Module 1 (utility)- This module encompasses all commands that render a service such as displaying, or printing, storing an image file or effecting an interface with other processing media such as the PDP11 system in the case of the microcomputer resident package.

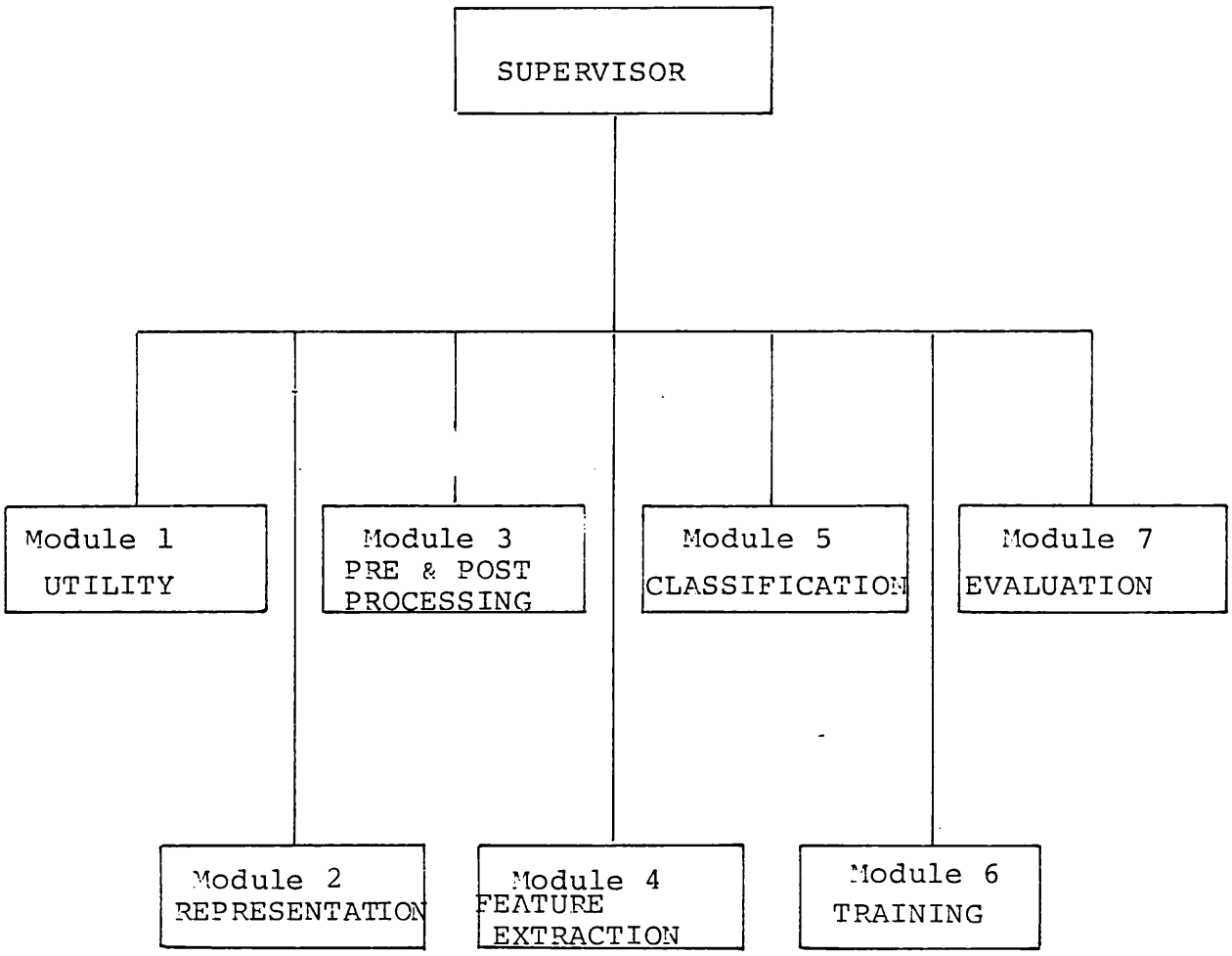


Fig 3.2 Organisation of CAVDEP

(ii) **Module 2 (representation)**- The commands within this module present the system with a digital image. The image in this instance will reside in the assigned buffer. The source may be a live scene or previously stored image file.

(iii) **Module 3 (pre&post-processing)**- This module accommodates commands which can be invoked to smooth, encode, enhance, or generally transform and segment a digitised scene. It further supports those command routines that detect, locate, and determine the orientation of targets within a scene.

(iv) **Module 4 (feature extraction)**- Encompasses all commands that select and measure features, i.e. construct the feature vector.

(v) **Module 5 (classification)**- All classification routines, i.e. functions responsible for execution of the decision rule, reside in this module.

(vi) **Module 6 (classifier training)**- Supports all those commands that aim at training of the classifiers.

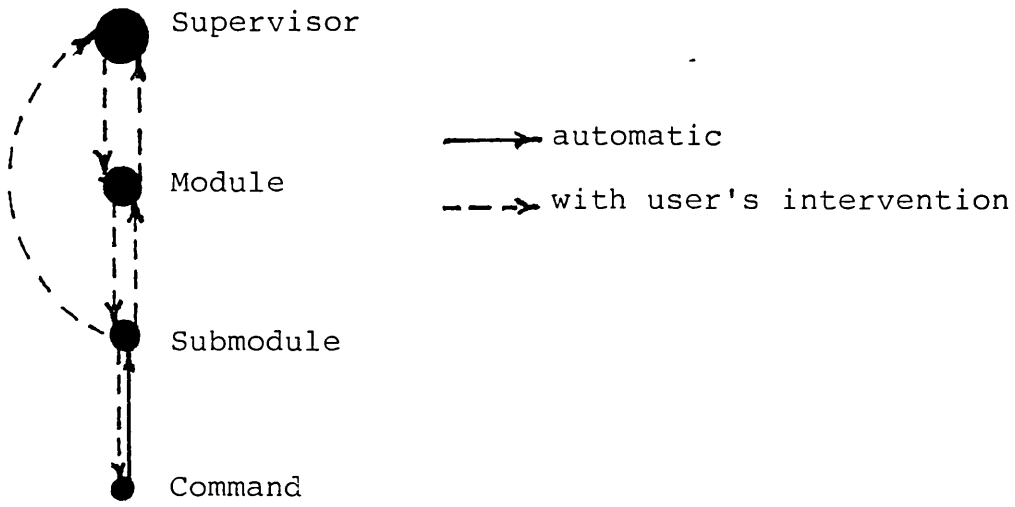
(vii) **Module 7 (evaluation)**- To establish the effectiveness of each sequence of operation and that of the individual commands, yard-sticks must be established, and the result must be measured

accordingly. This module is to embrace all routines that are concerned with evaluation of earlier executed command routines.

3.3 USER INTERFACE AND INTERACTION WITHIN CAVDEP

CAVDEP is a menu-driven package. At each level of its hierarchy, the courses of action open to the user is displayed and his option sought. Fig.3.3 shows such menus at the supervisor level, a module, and a submodule level. After execution of each command, the program returns to its parent submodule.

Movement within the package is confined to paths and nodes shown in Fig.3.1, with the exception of optional jump back to supervisor from submodule level. Hence the up-down movement within the package will be as shown below.



Movement with levels of hierarch in CAVDEP

Lateral movement in CAVDEP can only take place through linking the higher level nodes of the hierarchy. For example, the lateral motion between commands of different

```
*************************DIRECTORY************************
ENTER ANY OF THE FOLLOWING COMMANDS TO INVOKE THE

RESPECTIVE OPTION:


UT...ENTER UTILITY MODULE

RP...ENTER REPRESENTATION MODULE

PP...ENTER PRE&POST PROCESSING MODULE

FX...ENTER FEATURE EXTRACTION MODULE

CL...ENTER CLASSIFICATION MODULE

TG...ENTER TRAINING MODULE

VL...ENTER EVALUATION MODULE


^D...DISPLAY THIS PAGE AGAIN

^F...RETURN TO FLEX

^S...RETURN TO THE PACKAGE ENTRY POINT



      WHAT IS YOUR NEXT COMMAND?
```

Fig 3.3 (a). The menu of options presented upon entering the package.

```
**********PRE&POST PROCESSING MODULE**********

ENTER ANY OF THE FOLLOWING COMMANDS TO INVOKE THE RESPECTIVE OPTION:

GT...ENTER GRADIENT IMAGE SUBMODULE
BI...ENTER BINARY IMAGE SUBMODULE
SG...ENTER IMAGE SEGMENTATION SUBMODULE
EN...ENTER IMAGE ENHANCEMENT SUBMODULE
HT...ENTER HISTOGRAM CONSTRUCTION AND MANIPULATION SUBMODULE
PS...ENTER PROJECTION SET TRANSFORMATION SUBMODULE
IN...ENTER INTERNAL COMPONENTS SUBMODULE
LO...ENTER TARGET LOCATION SUBMODULE
OR...ENTER TARGET ORIENTATION SUBMODULE

^D...RETURN TO SUPERVISOR
^F...RETURN TO FLEX
^S...RE-EXECUTE THE CURRENT  ROUTINE


WHAT IS YOUR COMMAND?
```

Fig 3.3 (b). The command repertoire of a module

```
**********PRE&POST PROCESSING MODULE**********
ENTER ANY OF THE FOLLOWING COMMANDS TO INVOKE THE RESPECTIVE OPTION:

GT...ENTER GRADIENT IMAGE SUBMODULE
BI...ENTER BINARY IMAGE SUBMODULE
SG...ENTER IMAGE SEGMENTATION SUBMODULE
EN...ENTER IMAGE ENHANCEMENT SUBMODULE
HT...ENTER HISTOGRAM CONSTRUCTION AND MANIPULATION SUBMODULE
PS...ENTER PROJECTION SET TRANSFORMATION SUBMODULE
IN...ENTER INTERNAL COMPONENTS SUBMODULE
LO...ENTER TARGET LOCATION SUBMODULE
OR...ENTER TARGET ORIENTATION SUBMODULE

^D...RETURN TO SUPERVISOR
^F...RETURN TO FLEX
^S...RE-EXECUTE THE CURRENT  ROUTINE


WHAT IS YOUR COMMAND?
```

Fig 3.3 (c) The command repertoire of a submodule

submodules in a module is effected through the respective module level, while for commands based in different modules the link can only be made through the supervisor, through the respective modules and submodules to the desired commands.

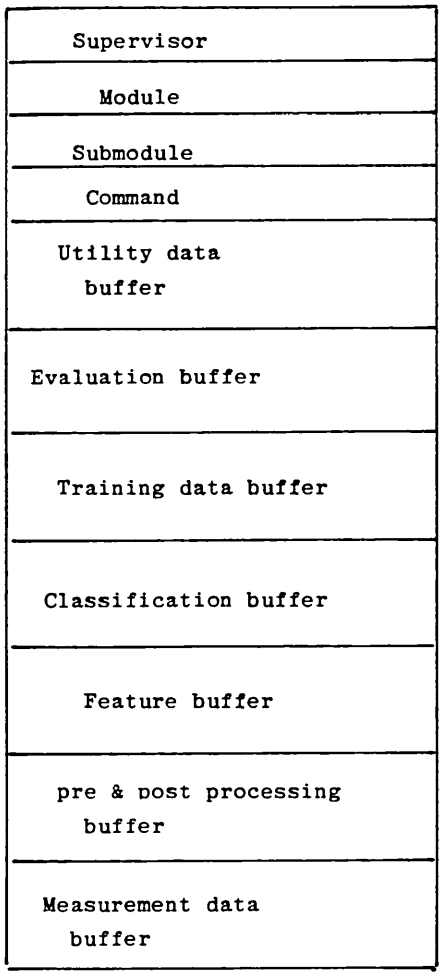
The logic needed for guiding oneself in CAVDEP is that underlying the functional block diagram of Fig.1.1.

The interaction between constituents of CAVDEP can best be revealed by examining its memory map.

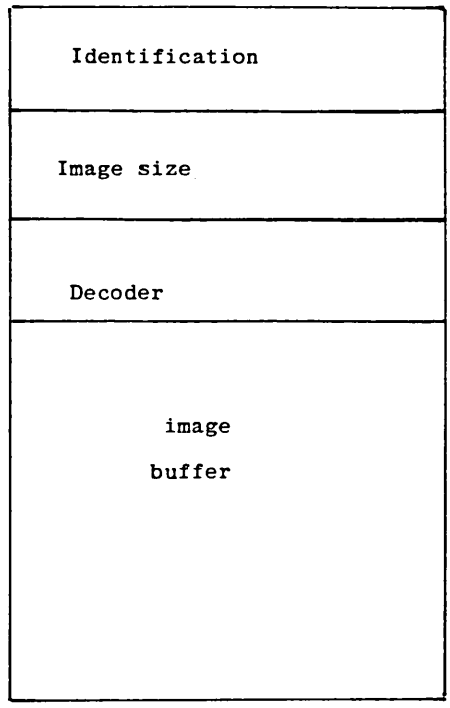
3.3.1. MEMORY MAP FOR CAVDEP

The memory map whithin which CAVDEP operates is shown in Fig.3.4. It should be borne in mind that the employed microcomputer does not offer more than 48 Kbytes of memory to the user as the other 8 kbytes is utilised by its operating system.

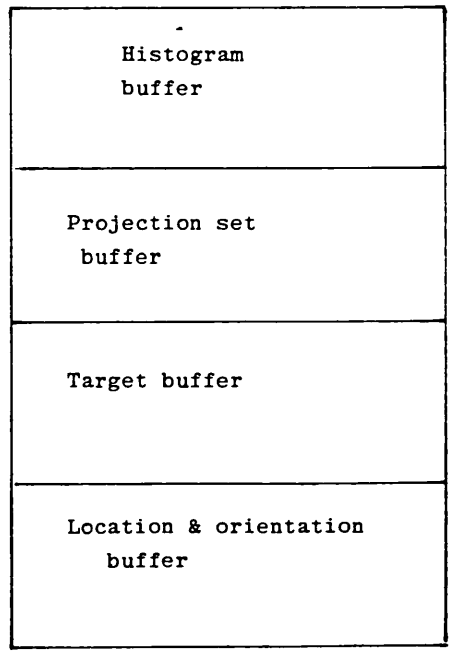
The supervisor resides at all time in its assigned buffer while the sequence of modules, submodules, and command routines will change in accordance with the choice of the user. The assembled version of all of the package routines initially reside on the disk. The operation commences by loading the supervisor in the assigned memory space. The supervisor then overlays the selected module while the submodules and commands are loaded as secondary and tertiary overlays by their respective higher echelon routines in the



a) Overall map



b) Expanded view of the Measurement Data Buffer



c) Expanded view of the pre & post processing buffer

Fig 3.4 A depiction of memory map in the microcomputer based package (CAVDEP)

package hierarchy. Thus at any one time, during the operation of the package, the supervisor, one of the seven modules, a submodule, and a single command routine will reside in the memory. The supervisor will remain at its assigned area of the memory as long as the package is enacted, while the lower echelon routines, i.e. the modules, submodules, and the commands give their places to routines of choice (i.e. in accordance with the input from the user).

The parameters between successive commands are communicated through a designated workspace in the memory.

The memory work space provides a stretch of dedicated memory which in turn is divided into seven dedicated parts, as follows:

a. The measurement data buffer provides 16 kbytes of memory for the acquired digital image, a decoder array for accessing individual pixels in the image, and the frame identifying parameters. The identifying parameters consist of an image size indicater and a frame identifier. The latter is a 6 digit number, the first 3 of which specify the nature of lighting, number of targets, and the digitisation command used in original acquisition of the image. The latter 3 digits are a mere serial number for subsequent reference. [The identifer is used in loading prerecorded image files.]

b. The pre&post processing buffer occupies 11 Kbytes, the larget stretch of memory after the measurement data

buffer. Its details can best be understood after a review of the segmentation process in section 3.5.3.4.

After assigning relatively small memory areas to histogram and projection buffers, the predominant part is occupied by target data. The target buffer is divided into five areas as CAVDEP caters for a maximum of 5 targets. Each area provides a separate region for stacking target boundary coordinates, as met during the boundary tracking operation (explained later in 3.5.3.4). Moreover the same information (i.e., the coordinates of the boundary points) are reorganised and retained in row-ordered array. The target data buffer also provides storage for a corner (angular) representation of the traced boundaries. The pre&post processing buffer further holds memory areas for moments, internal points, location and orientation related data.

c. The feature buffer stretches over a 1.5 Kbyte space of memory, and retains the feature-target data in as elements of a matrix along with a decoder for accessing locations of interest. Additionally space is provided for a feature vector buffer.

d. The classifiers are assigned a relatively small buffer space of Ø.5 Kbytes, as they are principally concerned with use of data stored in the training buffer.

e. The training buffer is for retaining parameters that are necessary for implementation of classification. It extends to 2 Kbytes, and provides storage for the training data associated with each prototype feature element and class of pattern. With its associated decoder this buffer may be viewed as a 3-dimensional array. The nature of information retained in this array varies for different type of classifiers. The stored data may be singular sample values or statistcal parameters extracted from the training samples. Furthermore, the label-class associated information, extracted from user is also retained in this buffer.

f. The evaluation data buffer is a stretch of memory for retaining information computed by commands of the evaluation module.

g. The utility data buffer provides a stretch of memory for service routines.

3.4. DESIGNATION OF INDIVIDUAL ROUTINES

AS all routines are disk-resident, then both for overlaying purposes, and others, they are named. In an expanding package, arbitrary assignment of names can soon lead to chaos. Hence the ensuing rule was observed.

a. The modules were named in commensurate with their mission as: UTILITY, REP (for representation), PPROC (for pre&post processing), FEATRX (for feature

(for pre&post processing), **FEATRX** (for feature extraction), **TRG** (for training), and **VLUATE** (for evaluation).

b. The submodules take the first letter of the parent module plus the two letter whose input invokes the respective submodule from the module level, e.g. **UDY** is the display submodule.

c. In the case of the command routines the above is extended by the additional two letters necessary for invoking an operator within a submodule, e.g. **UDYPT** which signifies the commoand routine for displaying a selected part of the image with higher resolution.

Every routine in the package is also identified by a reference number. In the case of modules it is a 1-digit number, while 2-digit and 3-digit numbers are assigned to submodules and command routines respectively, e.g. UDY is SMOD11, while **UDYPT** is CMD112.

3.5. CONTENT OF CAVDEP

Currently, CAVDEP suports in excess of 6Ø commands; each command is synonymous with a specific image processing, pattern recognition, or a supportive operation. In the ensuing sections CAVDEP is described in terms of the operations performed by its constituents. Such a description has been pursued to reveal the underlying algorithms of

the commands in a concise way. Some of its commands are standard image processing operators and have only been mentioned to indicate the overall capability of the package. Some others have been derived to reduce the on-line computational effort or have been tailored to meet the constraints of the supporting hardware. As will be evident, CAVDEP is emphatic on pattern classification aspect of the robotic vision problem.

For ease of reference some of the notations are re-introduced here, while others are explained within the text where deemed appropriate.

$A = \{a_{ij}\}$ the image at the input stage of an operation.

$E[\cdot]$ mean of (.).

$D(\underline{\varphi}) = D_i$ decision function, assigning the pattern represented by feature vector $\underline{\varphi}$ to class ω_i.

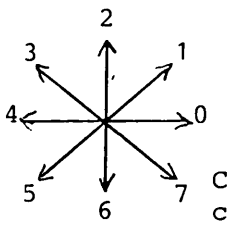
K variable, indicating the place (order) of a boundary point in the sequential stack.

k variable, indicating the class of prototype feature vector.

M variable, indicating the maximum classes of

$N_r(a_{ij}) | r = 0,...7$ the neighbouring pixels in accordance with the following mask.

3	2	1
4	●	0
5	6	7



Corresponding coupling vectors

T threshold value.

Z variable, indicating maximum number of targets in a segmented image.

$\Delta(.)$ distance function.

φ feature vector representing the sample pattern.

$\Phi^{(k)}$ prototype feature vectorof class "k".

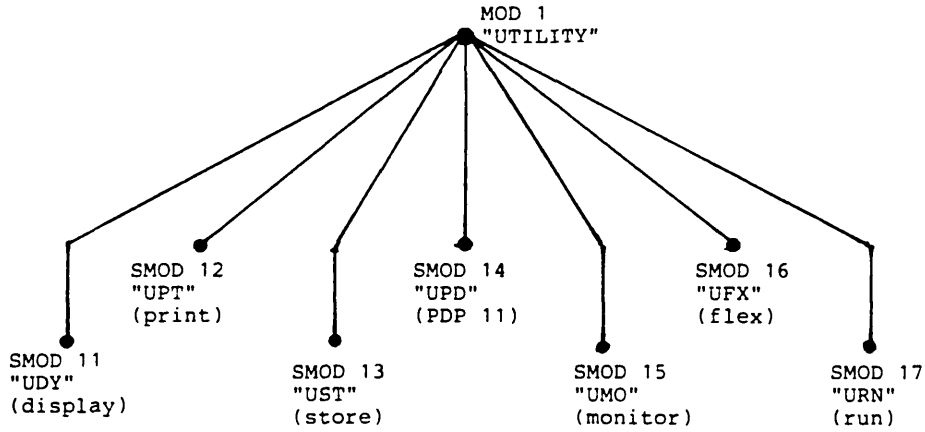
$P = \{P_{ij}\}$ the processed image at the output stage of an operation.

Γ_r the set of boundary points, representing the "rth" target in a segmented image.

$\Leftrightarrow$ corresponds to.

$\Rightarrow$ leads to.

3.5.1. Module 1- "UTILITY"



Structure of utility module down to submodule level.

This module encompasses all commands that render a service. The commands are grouped in 7 submodules as indicated above.

3.5.1.1. SMOD11- "UDY" (display)

The commands within this submodule display either the whole of the image with a lower resolution or part of the image, not exceeding 64 pixels in width, with the same resolution as that of the buffered image. The grey tones can be represented by 16 alphanumeric symbols of choice. Furthermore, the commands allow selective display of higher or lower bit planes.

3.5.1.2. SMOD 12- "UPT" (print) Prints as above, without the restriction of confining the high resolution image to 64 pixels.

3.5.1.3. SMOD 13- "UST" (store) The commands of this submodule allow storage of any size widow, extending to the whole, of the buffered image on the disk. The stored file will either be a text file (ie in ascii characters) or a binary file. The stored image may be expressed as

$$A = (a_{ij} : i = p, \ldots, p + r;\ j = q, \ldots, q + s) \qquad (3.1)$$

[The commands in CAVDEP are so designed to cope with images of varying sizes. The image size parameter in the measurement data buffer (Fig.3.3b) passes the information to subsequent stages, when a file is loaded from the disk. This is simply achieved by recording r and s in relation (3.1), above, alongside the rest of the image data.]

3.5.1.4. SMOD 14- "UPD" (PDP11) This submodule expands the processing network, by effecting a serial interface, to the PDP11 (RSX) minicomputer system.

Through selection of appropriate commands, various disk resident image files (binary or ascii) or memory resident images are transmitted to the PDP11. Transmission is based on sending ascii characters one by one and matching the returned echo to secure a reliable link.

Transmission of disk resident text files to the PDP11, or buffering text files incident from the PDP11 are among the range of facilities provided.

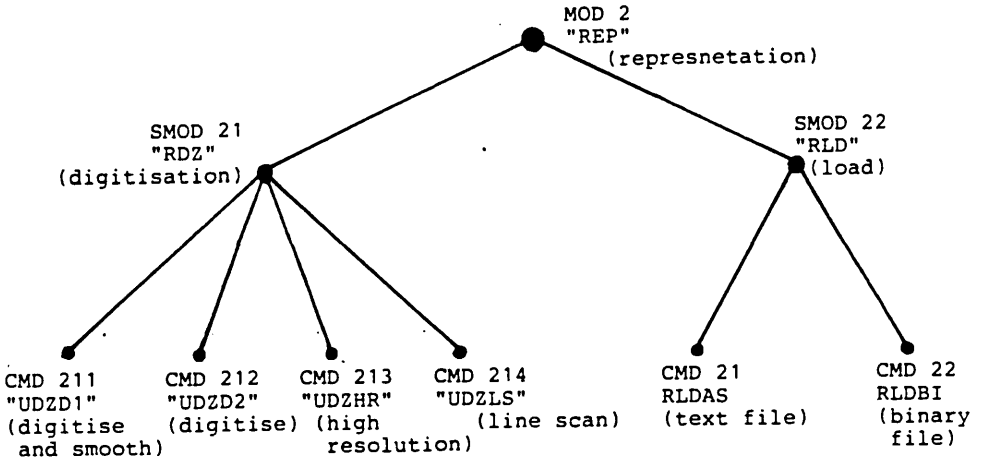
Alternatively, in the simple operation mode, the microcomputer becomes transparent to the terminal, and the user accesses the PDP11 system as one of the latter's terminals. Hence the content of the magtape, or the files that reside on the winchester disks of the PDP11 system can be checked.

3.5.1.5. SMOD 15- "UMO" (Monitor) This module contains a singular command which effects a dedicated monitor program from within the package. It allows to access any memory location with the view to read and write. This grossly facilitates debugging of new additions.

3.5.1.6. SMOD 16- "UFX" (Flex) Allows utilisation of specific Flex 9 (the microcomputer operating system) routines, e.g. cataloguing of disk resident files, from within the package.

3.5.1.7. SMOD 17- "URN" (run) This is again a submodule hosting a single command, whose aim is to execute any predefined sequence of commands automatically. This command is currently under development.

3.5.2.Module 2- "REP" (representation)



Structure of the classification module down to command level.

3.5.2.1. SMOD 21- "RDZ" (digitisation) The commands of this submodule provide a digital representation of, part or whole, of the scene viewed by the camera.

The digitiser transforms the scene into a a matrix of 288x1Ø24 of 4-bit elements. This is achieved by progressing a digitising column from left to right in 8 blocks each comprising 128 columns.

UDZD1 applies a moving average filter in the course of digitisation, and restores the image dimension to its correct aspect ratio. Here, only a central window of 128x128 pixels is buffered. The command also sets up an address decoder for subsequent pixel accession.

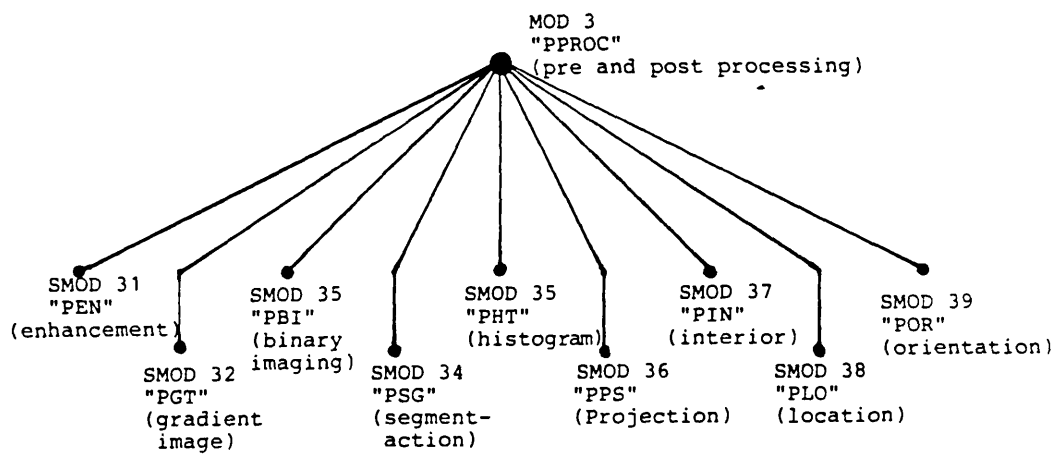
UDZD2 represents the scene with a matrix of 96x128 elements.

UDZHR Digitises the scene with a resolution of 1Ø24x288, yet only buffers that contained in a single blc ck, as 128x128 matrix.

UDZLS Digitises the scene data incident to a prespecified column, simulating a line scan camera.

3.5.2.2. SMOD 22- "RLD" (load) The commands of this submodule load a disk-resident image file into the memory.

3.5.3. Module 3- "PPROC" (pre&post processing)



Structure of the pre and post processing module

This module currently hosts 9 submodules each of which accommodate a variety of operators. The nature of operations is discussed next when considering the submodules.

is discussed next when considering these submodules.

3.5.3.1. SMOD 31- "PEN" (enhancement) This submodule supports commands that smooth or sharpen an image; complement or square its intensities; sink into background its intensities; or write onto any part of the image with pixels of desired grey levels.

Image smoothing and sharpening are performed via low pass and high pass filtering of the image. The operation is linear and in both cases can be expressed by the following convolution.

$$p_{ij} = \sum_{k=-1}^{1} \sum_{\ell=-1}^{1} h_{k\ell}\, a_{(i-k),\,(j-\ell)} \quad | i = 1,\ldots,m-1;\ j = 1,\ldots,n-1 \qquad (3.2)$$

h in the smoothing operation is derived from the elements of

$$H = \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

which is a spatial neighbourhood averaging operator. For the high pass filtering case the Prewitt approximation of the Laplacian operator [19] is used, i.e. h in relation (3.2) is derived from the elements of

$$H = \begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

The grey level squaring opration performed by "PENSQ" may simply be expressed as

$$P_{ij} = \frac{(a_{ij})^2}{16} \mid i = 0,\ldots, m-1$$

The grey level complementing command performs the following operation

$$P_{ij} = \bar{a}_{ij} \mid i = 0,\ldots, m-1;\ j = 0,\ldots, n-1$$

PENSK sinks the grey levels in an image up to a specified threshold into the background, i.e.

$$P_{ij} = \begin{cases} a_{ij} & a > T \\ 0 & a \leq T \end{cases} \quad i = 0,\ldots, m-1;\ j = 0,\ldots, n-1$$

PENWR enables to overwrite the pixels within any selected window of the image by grey tones of choice.

3.5.3.2. SMOD 33- PGT (gradient image) In this submodule, by appliction of either the Sobel [6] or the Robert cross operator [20], the grey level image is transformed into a gradient image.

In PGTSO (the Sobel-based command) pixels in the course of

a raster search are transformed as follows

$$P_{ij} = \{|N_1(a_{ij}) + 2N_o(a_{ij}) + N_7(a_{ij}) - [N_3(a_{ij}) + 2N_4(a_{ij}) + N_5(a_{ij})]| + |N_5(a_{ij}) + 2N_6(a_{ij}) + N_7(a_{ij}) - [N_3(a_{ij}) + 2N_2(a_{ij}) + N_1(a_{ij})]|\}/8$$

with $i = 1,.., m - 2$; $j = 1,.., n - 2$

$$P_{ij} = 0 \Big|_{\substack{i = 0,\ m-1 \\ j = 0,..,\ n-1}} \quad \text{and} \quad P_{ij} = 0 \Big|_{\substack{i = 0,..,\ m-1 \\ j = n-1}}$$

Transformation of the grey level to gradient image by the Robert operator in **PGTRT** proceeds according to

$$P_{ij} = [|a_{ij} - a_{i+1,j+1}| + |a_{i,j+1} - a_{i+1,j}|]/2$$

with $i = 0,.., m - 2$; $j = 0,.., n - 2$; and

$$P_{ij} = 0 \Big|_{\substack{i = m-1 \\ j = 0,..,n-1}} \quad \text{and} \quad P_{ij} = 0 \Big|_{\substack{i = 0,..,m-1 \\ j = n-1}}$$

3.5.3.3. SMOD 33- "PBI" (binary imaging)

The commands here extend from binary thresholding of the image to various operations on the image.

PBIBI binary thresholds the image according to

$$P_{ij} = \begin{cases} 0 & a_{ij} < T \\ 1 & a_{ij} \geq T \end{cases} \quad i = 0,..., m - 1,\ j = 0,..., n - 1$$

PBIBG binary threshods the image while deleting those points with gradients above a predefined threshold, as

$$P_{ij} = \begin{cases} 0 & \{a_{ij}: a < T\} \quad \{\text{Grad } (a_{ij}) > T \text{ grad}\} \\ 1 & a_{ij} > T \end{cases}$$

PBICO complements the image as

$$P_{ij} = \bar{a}_{ij} \mid a_{ij} = 0, 1;\ i = 0,\ldots, m - 1;\ j = 0,\ldots, n - 1$$

PBIRS removes isolated white pixels as

$$P_{ij} = \begin{cases} 0 & a_{ij} = 1,\ N_r (a_{ij}) = 0 \mid r = 0,\ldots,z \\ a_{ij} & \text{otherwise} \end{cases}$$

PBISH shrinks the white regions by removing all white pixels that have a black neighbour.

$$P_{ij} = \begin{cases} 0 & \text{if } a_{ij}=1 \text{ and } \emptyset\varepsilon\{N_r(a_{ij}) \mid r = 1,\ldots, z\} \\ a_{ij} & \text{otherwise} \end{cases}$$

PBIXP expands the white regions by converting the black

white pixel, as

$$P_{ij} = \begin{cases} 0 & a_{ij} = 0 \text{ and } \emptyset \varepsilon \{N_r(a_{ij}) | r = 1,\ldots,z\} \\ a_{ij} & \text{otherwise} \end{cases}$$

The shrink and expand commands can be excecuted for a prespecified number of iterations. [Equal iterations of PBISH's followed by PBIXP's can be used to reduce noise, and tneir execution in reverse order can alleviate interior inhomogenities, e.g. holes, blemishes, etc.

3.5.3.4. SMOD 34- "PSG" (segmentation) The commands within this submodule isolate targets in a gradient or a binary image, by tracing their boundaries. The segmentation capability is confined to a maximum of 5 targets per scene.

In PSGGT (segmentation of a gradient image) the steps taken are as follows:

(i) The gradient image is thresholded according to

$$P_{ij} = \begin{cases} a_{ij} + 10_{16} & \text{if } \text{Grad}(a_{ij}) \geq T_{grad.} \\ a_{ij} & \text{if } \text{Grad}(a_{ij}) < T_{grad.} \end{cases}$$

(ii) Search for initial boundary points starts from the

top left hand corner of the image, in a raster scan. The coordinates of the first edge point met, i.e. a point with $P_{ij} = a_{ij} + 10_{16}$, is marked as

$$P_{ij} = a_{ij} + 30_{16}$$

provided it satifies the criterion of not being due to target interior inhomogeneity. [This is to avoid tracing internal edge lines with a fictitious notion of external boundary tracking.]

(iii) With an initial sector search angle of -135 to +135 at the initial boundary point, search for a second boundary point progresses. The first unprocessed edge point, i.e. $P_{ij}=a_{ij}+10_{16}$, is assigned to the set of boundary points, τ_r , of the respective target, by being marked at initial image buffer as

$$P_{ij} = a_{ij} + 20_{16}$$

while its coordinates are stacked in the respective target buffer.

The sector search angle reduces to -90 to +135 degrees for subsequent boundary points. Boundary tracking is abandoned if the sum of the points encountered exceeds a predefined limit. When the angular search ceases to reveal a subsequent boundary point, a flexible recovery scheme is pursued. Recovery is based on initial expansion of the angular sector to -135 degrees, and if still unsuccessful, a jump along the last vector to extend the search to the next point. The latter steps can be repeated for

a specified number of times. Tracking is abandoned, only when the recovery scheme fails.

A cleaning process follows abandoned traces. Here, all markings inclusive of stacked boundary coordinates are nullified.

Boundary tracking for a target ceases when a closed loop is traced. A closed loop is concluded when the corresponding initial point is met for a second time. An acceptable closed loop must also manifest a point sum in excess of a predefined lower limit. [The upper and the lower limits for target boundaries as well as the nature of the recovery scheme are deduced from prior knowledge of targets and their supporting scenes.]

(iv) To avoid initiation of spurious tracking, a thinning operation is undertaken. Here, points with $P_{ij} = a_{ij}+10_{16}$ which occur in the neighbourhood of confirmed boundary points are reduced to $P_{ij} = a_{ij}$

(v) Tracking of internal edge lines, prior to establishment of their respective external boundary is avoided by "consulting" the **Seacrch Inhibit (SI) array.** This is a row-ordered array which indicates the intervals in which detected initial boundary points are invalid. [By observing this rule, an image can be segmented only by recording separate sets of

coordinates corresponding to the external boundaries of its targets.]

The rows in **SI array** correspond to those of the image. In scences containing multiple randomly appearing targets, the sequential boundary stack is transformed into a row-ordered array (immediately after succesful tracing of each target). The columns are then transferred to the respective rows of **SI array**, to provide an upto date information for the rest of tne search.

An optional angular (corner) representation of target boundaries can be effected during this segmentation process [addressed in more detail under MOD 4].

PSGBI, progresses very much in the same way as **PSGGT**. The exception being that, the edge points are extracted from a binary image.

3.5.3.5. SMOD 35- "PHT" (histogram manipulation) This submodule supports three commands the first of which, PHTHT, constructs the histogram of any desired part of the image by accummulating

$\{\nu_a | 0 \leq a \leq 15\}$ in the course of a raster scan. [ν = frequency of occurence of grey level a.]

PHTEQ performs histogram equalisation. This is a point

operation which yields a transformed image with flatter histogram. The algorithm comprises two steps, as follows:

(1) The transformed grey levels T(a) are computed via

$$T(a) = 16 \sum_{k=0}^{a} \frac{\nu_k}{\nu} \mid a = 0,\ldots,15$$

where

ν = total no. of pixels within window of interest

(2) The intial grey level image is mapped to the nearest T(a).

The last command, **PHTPL**, plots the histogram on the screen.

3.5.3.6. SMOD 36- "PPS" (projection) The commands of this submodule are concerned with transformation of binary images into their orthogonal projection sets, and plotting the resultant projection vectors. A detailed discussion of the topic is contained in chapter 5.

3.5.3.7. SMOD 37- "PIN" (interior) This submodule is concerned with segmentation of interior inhomogenities of isolated patterns. It operates much in the same way as the commands of PSG submodule. The extent of search is only confined to boundaries of targets, being extracted from their row-ordered stacks.

3.5.3.8. SMOD 38- "PLO" (location)

The general expression for coordinates of centroid (x,y) of a pattern f(x,y) in terms of its moments is

$$\bar{x} = \frac{M_{o1}}{M_{oo}} \quad , \quad \bar{y} = \frac{M_{1o}}{M_{oo}}$$

where

$$M_{mn} = \iint x^m y^n f(x,y)dxdy$$

M_{oo} is the area of the pattern;

M_{o1} is the first moment of area about y-axis;

M_{1o} is the first moment of area about the y-axis.

[x and y, above, correspond to the continuous domain while i and j are the corresponding dicrete domain axes.]

Area and moments, in CAVDEP, are computed via relations (3.5), (3.3) and (3.4). These relations have been formulated from methods described in [21].

$$(M_{o1})_r = \sum_{K\epsilon\tau_r} \{\alpha_K (i_K)^2 + \beta_K\} \mid r = 1,\ldots,z \qquad (3.3)$$

where

$$\alpha_K = \begin{cases} \frac{1}{2} & j_K > j_{K-1} \\ 0 & j_K = j_{K-1} \\ -\frac{1}{2} & j_K < j_{K-1} \end{cases}$$

$$\beta_K = \begin{cases} & i_K < i_{K-1},\ j_K > j_{K-1} \\ \frac{1}{2} & i_K > i_{K-1},\ j_K < j_{K-1} \\ 0 & i_K = i_{K-1} \\ & j_K = j_{K-1} \\ & i_K < i_{K-1},\ j_K < j_{K-1} \\ -\frac{1}{2} & j_K > j_{K-1},\ j_K > j_{K-1} \end{cases}$$

$$(M_{o1})_r = \sum_{K\epsilon\tau} \{\alpha_K (j_K)^2 + \beta_K (j_K)\} \mid r = 1,\ldots,z \qquad (3.4)$$

where

$$\alpha_K = \begin{cases} \frac{1}{2} & i_K < i_{K-1} \\ 0 & i_K = i_{K-1} \\ -\frac{1}{2} & i_K > i_{K-1} \end{cases}$$

$$\beta_K = \begin{cases} & i_K > i_{K-1},\ j_K > j_{K-1} \\ \frac{1}{2} & i_K < i_{K-1},\ j_K < j_{K-1} \\ & \\ 0 & \begin{matrix} i_K = i_{K-1} \\ j_K = j_{K-1} \end{matrix} \\ & \\ & i_K > i_{K-1},\ j_K < j_{K-1} \\ -\frac{1}{2} & i_K < i_{K-1},\ j_K > j_{K-1} \end{cases}$$

Hence the coordinates of the centroid (i_{cr}, j_{cr}) of target τ_r are determined from

$$i_{cr} = \frac{(M_{o1})_r}{(AA)_r}, \quad j_{cr} = \frac{M_{1o}}{(AA)_r} \mid r = 1,\ldots,z$$

3.6.3.9. SMOD39- "POR" (orientation) The traditional method of computation of orientation which has received widespread recognition is due Hu [22], and may be summarised as follows:

$$\theta = \tfrac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{2o} - \mu_{o2}}\right)$$

where

$$\mu_{mn} = \int\int_{-\infty}^{\infty} (x-\bar{x})^m (y-\bar{y})^n f(x,y)\, dxdy$$

Although determination of orientation as shown above does not demand a prior knowledge of target identity, but when it comes to utilising this information it must obviously be related to the target identity.

In CAVDEP computation of orientation is pursued as a post identification process, and not via computation of moments, as in above. Determination of orientation in CAVDEP is viewed as a location of at least two specific points of targets. Hence two peculiar points of the target, e.g. a sharp corner and the centroid, or centre of a hole and centroid can reveal the orientation of the target. Perception of orientation of rectangular shapes can alternatively be obtained via their orthogonal projections (as described in chapter 5), if the scene complexity permits.

Thus perception of orientation and the nature of its presentation remains target dependent.

3.5.4. MOD 4- "FEATRX" (feature extraction)

This module is based on two submodules, the first of which is concerned with computation of user-selected feature elements, while the other forms the feature vector, again in accordance with the choice of the user.

3.5.4.1 SMOD 41- "FFS" (feature selection)

This submodule supports 10 commands, each of which compute a predefined feature element, for the targets of a segmented scene.

The features and the commands that effect their computation are:

(1) Area (AA)- **FFSAA**

(2) Perimeter (PR)- **FFSPR**

(3) Number of sharp corners (SC)- **FFSSC**

(4) Minimum radial length (R1)- **FFSR1**

(5) Maximum radial length (R2)- **FFSR2**

(6) Number of holes (HN)- **FFSHN**

(7) Compactness (CP), Area/(Perimeter) - **FFSCP**

(8) R2/R1 (RR)- **FFSRR**

(9) Σ (number of points), extracted from projections, explained in chapter 5- **FFSSG**

(10) ψ - defined in chapter 5, only applicable to specific cases- **FFSSI**

Computation of area in **FFSAA** proceeds according to

$$(AA)_r = \sum_{K \epsilon \tau_r} (\alpha_K . i_K + \beta_K) \mid r = 1,\ldots,z \qquad (3.5)$$

where

$$\alpha_K = \begin{cases} 1 & j_K > j_{K-1} \\ 0 & j_K = j_{K-1} \\ -1 & j_K < j_{K-1} \end{cases}$$

$$\beta_K = \begin{cases} \frac{1}{2} & i_K < i_{K-1},\ j_K > j_{K-1} \\ & i_K > i_{K-1},\ j_{K-1} < j_{K-1} \\ 0 & i_K = i_{K-1} \\ & j_K = j_{K-1} \\ -\frac{1}{2} & i_K < i_{K-1},\ j_K < j_{K-1} \\ & i_K > i_{K-1},\ j_K > j_{K-1} \end{cases}$$

FFSPRQ computes the perimeter as follows

$$(PR)_r = \sum_{K \epsilon \tau r} \alpha_K \quad | \quad r = 1,\ldots,z$$

where

$$\alpha_K = \begin{cases} 1 & i_K = i_{K-1} \\ & j_K = j_{K-1} \\ \sqrt{2} & i_K \neq i_{K-1},\ j_K = j_{K-1} \end{cases}$$

FFSSC An angular description of target external boundaries can either be derived during the segmentation process, or here as a precursor to feature selection. The relation

between the incident and the emanating vectors that link succesive boundary points reveals this information. Here, angles are graded in terms of their acuteness, and the information is retained in a look-up table.

In **FFSR1**, the minimum radial length **R1** is determined according to

$$R_1 = \min_{K \epsilon \tau_r} \left[(i_K - i_c)^2 + (j_K - j_c)^2\right]^{\frac{1}{2}}$$

$$\Leftrightarrow \min_{K \epsilon \tau_r} \left[(i_K - i_c)^2 + (j_K - j_c)^2\right]$$

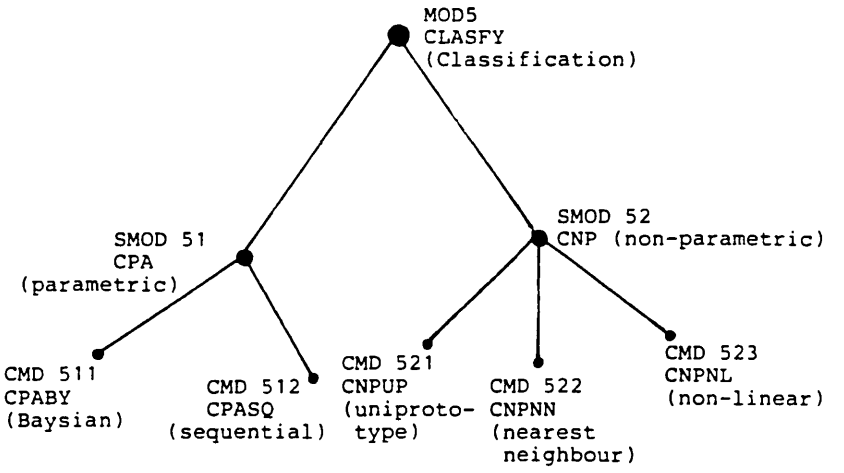
Similarly **FFSR2** computes R2 as follows

$$R_2 = \max_{K \epsilon \tau_r} \left[(i_K - i_c)^2 + (i_K - i_c)^2\right]^{\frac{1}{2}}$$

$$\Leftrightarrow \max_{K \epsilon \tau_r} \left[(i_K - i_c)^2 + (i_K - i_c)^2\right]$$

3.5.4.2. SMOD 42- "FFV" (feature vector formation) The principal command of this submodule is invoked to form the feature vector, in accordance with the choice of the user. The order of the selected feature elements are preserved, as some classifiers are sensitive to this order.

3.5.5. Module 5- "CLASFY" (classification)



Structure of classification module

3.5.5.1. SMOD 51- "CPA" (parametric classification)

In this section two parametric classification algorithms are presented. Both aim at reducing the on-line computation effort (i.e. that expended during the actual classification) to a minimum. The first algorithm, i.e. that underlying CPABY is a derivative of the Baysian decision rule (addressed in section 1.1.3.2). Class assignment in this algorithm progresses according to

$$D(\underline{\varphi}) = D_i \text{ if}$$

$$f(\omega_i|\underline{\varphi}) = \max_{k = 1,\ldots,M} [f(\omega_k|\underline{\varphi})] \qquad (3.10)$$

where

$$f(\omega_k|\underline{\varphi}) = \sum_{i=1}^{N} \{\varphi_i \frac{\hat{E}[\Phi_i^{(k)}]}{\sigma_i^2} - \tfrac{1}{2} \frac{(\hat{E}[\Phi_i^{(k)}]^z}{\sigma_i^2}\} \quad (3.11)$$

To justify the above, and to elucidate the underlying assumptions, that are also shared by the second algorithm of this section, relation (3.10) is derived from a well established datum, i.e. the Baysian decision rule.

Restating (1.5), i.e. classification with the Baysian decision rule

$$D(\underline{\varphi}) = D_i \text{ if}$$

$$R(\omega_i|\underline{\varphi}) = \min_{k=1,\ldots,M} [R(\omega_k|\underline{\varphi})]$$

$$= \min_{k=1,..,M} \left[\sum_{j=1}^{M} \lambda(\omega_k|\omega_j)\, P(\omega_j|\underline{\varphi})\right] \quad (3.12)$$

$$P(\omega_j|\underline{\varphi}) = \frac{P(\underline{\varphi}|\omega_j)P(\omega_j)}{\sum_{k=1}^{M} p(\underline{\varphi}|\omega_k)\, P(\omega_k)} = \frac{p(\underline{\varphi}|\omega_j)\, P(\omega_j)}{p(\underline{\varphi})}$$

To make the computation tractable, the following simplifying assumptions are made:

Assumption 1 Equal losses are allowed for all misclassification, i.e.

$$[\lambda(\omega_k|\omega_j)] = 1 \quad k \neq j; k,j = 1,\ldots,M$$

$$[\lambda(\omega_k|\omega_j] = 0 \quad k = j = 1,\ldots,M$$

Assumption 2 Equal probability of occurence for all

patterns, i.e.

$$P(\omega_1) = p(\omega_2) = ... = P(\omega_M) = p(\omega)$$

Assumption 3 Independent feature elements in the feature vector, i.e.

$$p(\underline{\varphi}|\omega_k) = \prod_{i=0}^{N} p(\varphi_i|\omega_k) \mid k = 1,...,M$$

Assumption 4 Equal variances for feature element variability for all classes of concern, i.e.

$$\sigma_i^{(1)} = \sigma_i^{(2)} = ... = \sigma_i^{(M)} = \sigma_i \mid i = 1,...,N$$

Assumption 5 Gaussian pdf associated with distribution of each feature element, in all classes of concern, i.e.

$$p(\varphi_i|\omega_k) = \frac{1}{\sigma_i\sqrt{2\Pi}} e^{-(\varphi_i - E[\Phi_i^{(k)}])^2/2\sigma_i^2}$$

The above assumptions reduces equation (3.12) to

$$R(\omega_i|\underline{\varphi}) = \frac{P(\omega)}{p(\underline{\varphi})} \cdot \min_{k=1,\ldots,M} \left[\sum_{j=1}^{M} \lambda(\omega_k|\omega_j)\, p(\underline{\varphi}|\omega_j) \right]$$

(due to Assumption 2)

$$= \frac{P(\omega)}{p(\underline{\varphi})} \cdot \min_{k=1,\ldots,M} \left[\sum_{j=1}^{M} p(\underline{\varphi}|\omega_j) - p(\underline{\varphi}|\omega_k) \right]$$

(due to Assumption 1)

$$= \frac{P(\omega)}{p(\underline{\varphi})} \cdot \sum_{j=1}^{M} p(\underline{\varphi}|\omega_j) + \frac{P(\omega)}{p(\underline{\varphi})} \min_{k=1,\ldots,M} \left[-p(\underline{\varphi}|\omega_k)\right]$$

$$\Leftrightarrow \min_{k=1,\ldots,M} \left[-p(\underline{\varphi}|\omega_k)\right]$$

$$= \max_{k=1,\ldots,M} \left[p(\underline{\varphi}|\omega_k)\right]$$

$$= \max_{k=1,\ldots,M} \left[\prod_{i=1}^{N} p(\varphi_i|\omega_k) \right] \qquad \text{(due to Assumption 3)}$$

$$\Leftrightarrow \max_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \ln p(\varphi_i|\omega_k) \right]$$

$$= \max_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \left\{ \ln\left(\frac{1}{\sigma_i\sqrt{2\Pi}}\right) - \frac{1}{2\sigma_i^2} \left(\varphi_i - E[\Phi_i^{(k)}]\right)^2 \right\} \right]$$

$$= \sum_{i=1}^{N} \ln \frac{1}{\sigma_i\sqrt{2\Pi}} + \max_{k=1,\ldots,M} \left[- \sum_{i=1}^{N} \frac{\left(\varphi_i - E[\Phi_i^{(k)}]\right)^2}{2\sigma_i^2} \right]$$

$$\Leftrightarrow \min_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \frac{\left(\varphi_i - E[\Phi_i^{(k)}]\right)^2}{2\sigma_i^2} \right] \qquad (3.13)$$

$$= \min_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \frac{\varphi_i^2 - 2\varphi_i E[\Phi_i^{(k)}] + (E[\Phi_i^{(k)}])^2}{2\sigma_i^2} \right]$$

$$= \sum_{i=1}^{N} \frac{\varphi_i^2}{2\sigma_i^2} + \min_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \frac{2\varphi_i E[\Phi_i^{(k)}] - (E[\Phi_i^{(k)}])^2}{2\sigma_i^2} \right]$$

$$= \max_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \frac{\varphi_i E[\Phi_i^{(k)}] - \frac{1}{2}(E[\Phi_i^{(k)}])^2}{\sigma_i^2} \right]$$

$$= \max_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \varphi_i \frac{\hat{E}[\Phi_i^{(k)}]}{\hat{\sigma}_i^2} - \frac{1}{2} \frac{(\hat{E}[\Phi_i^{(k)}])^2}{\hat{\sigma}_i^2} \right] \qquad (3.14)$$

$$= \max_{k=1,\ldots,M} [f(\omega_k|\underline{\varphi})] = f(\omega_i|\underline{\varphi})$$

where

$$\hat{E}[\Phi_i^{(k)}] = \frac{1}{n} \sum_{\ell=1}^{n} \Phi_{i\ell}^{(k)} \; ; \; \hat{\sigma}_i^2 = \frac{1}{n} \sum_{\ell=1}^{n} (\Phi_i^{(k)2} - \hat{E}[\Phi_i^{(k)}])^2$$

The terms $\frac{\hat{E}[\Phi_i^{(k)}]}{\hat{\sigma}_i^2}$ and $\frac{(E[\Phi_i^{(k)}])^2}{\hat{\sigma}_i^2}$ can be computed in the course of the training, which is an off-line process. Hence, expression (3.14) merely demands an on-line computation effort of

(N x multiplication + N x additions) per class.

This compares favourably with (3.13) that needs a comparative computation of

(N x subtraction + N x multiplication + N x divisions) per class. [(3.13) has been used by some investigators].

The algoritm described in the remaining part of this section is an extension of Probabilty Ratio Test (SPRT), [23,24]. This classification scheme is inherently a two-class test which through repeated use can be applied to multicatagory recognition problems.

Let $\underline{\varphi n} = (\varphi_1, \varphi_2, \ldots, \varphi_n)^T$ be the feature vector representing the sample pattern, and

$$\Lambda_1 = \frac{P(\omega_1|\varphi_1)}{P(\omega_2|\varphi_1)}$$

define the likelihood ratio in the first stage of a SPRT. For class assignment of the sample patterns at this and subsequent stages, an upper and a predefined lower bound (Λ^u , Λ^L) for Λ_i, i=1,...,n ,are introduced. Λ^u and Λ^L can readily be deduced from allowable classification error probabilities imposed by the nature of the recognition problem.

$$\Lambda^u = \frac{1 - P_{err}(\omega_2|\omega_1)}{P_{err}(\omega_1|\omega_2)}$$

$$\Lambda^L = \frac{P_{err}(\omega_2|\omega_1)}{1 - P_{err}(\omega_1|\omega_2)}$$

SPRT leads to a conclusive decision when the test outcome is

$$\Lambda_1 \geq \Lambda^u \quad \Rightarrow \quad D(\varphi_1) = D_1$$

or

$$\Lambda_1 \leq \Lambda^L \quad \Rightarrow \quad D(\varphi_1) = D_2$$

in which case the test terminates. Otherwise, i.e. when

$$\Lambda^L < \Lambda < \Lambda^u$$

the subsequent feature element, in the feature vector, is sought to augment the initial element, and a second test stage is entered. SPRT can proceed either until a conclusive decision is reached, or can cease after a predefined number of iterations. Λ^u and Λ^L are also known to have been modified dynamically, by some investgators, to expedite convergence [25].

With the previous assumptions still prevailing, Λ_1 may be expressed as

$$\Lambda_1 = \frac{\left(\frac{1}{\sigma_1 \sqrt{2\Pi}}\right) e^{-(\varphi_1 - E[\Phi_1^{(1)}])^2/2\sigma_1^2}}{\left(\frac{1}{\sigma_1 \sqrt{2\Pi}}\right) e^{-(\varphi_1 - E[\Phi_1^{(2)}])^2/2\sigma_1^2}}$$

$$\ln \Lambda_1 = \frac{1}{2\sigma_1^2}\left[2\varphi_1\left(E\left[\Phi_1^{(1)}\right] - E\left[\Phi_1^{(2)}\right]\right)\right.$$

$$\left. - \left\{\left(E\left[\Phi_1^{(1)}\right]\right)^2 - \left(E\left[\Phi_1^{(2)}\right]\right)^2\right\}\right]$$

$$= \frac{E\left[\Phi_1^{(1)}\right]-E\left[\Phi_1^{(2)}\right]}{\sigma_1^2}\left[\varphi_1-\tfrac{1}{2}\left(E\left[\Phi_1^{(1)}\right]+E\left[\Phi_1^{(2)}\right]\right)\right] \qquad (3.15)$$

From Eq.(3.15) the limiting values of φ_1 (φ_1^u, and φ_1^L) which correspond to $\Lambda_1 = \Lambda^u$ and $\Lambda_1 = \Lambda^L$ respectively are found to be

$$\varphi_1^u = \frac{\sigma_1^2}{E\left[\Phi_1^{(1)}\right]-E\left[\Phi_1^{(2)}\right]} \ln \Lambda^u + \tfrac{1}{2}\left(E\left[\Phi_1^{(1)}\right]+E\left[\Phi_1^{(2)}\right]\right) \qquad (3.16)$$

$$\varphi_1^L = \frac{\sigma_1^2}{E\left[\Phi_1^{(1)}\right]-E\left[\Phi_1^{(2)}\right]} \ln \Lambda^L + \tfrac{1}{2}\left(E\left[\Phi_1^{(1)}\right]+E\left[\Phi_1^{(2)}\right]\right) \qquad (3.17)$$

In the course of training, based on acceptable error probabilities, Λ^u and Λ^L are introduced. Then from relation (3.16) and (3.17), φ_1^u and φ_1^L are computed. This reduces the task at the initial classification stage to enforcement of the following decision rule:

$$D(\varphi_1) = D_1 \qquad \text{if} \qquad \varphi_1 \geq \varphi_1^u$$

$$D(\varphi_1) = D_2 \qquad \text{if} \qquad \varphi_1 \leq \varphi_1^{L}$$

However, with $\varphi_1^{L} < \varphi_1 < \varphi_1^{u}$, a subsequent test stage is entered. To obtain a general expression, the n th stage test is considered.

$$\Lambda_n = \frac{P(\omega_1|\underline{\varphi}_n)}{P(\omega_2|\underline{\varphi}_n)}$$

$$= \frac{p(\underline{\varphi}_n|\omega_1).P(\omega_1)}{p(\underline{\varphi}_n|\omega_1).P(\omega_2)} \qquad \text{from Equation (1.5)}$$

$$= \frac{p(\underline{\varphi}_n|\omega_1)}{p(\underline{\varphi}_n|\omega_2)} \qquad \text{(from Assumption 2)}$$

$$= \frac{\prod_{i=1}^{N} p(\varphi_i|\omega_1)}{\prod_{i=1}^{N} p(\varphi_i|\omega_1)} \qquad \text{(from Assumption 3)}$$

$$= \frac{p(\varphi_1|\omega_1)}{p(\varphi_1|\omega_2)} \cdot \frac{p(\varphi_2|\omega_1)}{p(\varphi_2|\omega_2)} \cdot \cdots \cdot \frac{p(\varphi_n|\omega_1)}{p(\varphi_n|\omega_2)}$$

From Equation (3.15)

$$\ln \Lambda_n = \frac{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]}{\sigma_1^2} \{\varphi_1 - \tfrac{1}{2}(E[\Phi_1^{(1)}] + E[\Phi_1^{(2)}])\}$$

$$+ \frac{E[\Phi_2^{(1)}] - E[\Phi_2^{(1)}]}{\sigma_2^2} \{\varphi_2 - \tfrac{1}{2}(E[\Phi_2^{(1)}] + E[\Phi_2^{(2)}])\}$$

$$+ \cdots\cdots$$

$$+ \ldots\ldots$$

$$+ \frac{E[\Phi_n^{(1)}] - E[\Phi_n^{(2)}]}{\sigma_n^2} \{\varphi_n - \tfrac{1}{2}(E[\Phi_n^{(1)}] + E[\Phi_n^{(2)}])\} \qquad (3.18)$$

Division by $\dfrac{E[\Phi_n^{(1)}] - E[\Phi_n^{(2)}]}{\sigma_1^2}$ of Equation (3.18), and

introduction of

$$w_n = \frac{\sigma_1^2}{\sigma_n^2} \cdot \frac{E[\Phi_n^{(1)}] - E[\Phi_n^{(2)}]}{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]} \qquad (3.19)$$

and

$$S_n = \tfrac{1}{2}(E[\Phi_n^{(1)}] + E[\Phi_n^{(2)}]) \qquad (3.20)$$

yields

$$\frac{\sigma_1^2}{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]} \ln \Lambda_n = w_1\varphi_1 + w_2\varphi_2 + \ldots + w_n\varphi_n - (w_1s_1 + w_2s_2 + \ldots + w_ns_n)$$

w_i, i=1,..., n may be viewed as weightings for successive feature elements. Now, if a weight vector is defined as

$$\underline{w}_n = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \quad \text{and} \quad \underline{s}_n = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$$

Then (3.18) may be expressed as

$$\frac{\sigma_1^2}{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]} \ln \Lambda_n = \underline{w}_n^T \underline{\varphi}_n - \underline{w}_n^T \underline{S}_n \qquad (3.21)$$

Rearranging Equation (3.21)

$$\underline{w}^T \underline{\varphi}n = \frac{\sigma_1^2}{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]} \ln \Lambda_n + \underline{w}_n^T \underline{S}_n \qquad (3.22)$$

As before, the limiting values of $\underline{\varphi}n$,(i.e. $\underline{\varphi}_n^u$ and $\underline{\varphi}_n^L$), that yield $\Lambda_n = \Lambda^u$ and $\Lambda_n = \Lambda^L$ respectively, are obtained from Eq.(3.21) as

$$\underline{w}_n^T \underline{\varphi}_n^u = \frac{\sigma_1^2}{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]} \ln \Lambda^L + \underline{w}_n^T \underline{S}_n \qquad (3.23)$$

$$\underline{w}_n^T \underline{\varphi}_n^L = \frac{\sigma_1^2}{E[\Phi_1^{(1)}] - E[\Phi_1^{(2)}]} \ln \Lambda^u + \underline{w}_n^T \underline{S}_n \qquad (3.24)$$

Hence classification at the nth stage of SPRT will proceed according to

$$\underline{w}_n^T \underline{\varphi}_n \geq \underline{w}_n^T \underline{\varphi}_n^u \Rightarrow D(\underline{\varphi}_n) = D_1 \qquad (3.25)$$

$$\underline{w}_n^T \underline{\varphi}_n \leq w^T \underline{\varphi}_{nL} \quad \Rightarrow \quad D(\underline{\varphi}_n) = D_2 \tag{3.25}$$

$$\underline{w}_n^T \underline{\varphi}_n^L \leq \underline{w}_n^T \underline{\varphi}_n \leq \underline{w}_n^T \underline{\varphi}_n^u \quad \Rightarrow \quad \text{A subsequent higher ordered test.}$$

[$\underline{w}_n^T \underline{\varphi}_n$ is the sum of the weighted version of the successive feature elements.]

The result obtained above is a general expression and remains valid for all stages of SPRT.

The compact algorithm presented through (3.19) to (3.25) differs from those reported in standard literature [23,24] . Their compact formula is based on excessive relaxation of contraints, namely extending the stated assumptions to

(i) $\sigma_1 = \sigma_2 = \ldots = \sigma_n$

(ii) $E[\Phi_1^{(k)}] = E[\Phi_2^{(k)}] = \ldots = E[\Phi_n^{(k)}] \mid k=1,2$

(iii) $E[\Phi_i^{(1)}] > E[\Phi_i^{(2)}] \mid i=1,\ldots,n$

As evident, the derived expressions, (3.19) to (3.25), are not based on such restrictive assumptions, and yet remain compact and almost equally efficient.

3.5.5.2. SMOD 52- "CNP" (nonparametric-classification)

The concept of measuring a distance in the feature space is common to all classifiers of this submodule. The metrics that govern such measurements are varied, a discussion of which is presented in Appendix B.

Distance measurement in classifiers of this submodule is performed either via City Block or Euclidean metrics.

CNPUP (a uniprototype classifier) is a minimum distance based classifier. It is intended for applications where the representing feature elements show little variability and cluster tightly around representive points, in the feature space.

With the City Block metric, classification proceeds as

$D(\underline{\varphi}) = D_i$ if

$$\Delta\left(\underline{\Phi}^{(i)},\underline{\varphi}\right) = \min_{k=1,\ldots,M} \left[\Delta\left(\underline{\Phi}^{(k)},\varphi\right)\right]$$

$$= \min_{k=1,\ldots,M} \left[\sum_{i=1}^{N} \left| \frac{\Phi_i^{(k)} - \varphi_i}{\frac{1}{M}\sum_{k=1}^{M} \Phi_i^{(k)}} \right|\right]$$

The "$\frac{1}{M}\sum_{k=1}^{M} \Phi_i^{(k)}$" term was introduced, in this work, as a heuristic measure to assign equal emphasis to all feature elements.

The multiprototype problem (in which feature vectors of the same class cluster tightly around a finite number of points, rather than a single point, in the feature space) can simply be considered as an M x n class uniprototype problem, where n is the number of clusters per class,(as depicted in Fig.3.5).

CNPNN (nearest neighbour)

Here classification is performed via 1-NN decision rule. In this classification scheme each class is represented by a finite number of points, which may be represented in terms of their constituent protype feature vector as:

$$\underline{\Phi}_j = (\Phi_{1j}, \Phi_{2j}, \ldots, \Phi_{Nj})^T \mid j = 1, \ldots, n$$

Classification with the City Block metric, here, proceeds as

$$D(\underline{\varphi}) = D_i \quad \text{if}$$

$$\Delta(\underline{\Phi}^{(i)}, \underline{\varphi}) = \min_{k = 1, \ldots, M} \left[\Delta(\underline{\Phi}_j^{(k)}, \underline{\varphi}); j = 1, \ldots, n\right]$$

$$\Delta\ (\underline{\Phi}^{(i)},\ \underline{\varphi}) = \min_{k\ =\ 1,\ldots,M} \left[\left(\sum_{i=1}^{N} \left| \frac{\Phi_i^{(k)} - \varphi_i}{\frac{1}{M} \sum_{k=1}^{M} \sum_{j=1}^{n} \Phi_{ij}^{(k)}} \right| \right); j = 1,\ldots,n \right]$$

" $\frac{1}{Mxn} \sum_{k=1}^{M} \sum_{j=1}^{n} \Phi_{ij}^{(k)}$" has been introduced in this work to assign equal emphasis to successive feature elements.

CNPNL (non-linear classification)

Here classification is based on

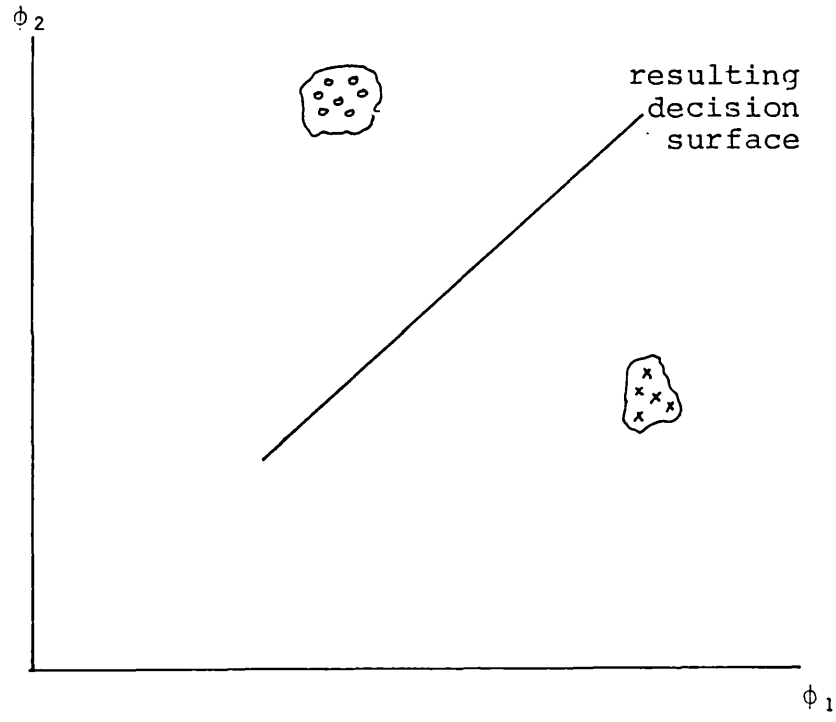
$D(\underline{\varphi}) = D_k$ if

$$\Delta_{NL}\ (\underline{\Phi}^{(k)}\ \underline{\varphi}) = \sum_{i=1}^{N} \Delta_{NL}\ (\Phi_i^{(k)},\ \varphi_i) = \emptyset$$

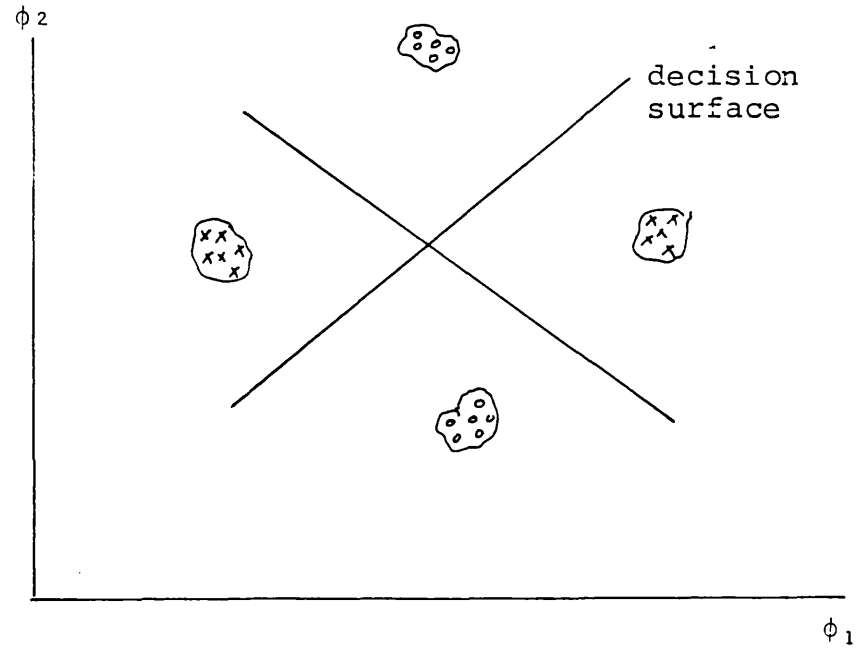
where

$$\Delta_{NL}\ (\Phi_i^{(k)},\ \varphi_i) = \begin{cases} 1 \text{ if } |\Phi_i^{(k)} - \varphi_i| > T_i \\ 0 \text{ if } |\Phi_i^{(k)} - \varphi_i| \leq T_i \end{cases}$$

This latter classification scheme progresses sequentially as far as feature elements are concerned. A variation of this classification scheme has been suggested for applications discussed in chapters 5 and 6.



a) Tightly packed clusters lending their selves to classification of CMD 521



b) Each class giving rise to a multitude (in their case two) tightly packed clusters.

Fig 3.5 Feature space representation of patterns suitable for uni and multi-prototype classification schemes.

3.5.6. MOD 6- "TRG" (training)

The training commands are concerned with provision of those parameters that are required for classification. These parameters are initially extracted from the training samples and the user; the user acts as the error free source in supervised training, as depicted in Fig.1.3. The type of parameters required by various classifiers may differ. This leads to supporting each classifier type by a dedicated training command. The training data for parametric classifiers extends to the following:

(1) number of training samples observed per (feature) element in each class, $n_i^{(k)}$.

(2) $\sum_{\ell=1}^{n_i^{(k)}} \Phi_{i\ell}^{(k)}$

(3) $\sum_{\ell=1}^{n_i^{(k)}} [\Phi_{i\ell}^{(k)}]^2$

(4) $\hat{E}\,[\Phi_i^{(k)}] = \frac{1}{n_i^{(k)}} \sum_{\ell=1}^{n_i^{(k)}} \Phi_{i\ell}^{(k)}$

(5) $\hat{E}\,[(\Phi_i^{(k)})^2] = \frac{1}{n_i^{(k)}} \sum_{\ell=1}^{n_i^{(k)}} [\Phi_{i\ell}^{(k)}]^2$

(6) $[\sigma_i^{(k)}]^2 = E\,[(\Phi_i^{(k)})^2] - (E\,[\Phi_i^{(k)}])^2$

From these data the terminal parameters required by the specific classification commands are computed.

The composition of the training data for non-parametric classifiers changes to the number of training samples

observed, $n_i^{(k)}$, and the actual sample values, $\Phi_{i\ell}^{(k)}$, $\ell = 1,\ldots,n_i^{(k)}$.

The training data are organised in a virtual 3-dimensional array, in the training buffer. The first two entries for accessing each block (set) of data are the order of element of interest (in the feature vector), "i" and the associated class number "k". "i" and "k" are linked through successive training samples, with the user acting as a verifier, as the training progresses.

Once the training data is compiled for a specific problem, it can be stored on the disk. The prerecorded data can subsequently be used directly or augmented by addititional training samples.

3.5.7. MOD 7- "VLUATE" (evaluation)

This module hosts only three commands. All of which are concerned with evaluation of feature elements as described below.

VFEVR (varibility)

This command determines the variability of a feature element, as a measure of its reliability. The parameter measured is the variance of the respective feature element in classes of concern. This command is almost identical to the training command which supports parametric classifiers.

VFEDI (discrimination)

This command provides an indication of the discrimination power of a given feature element in two classes ω_i , ω_j and is based on computation of

$$\Delta^2(\Phi, \omega_i, \omega_j) = \frac{|E[\Phi^{(i)}] - E[\Phi^{(j)}]|^2}{[\sigma^{(i)}]^2 + [\sigma^{(j)}]^2}$$

The ingredients of the above function are those obtained through a parametric classifier training command. This command is very similar to the "parametric training command" and the operation leading to computation of the above function remains essentially the same. To enable use of prerecorded training files,the commands of this submodule makes use of the same virtual array 3-dimensional used by the training commands.

VFECO (correlation)

The correlation of the two feature elements Φ_i ,Φ_j are assessed via

$$\rho(\Phi_i, \Phi_j, \omega_k) = \frac{E\{(\Phi_i^{(k)} - E[\Phi_i^{(k)}])(\Phi_j^{(k)} - E[\Phi_j^{(k)}])\}}{\sigma_i^{(k)} \sigma_j^{(k)}}$$

$$\Leftrightarrow \quad \frac{|E[\Phi_i^{(k)} \Phi_j^{(k)}] - E[\Phi_i^{(k)}] . E[\Phi_j^{(k)}]|^2}{[\sigma_i^{(k)}]^2 \; [\sigma_j^{(k)}]^2}$$

3.6. REMARKS

(1) CAVDEP is suitable for prototyping robotic vision systems, as opposed to being directly employed in industrial applications. The package can cope with a maximum of 5 targets in a 5 class environment. It is intended for the use of the vision system designer.

(2) Compared to similar interactive systems that are in high level, CAVDEP remains a step nearer to implementation, as its assembly coded routines can readily be commited to firmware. When its routines do not satisfy the speed requirement of a specific application, there and then, CAVDEP reduces to a test bed for formulating the overall algorithm, as those other systems that are implemented in high level. Introduction of new commands to the package can be achieved with minimal effort, by virtue of the package modularity. Nevertheless, development of new commands in assembly language is tedious, and requires initimate familiarity with assembly coding and the machine structure. To ease expansion, the command routines themselves remain highly modular, and in many cases new commands can be created by minor modification of an existing command or addition of specific subroutines to it. Modularity to this extent is believed can assist transfer of the package to other classes of machines , particularly the M68Ø0Ø micocomputer. Should the current digitiser be replaced by a faster digitiser the change will be confined to "representation module", and

a very limited modification to the commands of "segmentation submodule", should the new digitiser adopt higher quantisation levels.

(3) Using the Batchelor's analogy [17], CAVDEP may be viewed as "bag of tools"; it may suffice one application while may need to be extended for another. Applications are predominantly signified by the nature of target and scene complexities that they entail. The target and scene complexities that CAVDEP can cope with, are grossly affected by the inter-target disparity; disparity is assessed in terms of the package features. However, it is primarily designed to confront the following target complexities:

- planar;
- almost lightly textured;
- with or without interior non-homogeneities;
- convex or concave;

multiple stable positions (not exceeding the 5-target segmentation capacity).

Nonetheless, with sufficiently dissimilar targets the package capability may extend to non-planar targets.

As far as scene complexities are concerned, CAVDEP essentially expects scenes that are uniformly illuminated by a controlled source. The juxtaposition of targets in such a scene can fall within any of the following catagories; the effort expended in coping

with these situations increases with the complexity.

- a single target per scene;

- a single target per predefined disjointed window, not exceeding 5 in a scene;

- a maximum of 5 randomly appearing targets within disjointed minimum bounding rectangles;

- 5 randomly appearing isolated targets.

However, sufficiently dissimilar targets may be segmented and even be identified in touching or mildly overlapping situations. This is a modest capability and is confined to whatever can be achieved by sequential application of PBIBG, and PBISH, both described in section 3.5.3.3.

(5) Segmentation in scenes containing randomly appearing targets is based on target boundary tracking. As a prelude to segmentation, the image is either binary thresholded or converted into a gradient image. In the latter case, the operators used do not span beyond a 3x3 neighbourhood. This hinders edge detection performance when the actual edge is wider than the width of the operator, and more elaborate techniques are needed to cope with such cases. The manner in which segmentation commands avoid detection of internal edge

lines when searching for initial external boundary points is believed to be peculiar to CAVDEP. This feature which is based on prior accumulation of boundary coordinates in a row-ordered array grossly facilitates within boundary operations, i.e. segmentation of holes and protrusions.

(6) When the juxtaposition of targets lends itself to projection set operation (discussed in chapter 5), targets can be located via their median centre, as opposed to centroid. Location via median centre is insensitive to the distance of noise blobs from the actual centroid. Moreover, if the blobs occurs beyond the minimum bounding rectangle of the pattern concerned, susceptibility to noise blob area is also removed. This is due to the noise reduction measure taken in PPSPS command (discussed in chapter 5).

(7) The parametric classifiers, incorporated in CAVDEP, assume a Gaussian functional for the conditional pdf's of the successive feature elements in classes of concern. Verification of this assumption is an exercise which currently cannot take place within CAVDEP. Non-parametric classifiers are intended for use when such assumptions are ruled out.

As evident from the theoretical results, classification performance will be more dependent upon

the typicality of the training samples, than the actual type of classifier. Atypical training samples can degrade classification drastically. Their effect on the uni-prototype classifier (CNPUP, section 3.5.5.2) is paramount, as this classifier relies on a single training sample to represent each class. To avoid taking in an unrepresentative training sample, in the problem of chapter 5, the median or mean of sample sets have been suggested as the prototype value. Sequential classifiers are likely to perform better in presence of atypical training samples, provided the affected feature element appears further down in the composition of the feature vector.

(8) The first parametric classifier presented in section 3.5.5.1 takes a step in reducing the on-line computational effort. However, its significance will become apparent only when the feature vector is of high dimensionality, far beyond the current state of CAVDEP.

(9) The compact parametric sequential classifiers presented in [23,24] are in fact a special case of corresponding expression dervived in this work.
$w = 1$, $S > \emptyset$, $i=1,\ldots,n$ reduces (3.23) to (3.25) to those of [23,24].

CHAPTER 4

MACHINE VISION IN AUTOMATED HANDLING OF ENGINE COMPONENTS

4.1. INTRODUCTION

This chapter presents a microprocessor based vision system that has been designed to bring automation to a material handling process in the automotive industry. The programmable manipulator here is a computer controlled 3-axes transfer gantry, and the targets of interest are engine bearing caps that are presented and treated in sets. The non-planar components have variable reflectance and appear against an equally uncomprimising background.

The tasks extend from the detection of the state of component sets at various sites including populated pallets, to location of the pallets. Based on a novel method, the vision system can furtner be used in automatic positional alignment of the loader arm, and calibration of its movement.

The vision system provides its information on real-time basis to the gantry controller.

A version of this vision system has been installed in the Ford Diesel Engine Assembly Plant (at Dagenham, Essex), development of which has been a collaborative effort between Imperial College and R.D.Projects Ltd. System level design, software design and implementation remained with with the author while the detailed hardware design and implementation was done by R.D.Projects.

4.1.1. Description of the handling operation

Fig.4.1 depicts the overall handling machinary. As evident, the gantry loader travels across a beam between two sites: the initial site (i.e. the machine tool off-load station), and the terminal site (i.e. the washing machine station).

The operation cycle starts by emergence of certain engine components (i.e. bearing caps) from a production line. The components are compiled, automatically, in sets of five, supposedly, in a regular fashion at the initial site. A typical component set is shown in Fig.4.2. The loader then accesses the initial site, picks up a component set and deposits it at the terminal station, and returns to the initial site. This constitutes a normal cycle.

The components subsequently go through another process, namely, an industrial wash, whose details are of no significance apart from the fact that its disruptions must not impact the rhythm of the production line. To cope with that eventuality a contingency scheme is introduced whereby:

(i) A pallet is introduced as reservoir, such that successive component sets can be taken to the pallet instead of the terminal station.

Each pallet accommodates 28 component sets in respective slot-sets, i.e. each component set will sit in a corresponding slot-set. The slot-sets are arranged in 4 rows, with rows running parallel to the

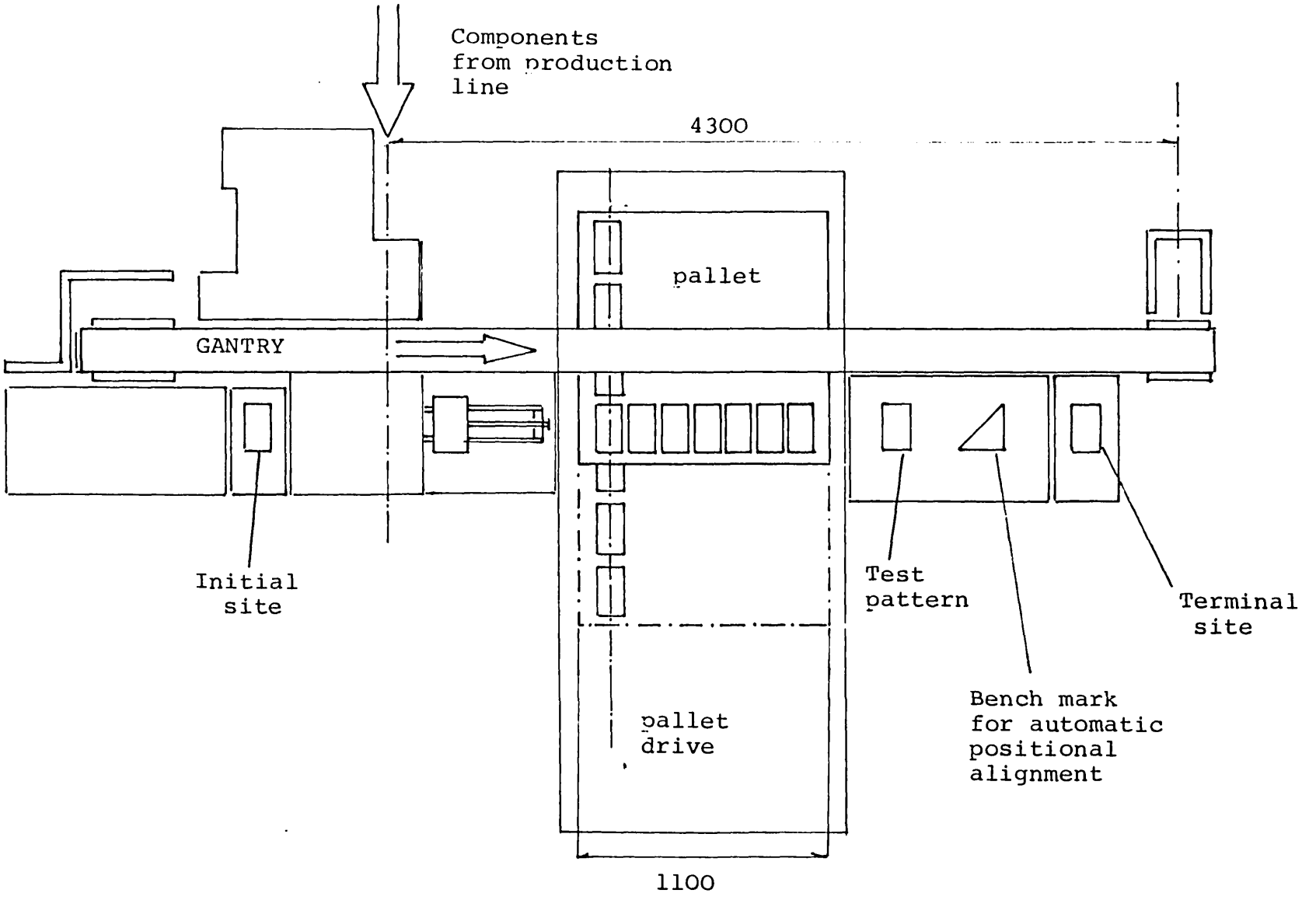


Fig. 4.1(a) The overall arrangement

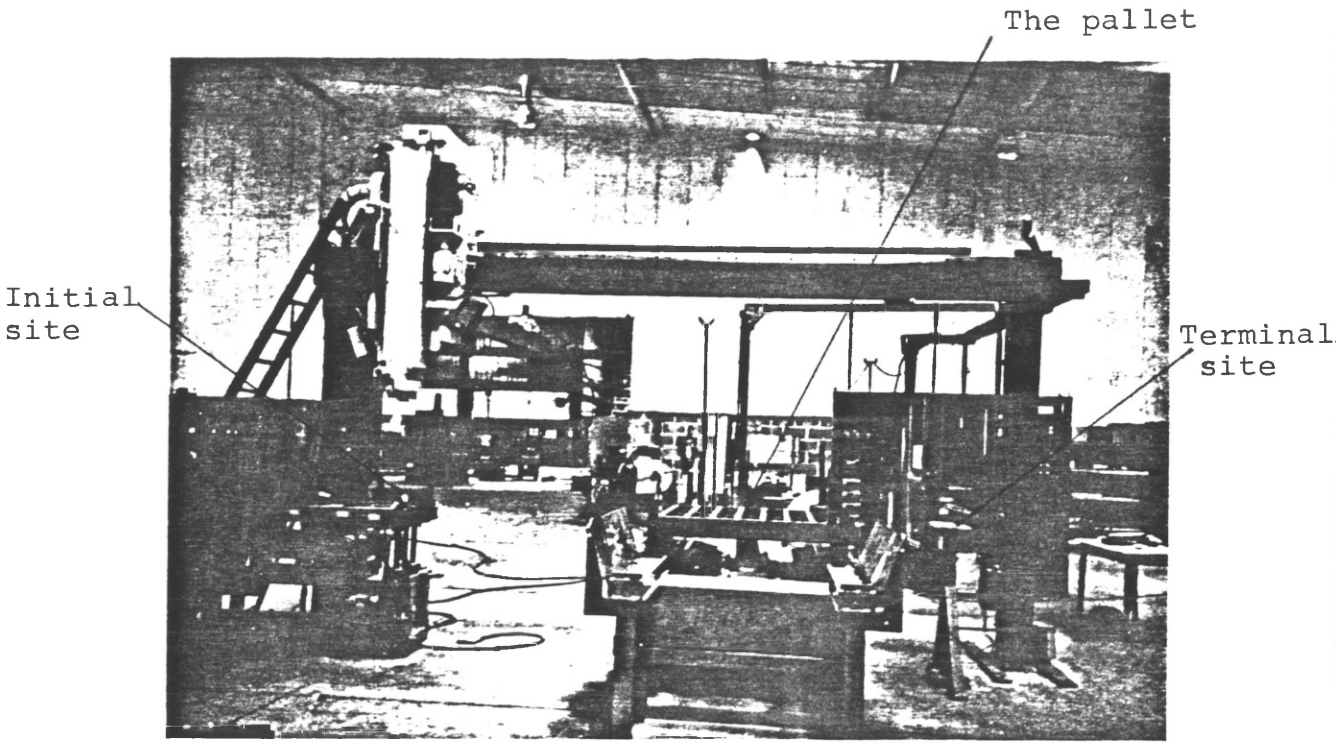


Fig. 4.1(b). The handling machinery

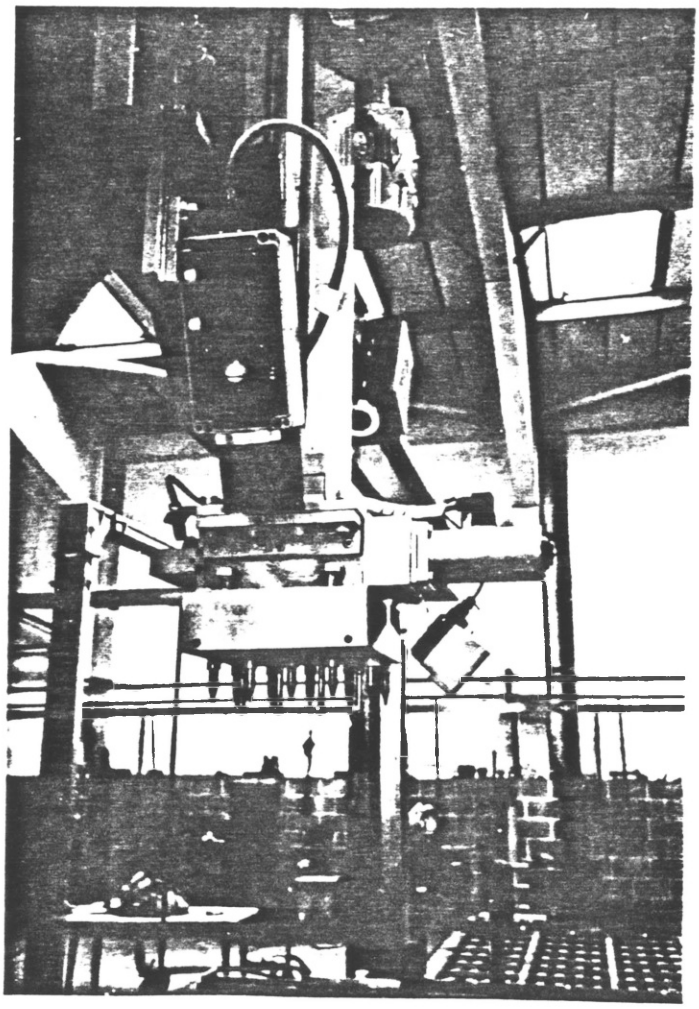


Fig. 4.1(c). The loader arm

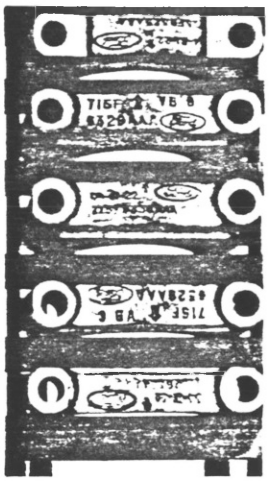


Fig. 4.2. A typical component-set

path of travel of the arm. A point worthy of note is that the loader arm can only access the row which lies immediately beneath its path. This row will be referred to as the "live row", and the rest as the "inactive rows". Fig.4.3 shows the pallet and the slot-set arrangement. Thus the sites that are attended by the loader arm during the loading cycle of the contingency phase are: the initial site, and the successive slot-sets in the live row of the pallet. [Throughout this chapter "loading" and "unloading" refers to the sites rather than the arm.]

(ii) Upon filling the 'live row', the pallet is repositioned to present the successive inactive rows as the 'live row'. When the first pallet is filled, a second pallet is introduced which goes through the same sequence.

(iii) Whenever normality is restored, the component sets are unloaded from the pallets and transferred to the terminal station by the loader arm. Thus during the unloading cycle, the loader attends the active row of the pallet as well as the initial and the terminal sites.

(iv) The pallets are stacked up on a transverse carriage which supports a maximum of 4 pallets. Appropriate pallet location is reached through automatic motion

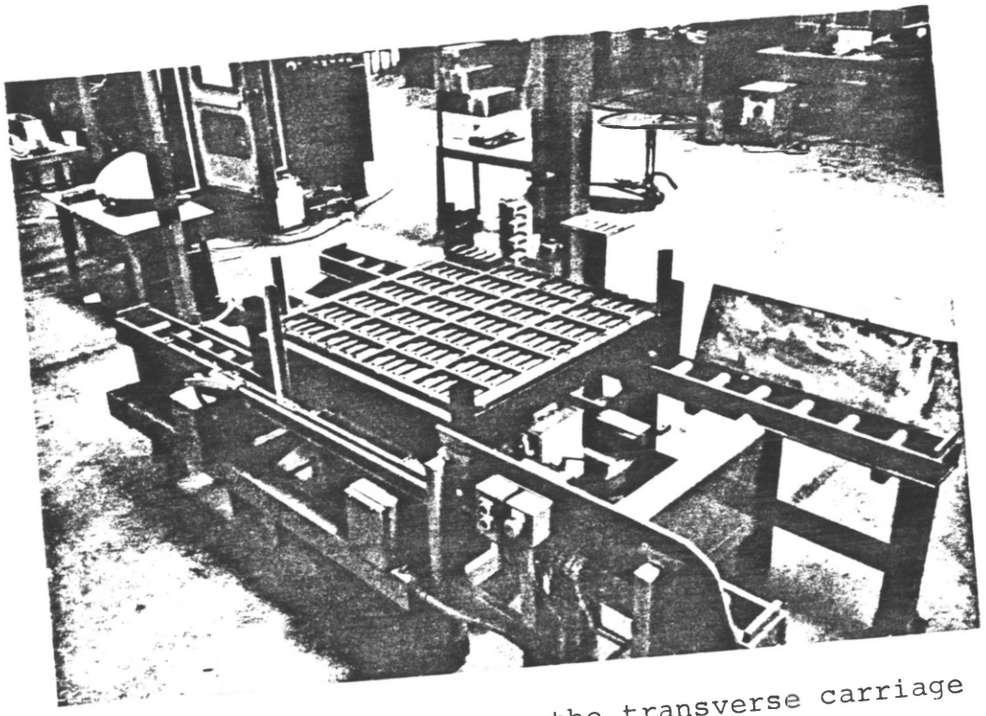


Fig. 4.3(a) The pallet on the transverse carriage

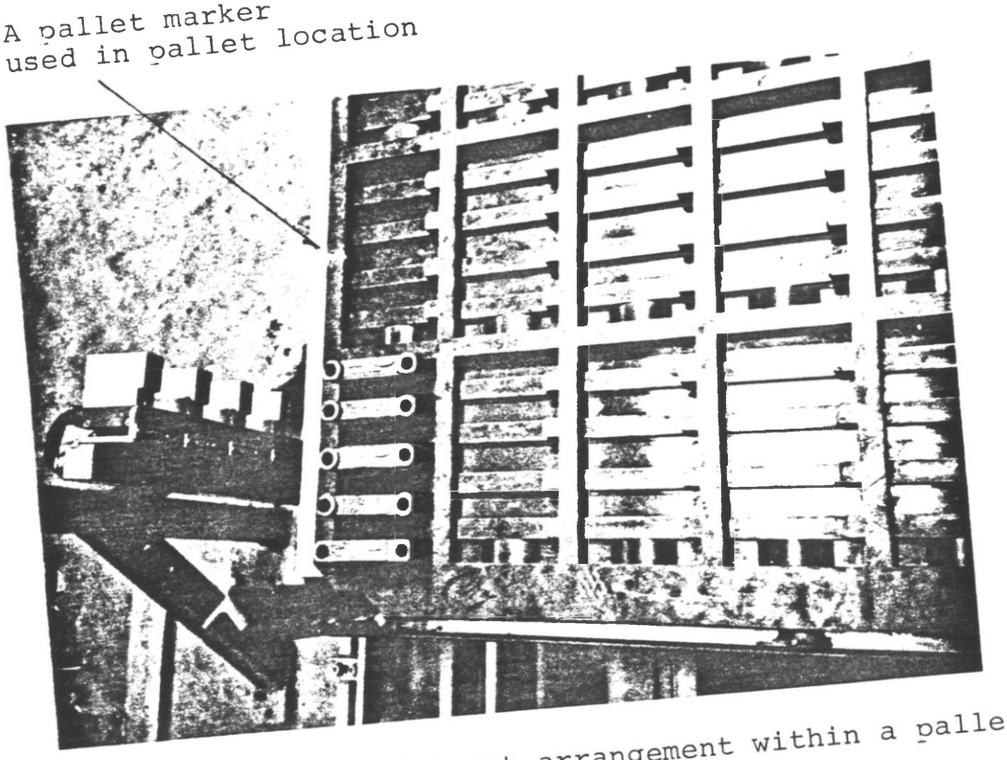


Fig. 4.3(b) The slot-set arrangement within a pallet

of the base carriage on a series of rollers.

4.1.2. Tasks to be performed by the vision system

(i) **Inspect the initial site** with a view to detect its state- the site must support an intact component set, i.e. five fully aligned components, prior to the accession of the arm.

(ii) **Inspect the terminal site** prior to deposition of new component set. The arm can only access this site if it is void of component.

(iii) **Recognise the state of successive slot-sets in the pallets,** i.e. determine whether they are empty, or hold an intact component set, or are partially full, or support one or more misplaced components, and in commensurate with the nature of the loading cycle either enable or inhibit arm accession.

(iv) **Locate the pallets-** Every time a pallet traverses to present a new row to the arm, it must assume a precise position. The vision system is tasked to verify this or detect and quantify any possible misalignment.

(v) **Positional alignment of the arm-** In the absence of an external [to the arm controller] source, the perceived position of the arm would be based on dead

reckoning. The vision system has the potential to act as such an external source, which can in turn be used to calibrate the arm's movements.

4.2. THE VISION SYSTEM

The necessity to inspect several distant sites with a reasonably high resolution and adoption of a viewing technique (discussed later) led to the selection and adoption of a line scan arm mounted camera.

This configuration is intended to view the slot-sets (or component sets) in the live row of each pallet as well as the inititial and the terminal sites individually.

4.2.1. Formulation of algoritms

The overall problem in the vision system is divided into three distinct subproblems:

a. Recognition of the state of slot-sets/component sets, be it in the initial site, terminal site or anywhere in the live row of each pallet;

b. Location of the pallet;

c. Positional alignment of the arm.

Each of the above demands a separate algorithm which will be addressed in the same order below.

4.2.1.1. Recognition of slot/component sets

Definitions:

A slot-set and a site are synonymous, so in the forthcoming discussion they will be used interchagebly.

x - a slot-set / component set;

a_{ij} - the ith pixel in the jth frame;

L_j - the jth frame;

L - the set of frames representing a slot or component set;

N - the number of pixels per frame;

k - the slot/strip number within a set;

u_k - distance (from datum) of the initial edge of slot k (in pixels); datum in this case is the beginning of the frame;

v_k - width of slot k;

s_k - distance of initial edge of strip k; and

t_k - width of strip k;

T - threshold;

Ω_1 - A void (clean) site. [Qualifies for arm accession only during the loading cycle.]

Ω_2 - A site supporting an intact component set. [Can be accessed only during the unloading cycle.]

Ω_3 - A partially full site, or a site supporting one or more misplaced components.

The scene of interest in this case extends to a slot-set plus a fine peripheral margine around it, as in Fig.4.4.

Data acquisition in such a scene takes place through seven equispaced frames, which span over the whole site as indicated in Fig.4.4. Hence

$$L = \{L_j : j=1,\ldots,7\}$$

where

$$L_j = \{a_{ij} : i=1,\ldots,N\}$$

A fundamental assumption here is that although the components may vary in brightness, but they should always remain brighter than the background. Even if this criterion is violated, a catastrophy is still averted as explained later, when considering reliability.

The image is binary thresholded according to

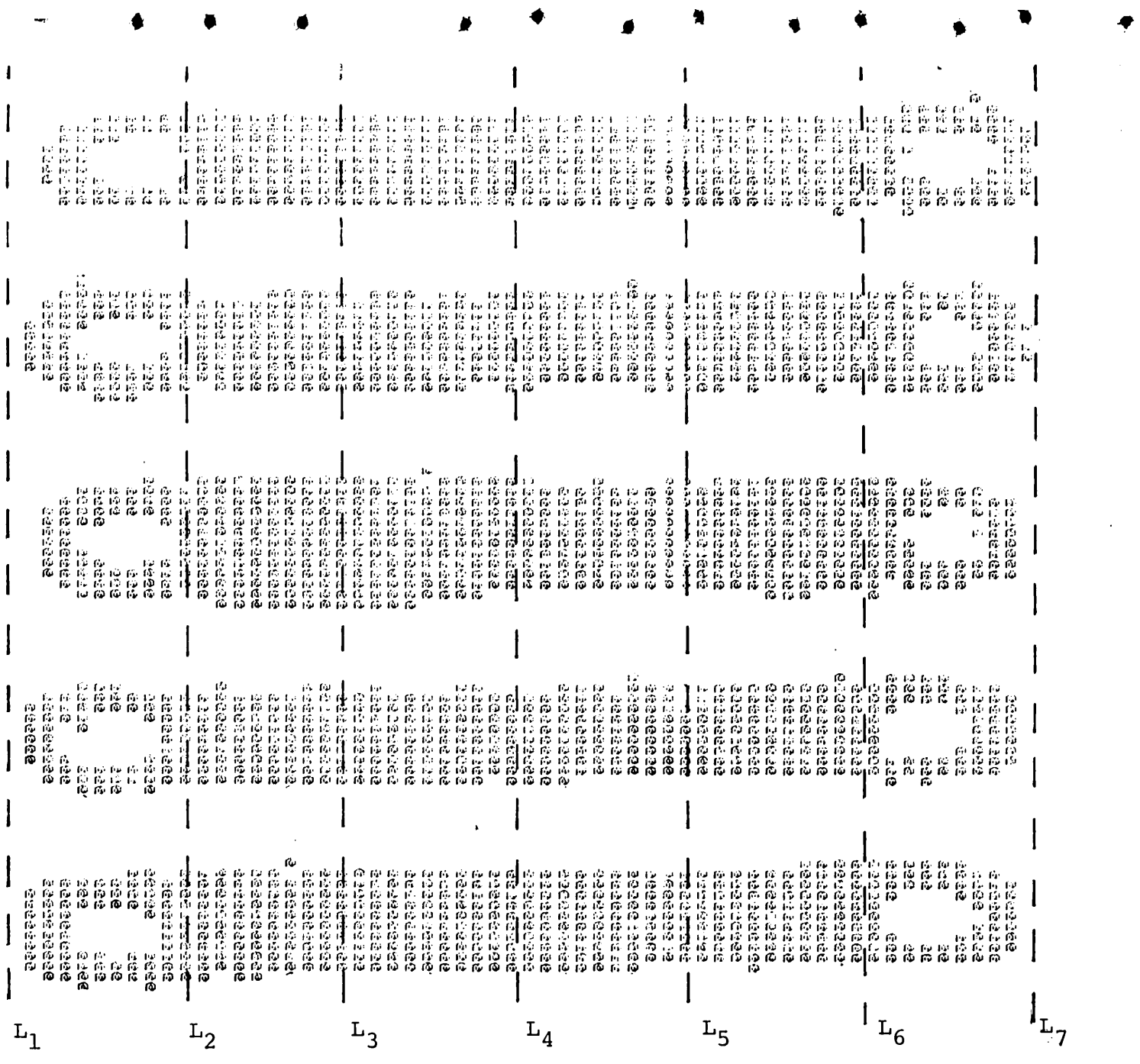


Fig. 4.4. The scene of interest in the site recognition run

$$a_{ij} = \begin{cases} 1 & a \geq T \\ \emptyset & a < T \end{cases}$$

Recognition is based on a sequential classification scheme. The site under consideration either fails to conform with the required state or qualifies for the subsequent stage of classification. The former leads to immediate inhibition of the arm from accessing a site. A site is inferred to be fit for the arm accession if and only if it satisfies all stages of the classification.

The processing associated with L1 is identical to that of L7. These lines, as evident, lie immediately before and after the slot-set. The aim here is to establish whether there exists a protruding bearing cap (i.e. a misplaced component) or not. Such a component set is depicted in Fig.4.5. The solution is based on convolving the following operator

$$H = \begin{matrix} 1 \\ 1 \\ 1 \end{matrix}$$

with the binary thresholded image data due to the acquired frame, and infering that a protruding component exists whenever

$$\left[p_{ij} = 1/3 \sum_{k=1}^{3} h_k a_{i-k} \;\middle|\; i = 2,\ldots,N-1;\ j = 1,7 \right] = 1 \qquad (4.1)$$

or alternatively

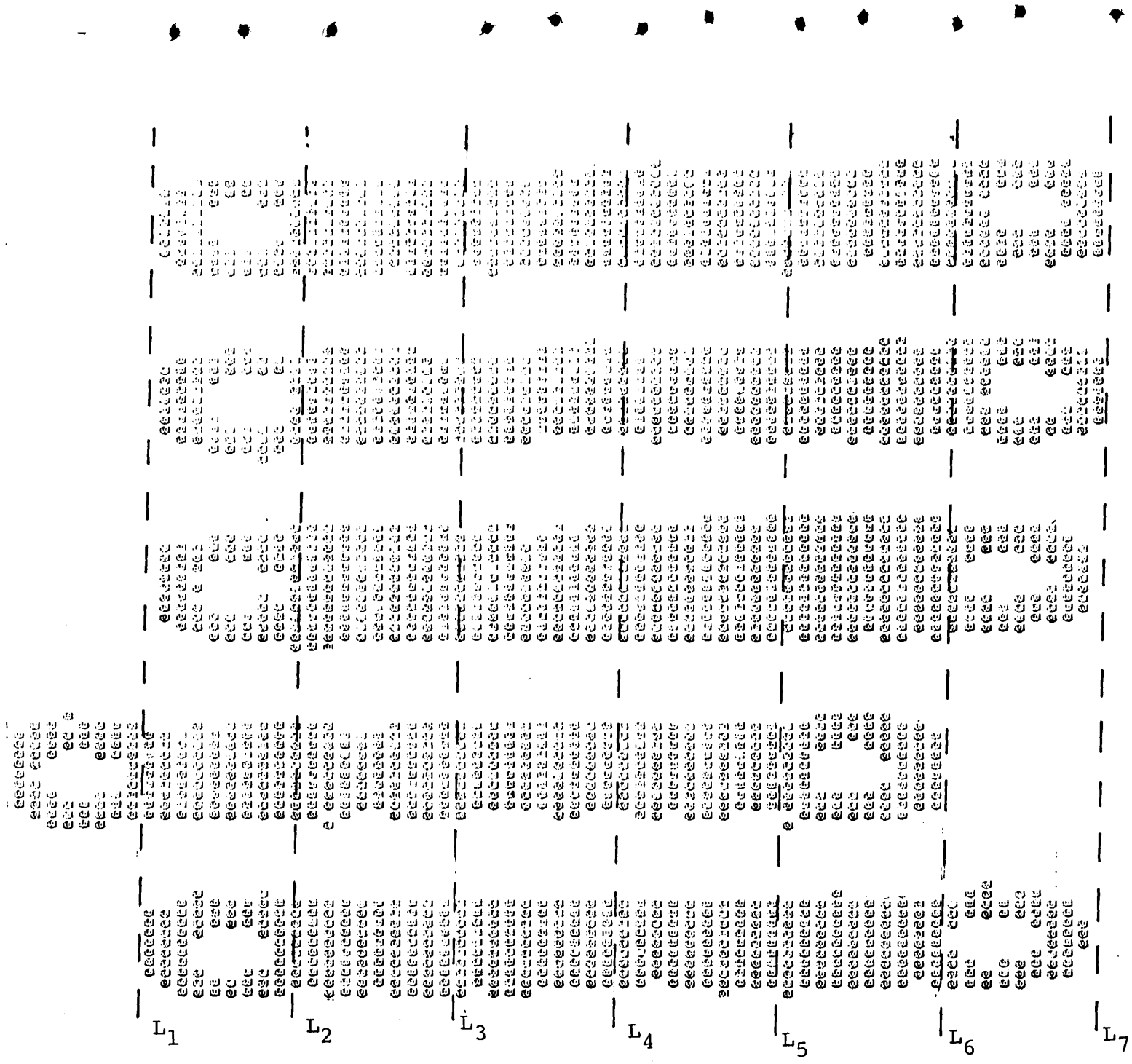


Fig. 4.5. A component set exhibiting a protruding component

$$\sum_{i=I}^{I+2} a_{ij}=3 \qquad I=1,\ldots,N; \quad j=1,7 \tag{4.1}$$

The criterion of three rather than a singular white pixel is merely a heuristic persistence measure aimed at removing false alarms due to noise.

In the remaining frames the aim of the processing is to establish the state of each slot, and verify that the adjacent pallet strips are not obscured by any of the components.

There are three states that can be associated with each slot.

$\omega 1$ - empty

$\omega 2$ - full

$\omega 3$ - partially full

These states can hence constitute the patterns of interest in a classical pattern recognition problem.

The number of white pixels within the boundaries of each slot, i.e.

$$\rho_{kj}=\sum_{i=u_k}^{u_k+v_k} a_{ij}, \qquad k=1,\ldots,5; \quad j=2,\ldots 6 \tag{4.2}$$

is found to provide an effective feature for assignment of

the slots into any of the above catagories. This feature shows tolerable variability and high discrimination power as far as $\omega 1$ and $\omega 2$ is concerned. Let

η_k and Φ_k denote the prototype measurements for classes of empty and full respectively; and

ε_k and μ_k represent the maximum allowable deviation of samples of each class from their respective prototypes.

Classification of slots at various sites proceeds according to

$$\{\phi_{kj}\}_1^1 \subset \begin{cases} \omega_1 & \forall \ \phi_{ij} - \eta_k \leq \varepsilon_{kj} \\ \omega_2 & \forall \ \Phi_k - \phi_{kj} \leq \mu_{kj} \\ \omega_3 & \text{if } \{\phi_{kj}\}_1^1 \notin \omega_1 \cup \omega_2 \\ & \text{i.e. } \eta_k + \varepsilon_k < \phi_{kj} < \Phi_{kj} - \mu_{kj} \end{cases} \qquad (4.5)$$

$$k = 1,\ldots,5$$
$$j = 2,\ldots,6$$

The inner frames (2 to 6) are further processed to reveal clear pallet strips. Interest is confined to

verification of their cleanness and hence detection of any existing misplaced component. Namely, when the following condition prevails the site is dismissed as one accommodating a misplaced component.

$$\sum_{i=s_k}^{s_k+t_k} a_{ij} \gtrless \tau_k, \qquad k = 1,\ldots,6; \quad j = 2,\ldots,5 \qquad (4.4)$$

where τ_k is a dedicated threshold.

Corollary - Class assignment proceeds as follows:

$\chi \in \Omega_1$ iff

$$\sum_{i=I}^{I+2} a_{ij} < 3, \qquad I = 1,\ldots,N-2; \quad j = 1,7$$

$$\{\phi_{kj}\}_1 \subset \omega_1, \qquad k = 1,\ldots,5; \quad j = 2,\ldots,6$$

$$\sum_{i=s_k}^{s_k+t_k} a_{ij} < \tau_k \forall \qquad k = 1,\ldots,6; \quad j = 2,\ldots,6$$

$\chi \in \Omega_2$ iff

$$\sum_{i=I}^{1+2} a_{ij} < 3, \qquad I = 1,\ldots,N-2; \quad j = 1,7$$

$$\{\phi_{kj}\}_1 \subset \omega_2 \qquad k = 1,\ldots,5; \quad j = 2,\ldots,6$$

$$\sum_{i=s_k}^{s_k+t_k} a_{ij} < \tau_k \quad \forall \quad k = 1,\ldots,6; \quad j = 2,\ldots,6$$

$\chi \in \Omega_3$ if any of the following conditions prevails

$$\sum_{i=I}^{I+2} a_{ij} > 2 \quad I = 1,\ldots,N-2; \quad j = 1,7$$

$$\int_l \phi_{kj} \int \omega_1 u \omega_2$$

$$\sum_{i=s_k}^{s_k+t_k} a_{ij} > \tau_k \quad \forall \quad k = 1,\ldots,6; \quad j = 2,\ldots,6$$

Training: The objective here is to determine all unknown parameters required in the site classification algorithm, met above.Training proceeds at three successive stages:

(i) Measurement of

$\{u_k, v_k : k=1,\ldots,5\}$ and $\{s_k, t_k : k=1,\ldots,6\}$

These parameters relate to the boundaries of successive slots in a site, and are extracted from a highly reflective model of a component set,exhibited to the vision system. The component models must fit the corresponding slots as tightly as possible. The above parameters are determined as shown below.

$$u_k = u_{k-1} + v_{k-1} + \upsilon_k \qquad k = 1,\ldots,5$$

$$v_k = \nu_k - v_k \qquad u_0 = v_0 = 0$$

where

υ_k = min (I), with I satisfying

$$\sum_{i=I}^{I+2} a_{ij} = 3 \qquad I = u_{k-1},(u_{k-1}+1),\ldots,N$$

and

ν_k = min (I) with I satisfying

$$\sum_{i=\bar{I}}^{I+2} a_{ij} = 0 \qquad I = u_k,(u_k+1),\ldots,N$$

(ii) Measurement of prototype features

$\{\Phi_k : k=1,\ldots,5\}$

The vision system is exposed to the least reflective components that it is expected to encounter. Then

$$\Phi_k = \sum_{i=u_k}^{u_k+v_k} a_{ij} \qquad k = 1,\ldots,5; \quad 2 \leq j \leq 6$$

(iii) Measurement of prototype features

$\{\eta_k : k=1,\ldots,5\}$

An empty site is viewed, from which

$$\eta_k = \sum_{i=uk}^{u_k+v_k} a_{ij} \qquad k = 1,\ldots,5 \quad 2 \leq j \leq 6$$

The remaining parameters, i.e. μ_k, ε_k, and τ_k are

introduced with a view to width of the dead band met in the pallet location run.

4.2.1.2. Pallet Location The position of each row in a pallet is signified by a corresponding ribbon like mark, referred to as the pallet marker for that row, as in Fig.4.3. The correct positions of the pallets are inferred from coincidence of the pallet marks with a virtual reference marker.

The marks are white and enjoy an improved contrast compared with the components case. The virtual reference marker is also of the same width as the pallet markers.

Every time a pallet is traversed to present a new live row, the reference marker and the pallet marker corresponding to that row must coincide. The task here is to detect and quantify any misalignments and feed it as positional feedback or command input to the servo system responsible for the pallet movement.

In the course of the training, the distances from the beginning to the leading edge of the reference marker, 'F', are computed as [Fig.4.6]

$$F = \min [I], \text{ with } I \text{ satisfying}$$

$$\sum_{i=I}^{I+2} a_i = 3 \qquad I=1,\ldots,N-2$$

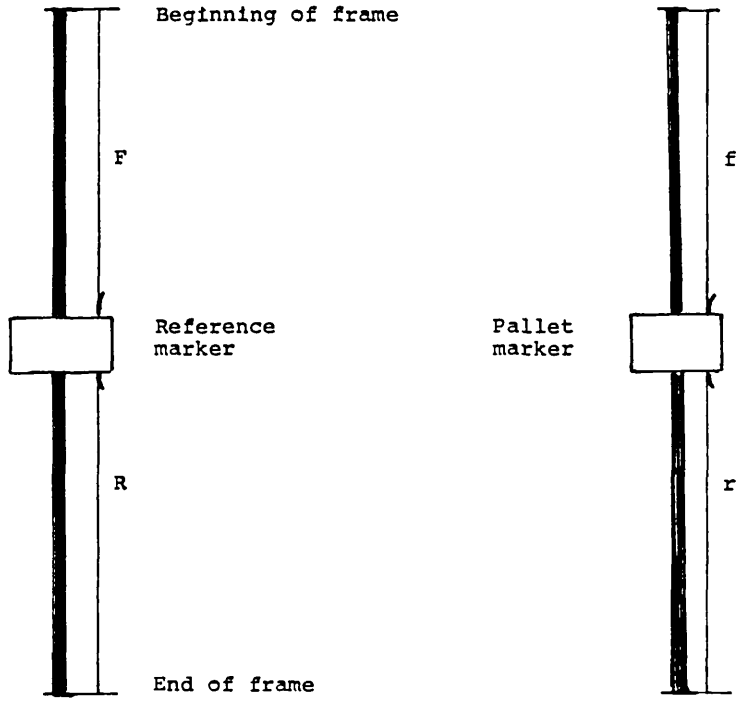


Fig. 4.6. Parameters in the pallet location run

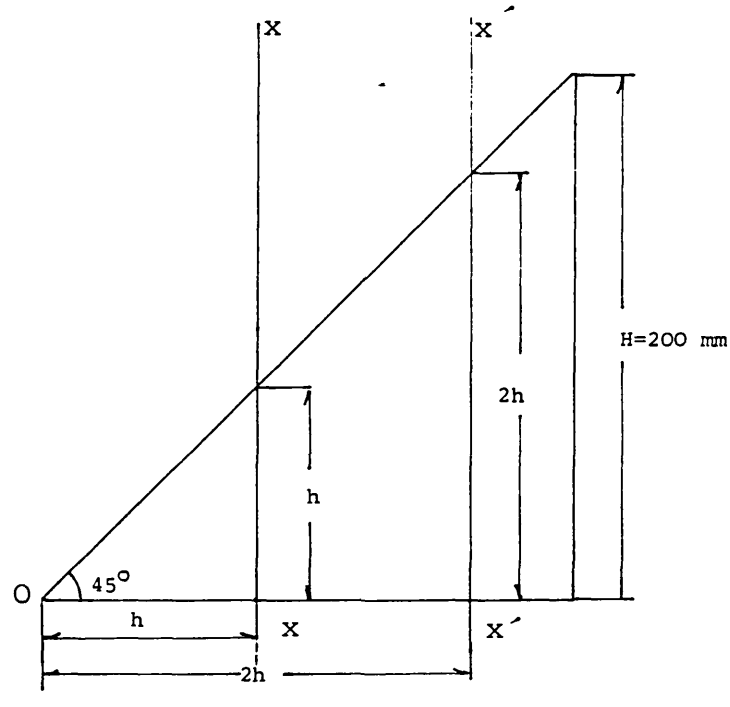


Fig. 4.7. A schematic of the pattern used as bench mark for positional alignment of the arm

Similarly the distance between the trailing edge of the reference marker to the end of the frame, 'R', is computed from

$$R = \min [I] \text{ with } I \text{ satisfying}$$

$$\sum_{i=I}^{I+2} a_{N-i} = 3 \qquad I=1,\ldots,N-2$$

When the run starts, similar distances (f and r) are measured with the pallet marker in sight. Thence the required forward or reverse motion pallet motion (M_f, or M_r) are deduced.

$$M_f = F - f$$

$$M_r = R - r$$

To verify the authenticity of the pallet marker against those spurious marks that may compete with it, the width of the observed marker is matched against that of the reference marker as follows.

$$[N - (F + R)] - [N - (f + r)]$$

$$= (f + r) - (F + R) \leq \tau_L$$

Choice of τ_L is affected by optical distortion and parallax.

4.2.1.3. Positional alignment of the loader arm The

problem is seen as determining the distance of a moving line from a datum point.

If a datum is defined in the horizontal plane at which the camera optics can focus the scene, then the measured distance will relate to the position of the camera and hence the arm. Transmission of the measured distance then enables the arm controller to correlate the actual position with its perceived notion, and effect the necessary corrections.

The devised method employs the triangular pattern of Fig.4.7 as the bench mark. The O vertix of this pattern constitutes the datum.

The steps involved in establishing the above mentioned distance measurement are:

(i) With the prior knowledge of the arm position (based on dead reckoning), the camera is brought to view any arbitrary line XX within the specified triangle. If the scene viewed does not exhibit a solid line (of white pixels), it is inferred that the arm is grossly misplaced. Through activation of an alarm, external help is sought to position the arm over the bench mark (i.e. the triangular pattern).

(ii) If the scene viewed indicates a solid line of length h such that $h < H/2$, the camera is moved further away from the vertix O by a perceived distance h. Now if the new scene exhibits a solid line of length $2h \pm \delta$ (where δ indicates an accepted tolerance), the

authenticity of the bench mark is verified and moreover the arm position is concluded to be a distance of 2h away from vertix O. Otherwise the viewed pattern is dismissed as some other than the bench mark, and external help is sought for coarse alignment of the loader arm. (δ should be a reasonably small tolerance to distinguish between the length 2h on the bench mark and other patterns present along the path which may inadvertantly come into view in the event of coarse misalignment of the arm.) In instances when $h > H/2$, the camera is moved towards O, and the procedure is modified accordingly.

The above steps can be executed during the normal excursions of the arm to and from the initial site.

The positional information can further be used by the arm controller to calibrate the arm movements.

4.2.2. Hardware aspect

Fig.4.8 shows the overall configuration of the vision system, in relation to external equipment with which it must interact.

4.2.2.1. The Image Acquisition Subsystem

The constituent components of this subsystem are presented as follows:

(i) **The sensor:**

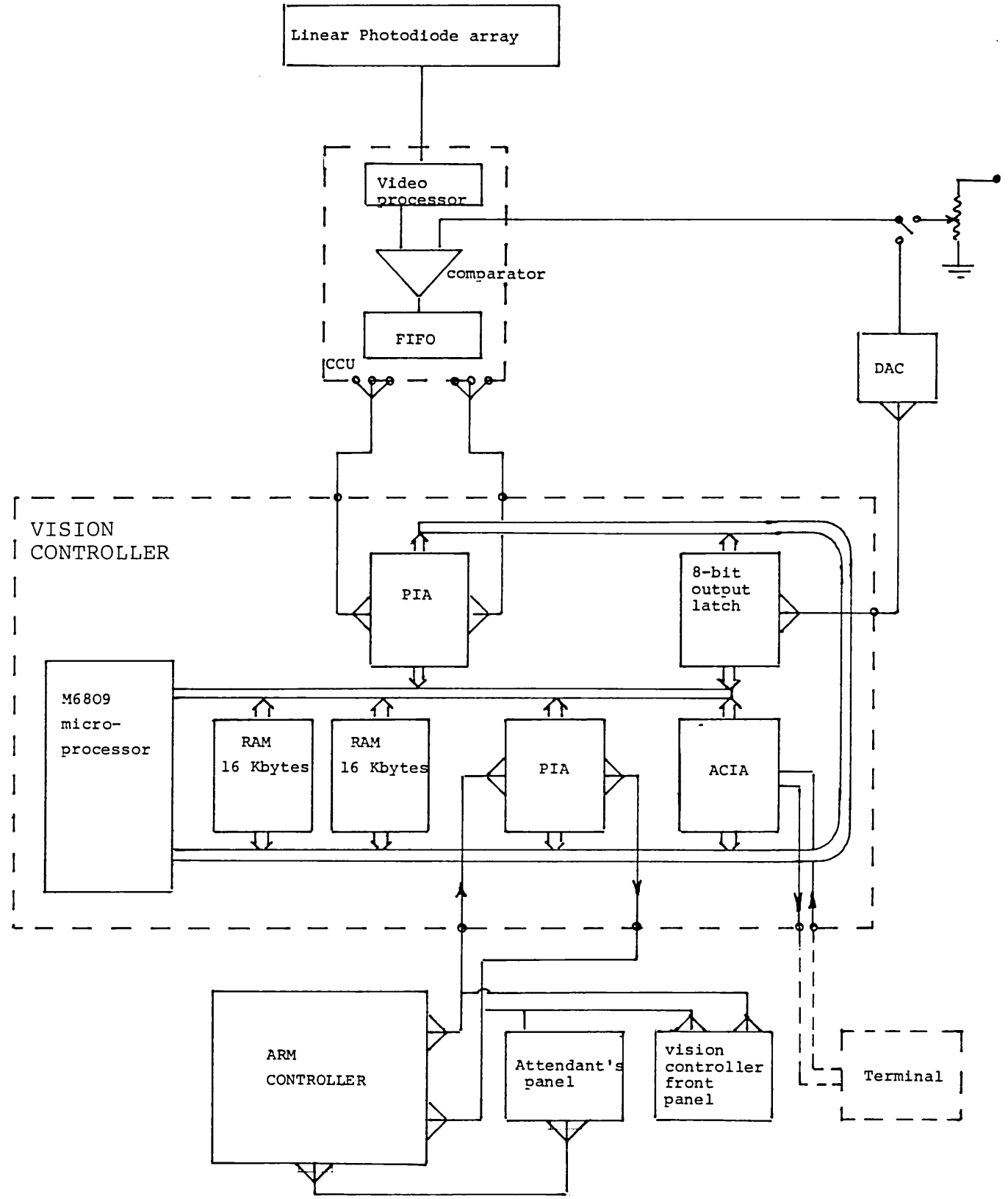


Fig. 4.8. The overall configuration of the vision system.

A linear array of photodiode elements forms the sensor. Selection of a photodiode array as opposed to a CCD array was motivated by the susceptibility of the latter to blooming. The desired resolution (ρ) of Ø.5 mm in a field of view (R) of 25Ø mm demands an array with element population (Ne) of

$$Ne = \frac{R}{\rho} = 500$$

This led to the selection of an "off-the-shelf" array of 512 elements.

An IPL 512M5 based camera (product of Integrated Photometrix Ltd.) was acquired as the sensor.

The size of such a detecter (r) is 12.7 mm. The aperture width (measured perpendicular to the direction of the line scan) is 5 mil (130µm), which yields a transverse resolution of 2.4 mm. The 1 mil aperture option was discarded due sensor demand of higher illumination or longer frame acquisition time.

(ii) **The optics:**

The camera accepts a U-mount (i.e. Universal mount, 42 mm screw fit) lens.

The handling machinary constrains the sensor working distance to a range of 6Ø-7Ø cm. From Eqs (2.1) and (2.2) the required focal length is deduced to be:

$$f = 35 \text{ mm}$$

The above focal length is appropriate for a lens mount which allows the lens to sit a distance equal to that of the calculated focal length away from the photosensitive surface (in this case the photodiode array). But the U-mount lens sits further away from the array, with the additional distance acting as an inherent extension which reduces the working distance as well as the angle of view. As a result, recourse had to be made to a lens of lower focal distance, namely a a 28 mm f/2.5 to secure the desired angle of view while remaining in focus. The depth of field (D_f) associated with this focal length and a relative aperture (α) of 2.5 and the diameter of circle of confusion (c) of Ø.Ø25 mm is obtained from relation (2.3)

$$D_f = 2.4 \text{ cm}$$

Total depth = 4.8 cm

This depth can comfortably cope with the difference in height between the components and the pallet markers. But due to the adopted viewing technique (explained next) the depth of field is not utilised to advantage.

(iii) **The illumination:**

The response of the sensor (inclusive of video processing support circuitry) to the incident light

may be expressed as

$$\int_0^{T_L} \int_{\lambda_1}^{\lambda_2} I(y,\lambda)\, s(y,\lambda)\, d\lambda\, dt \qquad (4.8)$$

where

I is the intensity of incident illumination;

s is the sensitivityn of the sensor;

y is the distance along the linear array;

λ is the wavelength of the light;

λ_1 and λ_2 specify the boundaries of the usable spectrum of the sensor; and

T_L is the line scan time.

Naturally, wherever possible the above variables should be adjusted to secure a tolerable contrast. Bearing in mind the wide spectrum of the silicon based photodiode array, in the first instance it may appear expedient to resort to spectral illumination and paint the background with a favourable colour. However, as the components stand up to 3 cm above the pallet, they are liable to cast a shadow on the background, and hence obscure the supposed desirably painted background. Thus recourse to spectral illumination or use of spectral filter was abondaned in favour of a more robust viewing technique.

The schematics of Fig.4.9 and Fig.4.1Ø depict the viewing technique and the resultant illumination on the scene. Here illumination is provided by a

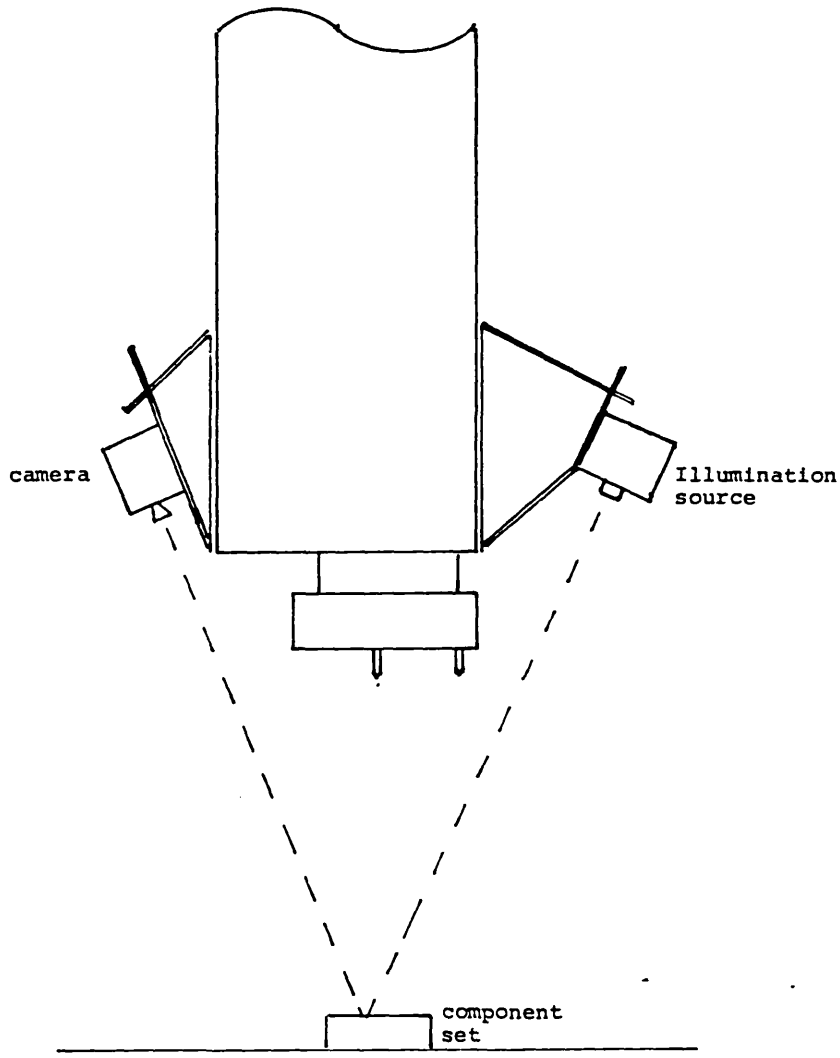


Fig. 4.9. The geometry of viewing technique

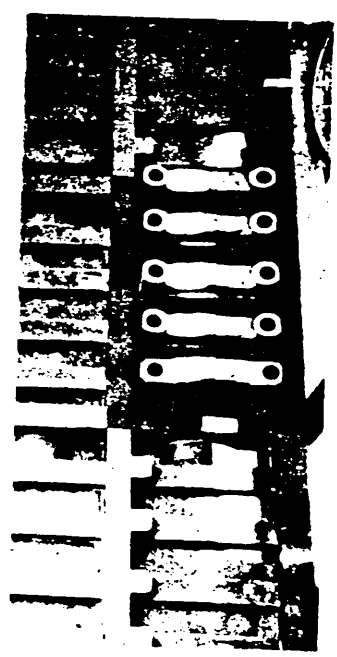


Fig. 4.10. The resultant illumination

focussed source which projects a stripe of light across the scene of interest, as shown in Fig.4.1Ø. The light source and the sensor are so aligned that the sensor is only exposed to luminous energy reflected from the components, irrespective of the pallet shade. Hence the vision system will be immune to variation of the background reflectance which is an important attribute for vision of a pallet which is subjected to continuous impact of components.

Extension of T_L to beyond 1Ø msec, although desirable from exposure point of view, should be avoided in recognition runs unless accompanied by reduction in the speed of the gantry.

A tungsten halogen lamp operating at a nominal voltage of 24 V forms the supposed point source in the focussed illumination source. The low operating voltage of 24 V demands a relatively thicker filament in the lamp which makes it more resistant to abrupt motion of the arm.

The lamp is 25Ø W, and has a nominal life of 2ØØØ Hrs. The lamp's life can be extended substantially by operating it at a slightly lower

voltage.

To avoid generation of excessive heat, the light is switched under program control only during the site inspection periods.

(iv) **Camera-controller interface** The video output from the camera is passed through a comparator where it is thresholded according to a preset level. The binary image data are then passed to a camera control unit (CCU) where the location of the successive transitions are extracted and stacked in a FIFO. The stacked data may be viewed as run-length coded data, but due to the nature of processing in this system, it needs to be reconstructed at the designated image buffer in the vision controller.

The vision controller actually interacts with the camera control unit, rather than the camera itself. The precise nature of the interface appears in Fig.4.13.

4.2.2.2. Micro controller A 68Ø9 based microcontroller forms the vision controller. It is centred around a single board controller which had been designed for an earlier apllication. In its new configuration it has been extended to 3 boards, and supports

A MC 68BØ9 microprocessor

16 K byte of ROM

16 K byte of CMOS RAM

3 ACIA-RS232 Serial Communication ports

2 PIA- MC68B21 parallel communication ports

It was initially intended to transfer the image data via a DMA arrangement to the RAM, as reported in [26]. Subsequent relaxation of the arm's nominal speed to 0.26 m/sec led to the adoption of a standard parallel interface for this purpose.

The vision controller must interact with the following equipments:

- The arm controller, namely a Bosch PC400 programmable controller;

- The Camera Control Unit (CCU);

- The illumination source;

- Its own front panel;

- The attendant's panel.

Communication with the above peripherals is secured through four 8-bit ports, provided by the data lines of the two PIA's, and a serial RS 232C port, as illustrated in Fig.4.8. The lines at ports 1&2 are fitted with opto-isolators.

The tree chart of Fig.4.11 illustrates the various sequence of operations performed by the vision system.

Fig.4.12 depicts the front panel of the vision controller from which all sequences are initiated except those in the hot run, which remain dependent upon signals from the arm controller.

Fig.4.13 shows the detailed interconnections of the vision controller to its peripherals except the terminal which is connected via the RS 232 port only during the cold run.

4.4. PERFORMANCEAND RELIABILITY

The viability of the vision system at the proposal stage was demonstrated via simulation using the test-bed of chapter 2. Here four digitising columns, of a single block, were used to simulate the action of a moving line scan camera. Data acquisition through the first three columns simulated acquisition of L_5 , L_6 , L_7 , while the last column was dedicated to the pallet location run.

In the abscence of a motorised platform, for transverse motion of the pallet, manual movement of a real reference marker (instead of the virtual marker) was used to simulate the pallet motion.

At a subsequent stage, the line scan camera and the

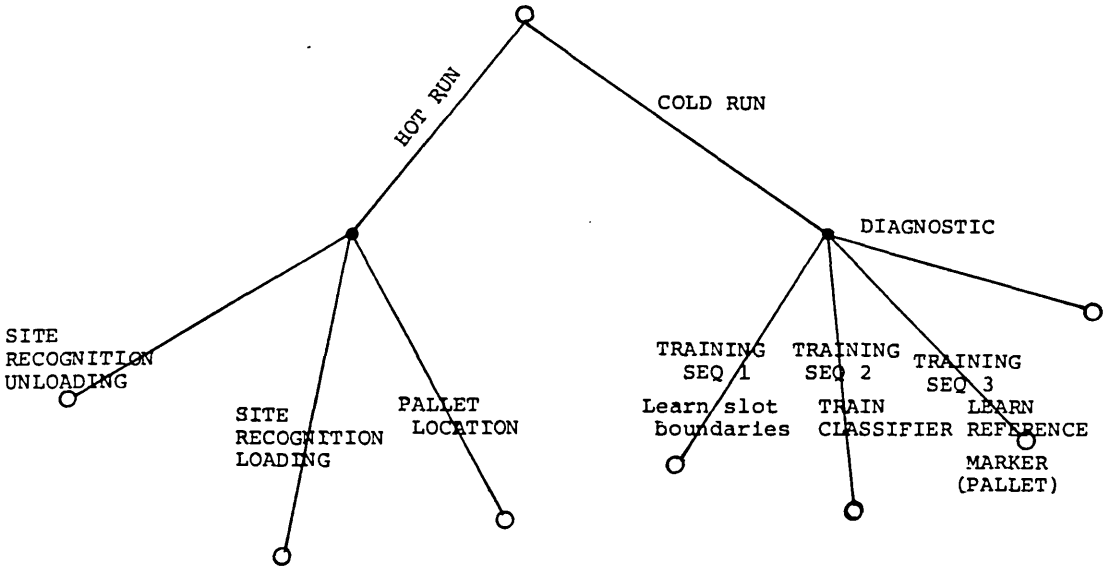


Fig. 4.11. Various modes of operation

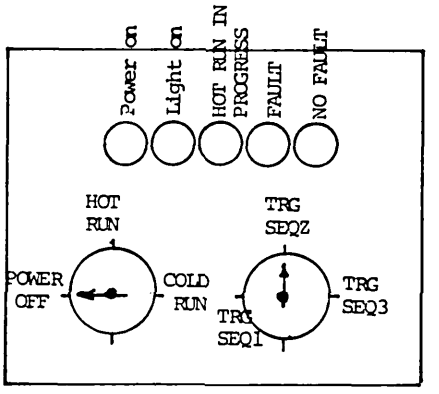
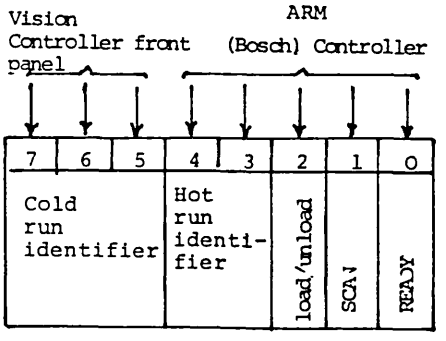
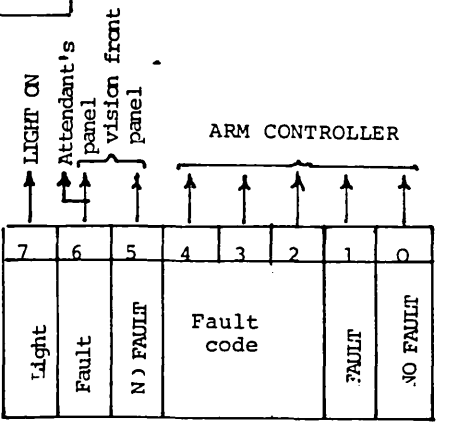


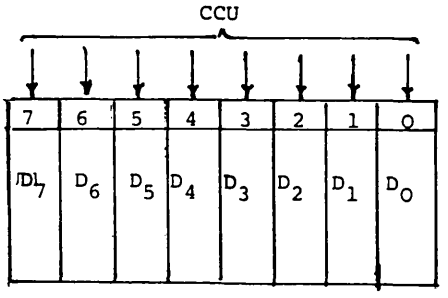
Fig. 4.12. The front panel of the vision controller



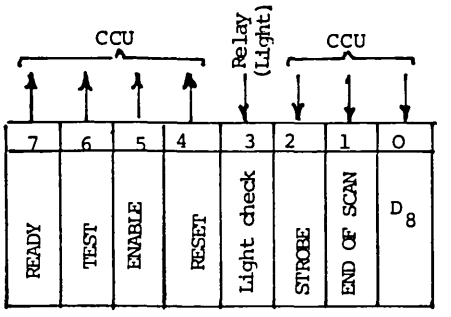
a) Port 1 - Command port



b) Port 2 - Response port



c) Port 3 - Image data port



d) Port 4 - Camera control signals

Fig. 4.13. I/O connections in the vision controller

light source were set-up in the laboratory, as shown in Fig.4.14, and interfaced to the development microcomputer.

The initial software was expanded and modified to match the new sensor, and incorporate the precise nature of interface with the peripherals. [Software implementation at all stages was in assembly.]

Eventually the software was blown into EPROMS and planted in the vision controller, and the performance of the stand-alone vision system was tested at the laboratory and subsequently at the installation site.

The system demostrated that it is capable of performing the tasks efficiently. The trial extended to detection of almost any concievable component misplacement inclusive of overlapping of identical components and appearance in an inverted fashion.

4.4.1. Speed of operation

(i) The processing associated with pallet location takes less than 30 mSec inclusive of the frame acquisition time. (Precise positioning of the pallets extends to such areas as the response of the pallet drives which obviously lies beyond the realm of influence of the vision system.

(ii) The processing associated with scan lines 1 to 7

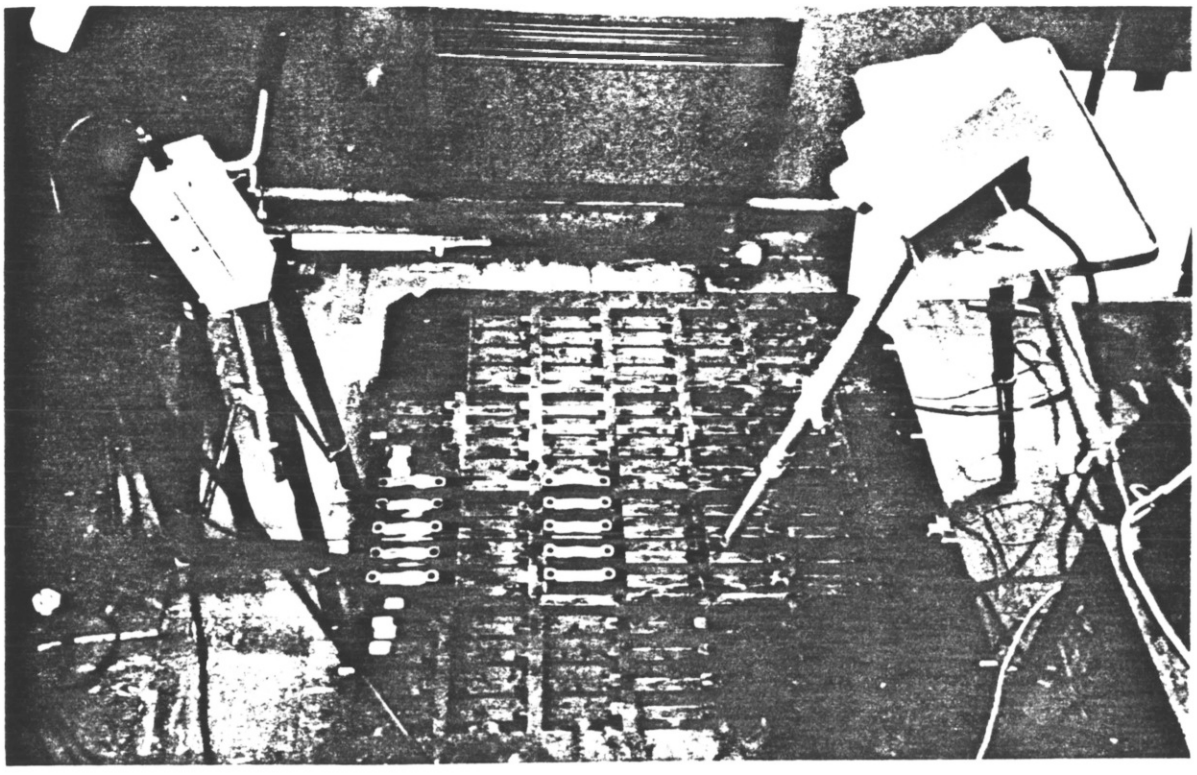


Fig. 4.14. The actual image acquisition subsystem when coupled to the microcomputer and undergoing test

The vision panel housing the vision controller and CCU

The terminal is only connected during the setting up or diagnostic phases

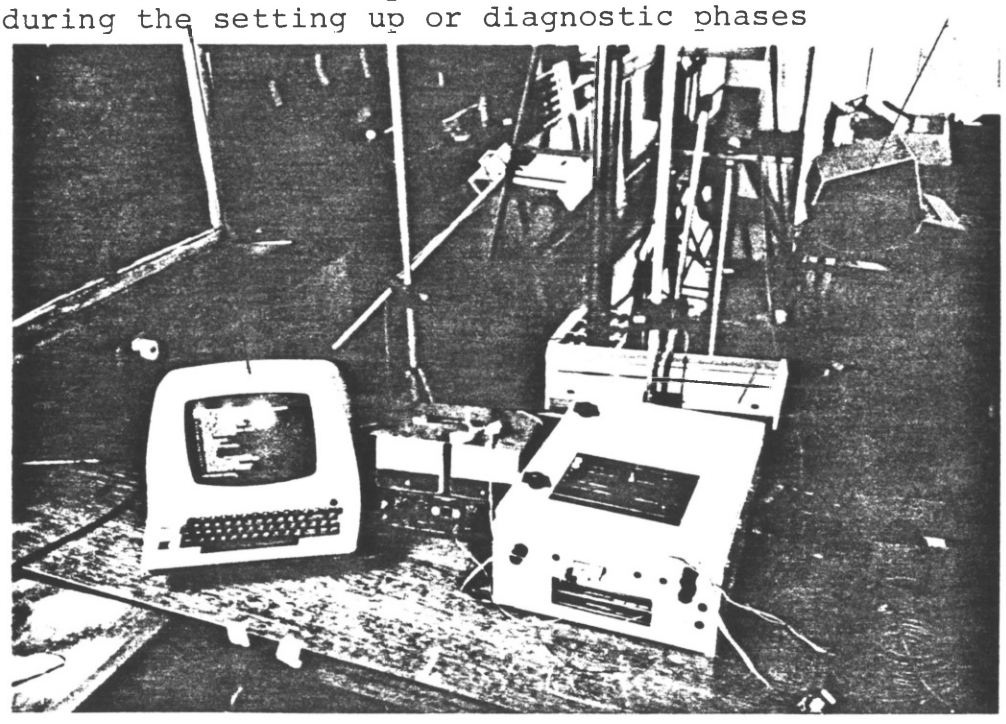


Fig. 4.15. The stand-alone vision system undergoing final tests prior to installation

(met in the slot/component set analysis) lasts less than 4Ø mS . Considering the nominal speed of Ø.26 m/s of the arm, the interval between successive acquisitions extends beyond 8Ø mS . Hence the processing of each frame can be completed prior to acquisition of the subsequent one. Only in the case of the initial and the terminal stations a slight extension of the loader arm's span of excursions will be required whose impact is negligible.

4.4.2 Reliability

The proviso for a satisfactory binary image is, prevailence of a distinct contrast between the components and the background and suitibility of the threshold for segregating the components from the background.

Contrast can be undermined in three ways:

(a) variation in the level of illumination;

(b) development of a malfunction in the sensing circuit.

(c) poor reflectivity of the components.

[By virtue of the viewing technique adopted, reflectance of the background (i.e.the pallet) cannot undermine the contrast.]

To cope with the first two instances, a dedicated

component set is used as the test pattern. This component set should be the least reflective of the expected sets. The test pattern is sited somewhere along the path of travel of the loader arm, see Fig.4.1. The vision system can then regularly attempt to recognise the test pattern, during its excursions to and from the terminal site. Any deviation from the correct classification of the test pattern then initiates a sequence of events during which the arm will be hovering over the test pattern, while the threshold is adjusted under programme control in prespecified increments. If correct classification is achieved by adjustment of the threshold within a predefined range, then the vision system can proceeds functioning with an updated threshold. Otherwise (i.e. in the event of sustained misclassification), the operation is terminated and an alarm to summon the operator is activated. However, inspection at the initial stage serves the purpose of verifying the validity of the preset threshold for the normal cycle and confirms the detectibility of the components in their subsequent processing in all cycles. A too poorly reflective component set will give rise to misclassification of an otherwise intact component set (at the initial site), as it would deviate in some way from normal when the arm controller expects an intact set. This will naturally lead to intervention of the attendant through activation of an alarm.

The reliability of the system is further enhanced by on-line built-in test feature which checks for proper

functioning of the camera, and the light source as well as the consistency of the command signals from arm controller. This is achieved by monitoring the signals indicated in Fig.4.13, and reporting (to the loader controller) any irregularity through a fault code. Even partial or total power supply failure of the vision system is refelected through a dedicated fault code.

The vision system is also supported by an extensive diagnostic software which facilitates fault analysis. When in diagnostic mode (effected through front panel switching), the directory of Fig.4.16 is displayed on the VDU of the terminal. [The terminal is not part of the permenant equipment, it is connected only during diagnostic and training runs as in Fig.4.8.] These commands enable activation

of all signals to the CCU and the gantry controller individually as well as checking the responses, or input from them. The system can also be checked dynamically and the acquired frames can be viwed on the VDU, as in Fig.4.17. The diagnostic is also supported by a monitor program which enables accessing memory locations with the view to read or write or execute the resident program from any stage. This latter feature turns the microcontroller into a modest microcomputer, as new programs can be entered in machine code in the available RAM area, should the need arise, e.g. writing a small test program.

```
*********************************DIAGNOSTIC*******************************

ENTER ANY OF THE FOLLOWING COMMANDS TO INVOKE THE RESPECTIVE OPTION
AT...PERFORM AUTOMATIC SEQUENTIAL TEST OF CAMERA, CCU, AND CONTROLLER
AD...PERFORM SYSTEM DYNAMIC TEST INCLUSIVE OF BOSCH SIGNALS
SC...COUNT NO OF SCAN PULSES PER READY. (incident from Bosch)
TT...TEST- send TEST signal to CCU
NT...Not TEST- cease the TEST signal
RT...RESET- reset FIFO in CCU
NR...Not RESET- cease reset signal to CCU
EA...ENABLE- video into FIFO
DA...DISABLE
DY...Send READY to CCU
NY...Cease READY
ES...DETECT End Of Scan (EOS)
RF...READ FIFO
LT...LIGHT ON
NL...NO LIGHT
RP...READ PORT1 & INPUT BYTE AT PORT4
WP...WRITE TO PORT2 & OUTPUT BYTE AT PORT4
MO...GO TO MONITOR PROGRAMME
VB...VIEW THE FRAME BUFFERS
^D...DISPLAY THE DIAGNOSTIC DIRECTORY
^S...GO TO THE START OF THE ROUTINE


WHAT IS YOUR NEXT COMMAND?
```

Fig 4.16. The diagnostic directory.

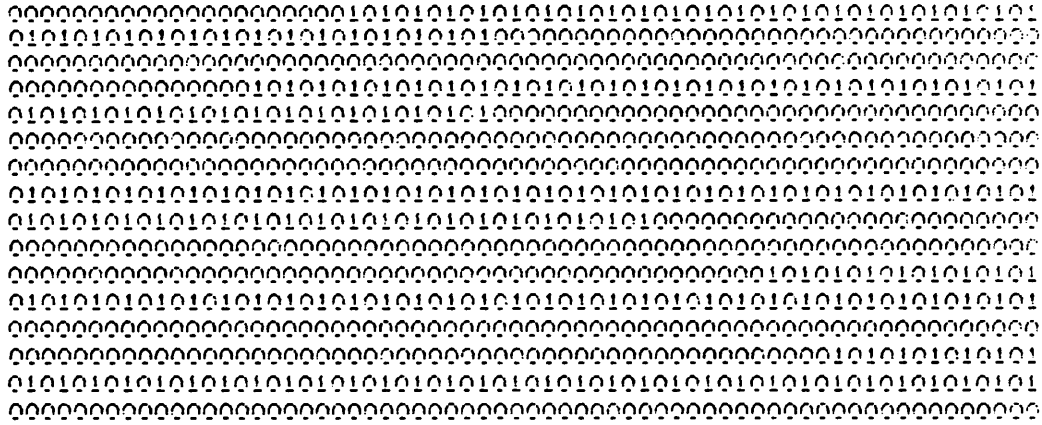


Fig 4.17. A presentation of the buffered frame (showing 5 bearing caps) - obtained via execution of command "AT" of the diagnostic mode.

4.5. REMARKS

(i) The vision system is capable of detecting almost any conceivable misplaced components inclusive of overlapped and inverted components, when inspecting a site. Some of the possible variations of sites supporting misplaced components are shown in Fig.4.17.

(ii) The system is biased towards giving false alarm rather than misclassification leading to undue accession of the arm.

(iii) Pallet location can be performed to an accuracy of better than 1 mm.

(iv) The vision system virtually presents no loss of time to the normal operation cycle of the arm.

(v) The overall hardware cost of the system stands at £4200.

(vi) The installed version does not incorporate the automatic threshold adjustment feature (discussed in 4.4.2), nor the "DAC" shown in Fig.4.8. It was learnt later that the gantry could not provide space for supporting the "test set".

(vii) The camera incorporates two thresholds, one of which eliminates signals below a minimal intensity while

the other thresholds those which lie below a specified percentage of the maximum intensity in the frame. The light source (a modified and ruggedised slide projector with a slit in the slide holder) does not provide an even intensity across the projected strip. In the installation, excessively dark end components against bright centre components were found to be unduely eliminated. To remove this possibility, the integration time of the camera was increased at the expense of a marginal reduction in the arm speed, which did not impact the operation cycle.

(viii) In the course of the trial runs both in loading and unloading cycles, the system proved fully capable of coping with rusty components, and even partially blue painted components sets. The system is going to be commissioned shorty (November 1983). The photograph of Fig.4.18 shows the vision system as installed on the loader arm.

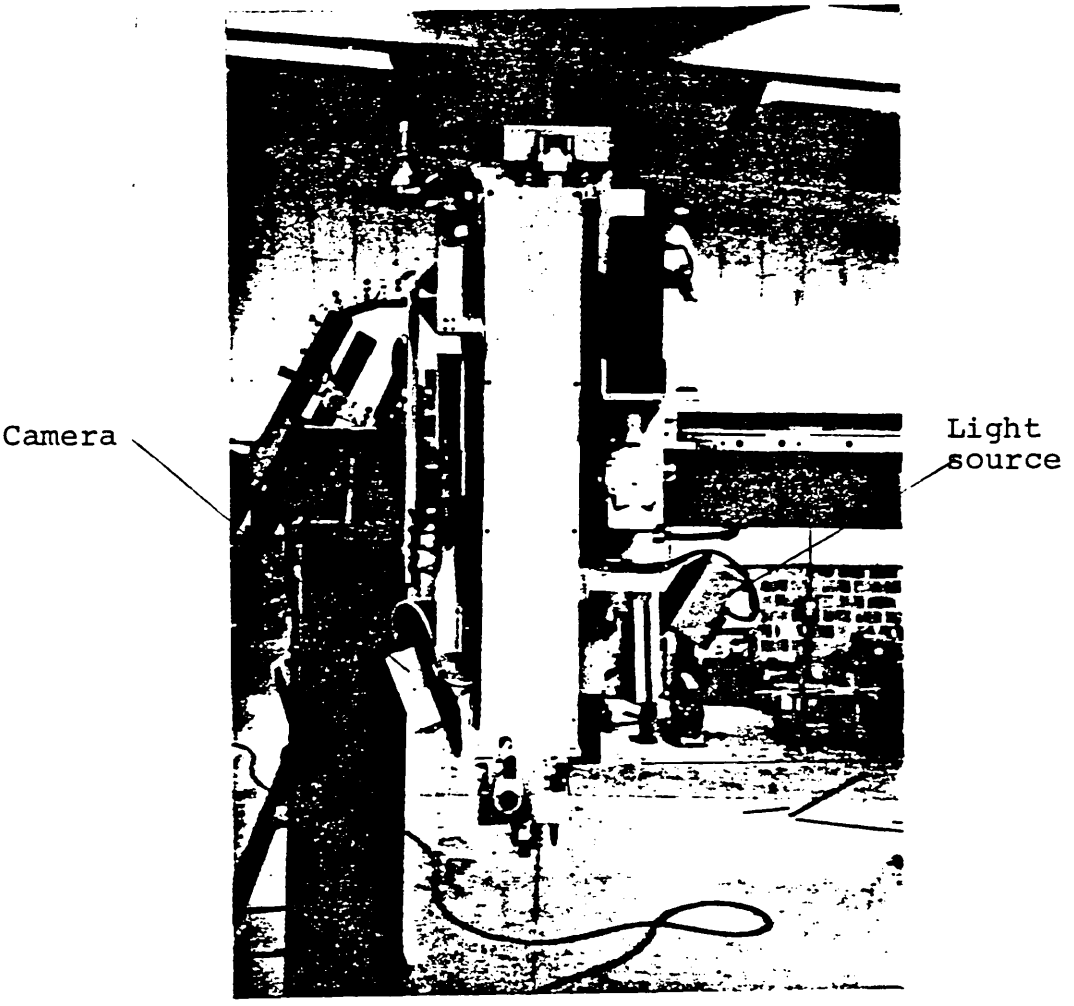


Fig. 4.18 The camera and the light source as installed on loader arm

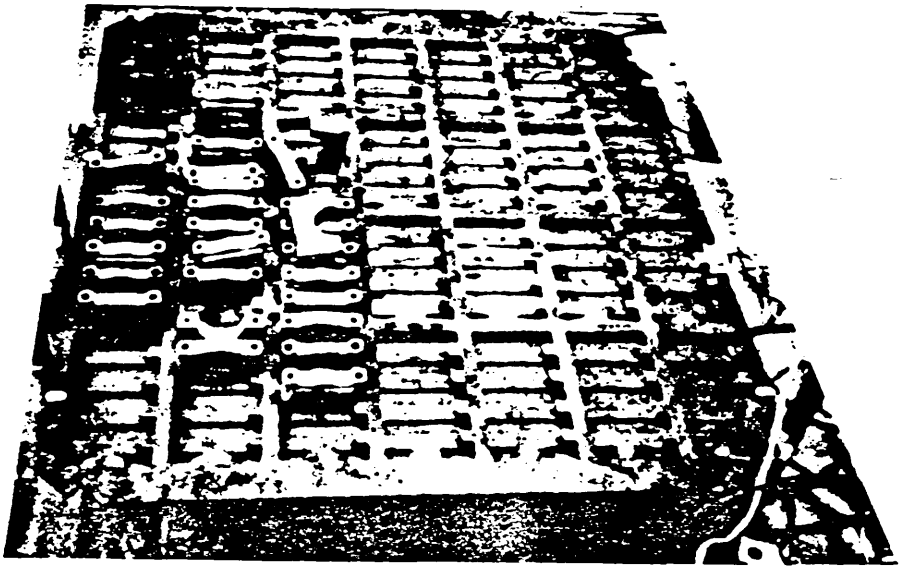


Fig. 4.19 A depiction of possible misaligned components

CHAPTER 5

ORTHOGONAL PROJECTIONS

IN RECOGNITION OF ENGINE COMPONENT SETS

5.1 INTRODUCTION

Representation of binary patterns by their orthogonal projections has for a long time been an attractive proposition to workers in the area of pattern recognition. Much of the recent literature in this area is focussed on characterisation of binary patterns with a view to reconstruct the original patterns from their projections [27-30]. The technique is plagued by its intrinsic ambiguity whose characterisation has also received extensive attention.

The purpose of this chapter is to demonstrate how orthogonal projections can be used to locate, identify and perceive the orientation of the same component set met in chapter 4, were they to appear randomly on a conveyer for subsequent manipulation by an industrial robot. As a prelude to the solution of the ambiguity problem, associated with the patterns of concern, the relevant part of the projection theory is stated. Then a recognition algoritm is presented which entails analytical derivation of a translation, orientation, size invarient feature from the orthogonal projection sets. The work then extends to derivation of an expression for computation of orientation from projections and removal of related ambiguities.

5.2 THEORY

Any account of orthogonal projections would be seriously deficient if reference were not made to Ryser and his

monogram on combinatorial mathematics in 1963 [31] where in considering (Ø,1)-matrices he addresses several combinatorial questions which form the basis of subsequent work in this field.

A binary pattern may be conceived as a (Ø,1)-matrix of size mxn. In such a matrix the orthogonal projection vectors RS and RC are defined as

$$RS = (rs_1, rs_2, \ldots, rs_m)$$

where

$$rs_i = \sum_{j=1}^{n} a_{ij}, \qquad i = 1,.., m$$

and a refers to a matrix element (or a pixel in the binary pattern);

Similarly

$$CS = (cs_1, cs_2, \ldots, cs_n)$$

and

$$cs_j = \sum_{i=1}^{m} a_{ij}, \qquad j = 1,.., n$$

Thus any binary pattern can be decomposed into a set of orthogonal projection vectors , or projection set PS, such

that

PS = PS (RS, CS)

If all patterns could be discriminantly represented by their projection sets, then the data compression aspect of the technique would have rendered it extremely attractive. Unfortunately orthogonal projection sets do not uniquely characterise all binary patterns, and give rise to ambiguities in many cases. The ambiguity problem is well studied, and to elucidate the potential and limitations of the technique its salient points are discussed here.

A projection set PS(RS,CS) by definition is unique, nonunique or inconsistent in accordance with the ensuing criteria.

a. It is **unique** if and only if it determines one binary pattern.

b. It is **nonunique** if and only if it determines more than one binary pattern.

c. It is **inconsistent** if and only if it does not correspond to a binary pattern.

Alternatively a binary pattern, A, is said to be ambiguous if there exists another binary pattern, B, with identical projection set, otherwise it is unambiguous.

The following matrices provide examples of ambiguous

patterns.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

As evident they both give rise to an identical projection set PS=PS(RS,CS), where

$$RS = (2,3,3,1)$$
$$CS = (1,4,3,1)$$

To characterise the ambiguity, consider the 2x2 matrices

$$S_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad S_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Ryser defines an interchange as a transformation of the elements of S into type S or vice versa. It is apparent that such a transformation does not affect the corresponding projection set. S_1 and S_2 are known as interchange or switching components.

A switching component may in general be defined as a 2x2

matrix

$$\begin{bmatrix} a_{ij} & a_{i,(j+q)} \\ a_{(i+p),j} & a_{(i+p),(j+q)} \end{bmatrix}$$

with $p, q, \neq 0$ and

$$a_{ij} = a_{(i+p)'(j+q)} = 1 \text{ or } 0$$

$$a_{(i+p),j} = a_{i,(j+q)} = \bar{a}_{ij}$$

Theorem 5.2.1: A binary pattern is ambiguous if and only if it contains a switching component.(This theorem is due to Ryser)

The earlier examples of ambiguous patterns, A and B, contained swithching components in their undermentioned submatrices.

$$S_1 = \begin{bmatrix} a_{21} & a_{24} \\ a_{31} & a_{34} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and

$$S_2 = \begin{bmatrix} b_{21} & b_{24} \\ b_{34} & b_{34} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Detection of a switching component by the brute force approach in an mxn matrix can, in the worst case, require an exhaustive inspection of C_2^m x C_2^n 2X2 matrices, (e.g. $66x10^6$ in a 128 x128 resolution).

Algorithms have been developed to cope with ambiguity detection problem in more efficient ways. One such algorithm pursues a deletion sequence in which rows and columns accommodating only 1's and only Ø's are deleted resulting in smaller matrices. Now if the deletion process progresses until the ultimate elimination of all elements, the patten is deduced to be unambiguous.

The ensuing binary patterns are examples of ambiguous and unambiguous patterns being discovered through the sequential deletion technique.

$$A = \left[\begin{array}{c|ccc} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \longrightarrow \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ \cancel{0} & \cancel{0} & \cancel{0} \end{array}\right]$$

$$\longrightarrow \left[\begin{array}{c|cc} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cc} \cancel{0} & \cancel{0} \\ 0 & 1 \\ 1 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}\right]$$

Hence A is ambiguous.

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Thus B is unambiguous

Characterisation of ambiguity in binary patterns was pursued by Wang [32] and others who based on earlier ideas of Ryser introduced the canonical concept, which led to even a more efficient algorithm. Nonetheless the former technique is adequate for solutions considered here in, as on-line detection of ambiguity is not a requisite of this particular application.

5.3 STATEMENT OF PROBLEM

Consider the handling process of Chapter4, with the following qualifications:

(i) The same componet set as that met in chapter 4 forms the pattern of interest. The components, this time, upon emerging from the production line are not pushed

inside a slot-set but are allowed to sit next to one another forming a compact set. The components could, perhaps, be pushed into a holding tray that is only fit to simply hold them together. They are further delivered to a conveyer mechanism rather than a stationary site, as in the previous case. A typical set appears in Fig. 5.1.

(ii) The conveyer next carries the component set past an inspection point, where if found to be intact its location and orientation is determined and transmitted to an industrial robot which picks and delivers the set to the next process. A component set identified as "not intact" is left on the conveyer and driven to a point where it is attended by the operator for a subsequent recycling phase.

The problem can alternatively be expressed in terms of capabilities of the vision system as follows.

(i) Identifing the successive component sets as presented for inspection somewhere along the conveyor.

(ii) Location of those component set that have been identified as an intact component set.

(iii) Computing the orientation of the intact component sets.

(iv) Transmission of location and orientation information

Fig. 5.1. A typical component set

to the robot controller.

5.4 ALGORITHMS FOR SOLUTION OF THE PROBLEM

The overall problem is divided into three distinct subproblems, viz

a. Recognition of component sets

b. Location of component sets

c. Perception of orientation of intact component sets

These will be addressed in the same order herein.

5.4.1 Recognition of component sets

A component set may assume any of the following classes of configurations:

$\omega 1$: An "intact" component set, i.e. comprising 5 fully aligned components.

$\omega 2$: A component set consisting of a lesser number of components, e.g. 4, 3, 2, 1, or being void of component.

$\omega 3$: A component set being composed of 5 components but with one or more overturned components.

$\omega 4$: A component set comprising 5 components but with one or more of the components being misaligned.

It will be shown later that the proposed recognition

algorithm can readily be expanded to indicate the number of missing components in ω_2 , while indicating whether they are aligned and do form a compact set or not.

A binary thresholded image of an intact component set when sitting on a brighter background will be that of series of contiguous rectangles producing a larger rectangle. The diagram of Fig.5.2 depicts an schematic of such a component set along with its projection set.

Before getting any further it is deemed proper to consider the ambiguity aspect of the problem.

Let the following notations prevail:

A = the set of pixels representing the components within a scene.

$PS_{\omega k}$ = the orthogonal projection set associated with the class ω_k

The sum of the pixels due to a component set is given by:

$$\sigma_k = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}, \quad A \subset \omega_k; \quad K=1,2,3,4$$

The following relation prevails between , the row sums $\{rs_i\}$, and the column sums $\{cs_j\}$.

$$\sigma_{\omega k} = \sum_{i=1}^{m} rs_i(k)$$

$$= \sum_{j=1}^{n} cs_j(k)$$

As

$$\sigma_{\omega 2} < \sigma_{\omega 1} < \sigma_{\omega 3}$$

therefore

$$PS_{\omega 2} \neq PS_{\omega 1}$$

and

$$PS_{\omega 3} \neq PS_{\omega 1}$$

Corollary: patterns of class ω_1, ω_2 and ω_3 cannot give rise to identical projection sets and hence will not be ambiguous with respect to each other.

In a binary thresholded image an intact component set when appearing against a brighter background gives rise to a convex symmetric shape which is only unambiguous when its orientation (angle θ in Fig.5.2, i.e. the angle that its major axis makes with the horizontal) equals any multiple of $\pi/2$, otherwise it is ambiguous.

Ambiguous patterns in this problem fall into two catagories:

a. those which can be realised by a set of five upright components;

b. those that satisfy the ambiguity criterion mathematically but cannot possibly be imposed on the component sets, as they demand physical distortion of the components.

The former class are those that still assume an "intact" form but appear with different orientations. They will be addressed later along with the orientation problem. The latter by virtue of their definition cannot occur and hence is dismmissed.

5.4.1.1 Feature Selection: To perform classification a translation-orientation independent feature is required. From the diagram of Fig.5.4 the following equations can readily be deduced:

$$\ell \cos\theta + w \sin\theta = P \qquad (5.1)$$

$$\ell \sin\theta + w \cos\theta = \Gamma \qquad (5.2)$$

From equations (5.1) and (5.2):

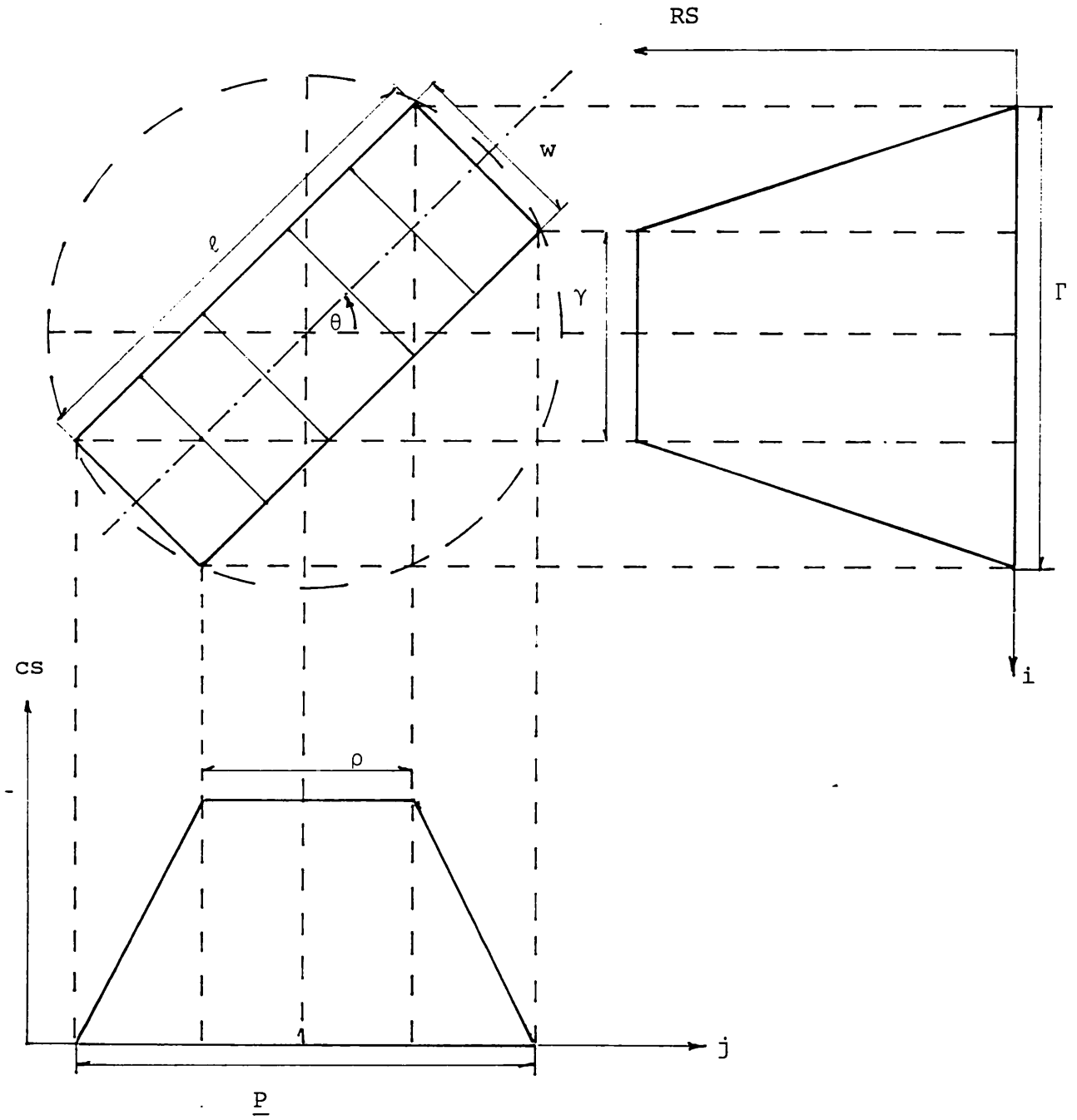


Fig 5.2 The geometry of the projection set for a compact component set corresponding to Equation (5.7)

$$(\ell+w)\ (\cos\ \theta + \text{Sin}\ \theta) = P + \Gamma \qquad (5.3)$$

Also from Fig.5.4

$$\ell\ \cos\ \theta - \omega\ \sin\ \theta = \rho \qquad (5.4)$$

$$\ell\ \sin\ \theta - \omega\ \cos\ \theta = \gamma \qquad (5.5)$$

Hence

$$(\ell-\omega)\ \ (\cos\ \theta + \sin\ \theta) = \ell + \gamma \qquad (5.6)$$

Division of equation (5.3) by (5.4) yields:

$$\psi = \frac{\ell + w}{\ell - w} = \frac{P + \Gamma}{\rho + \gamma} \qquad (5.7)$$

Equations (5.4), (5.5) and (5.7) are only valid when an intact component set assumes orientations that give rise to

$$P \geq \ell\ \cos\ \alpha + w\ \sin\ \alpha \qquad (5.8)$$

$$\Gamma \geq \ell\ \cos\ \alpha + w\ \sin\ \alpha \qquad (5.9)$$

where $\alpha = \tan^{-1} \frac{w}{\ell}$

When either of the above criteria, i.e. (5.8) or (5.9) is violated, then either Eq.(5.4) or Eq.(5.5) must be replaced by Eq.(5.10) or (5.11) respectively. [Fig.5.3 exemplifies such a situation.]

$$w \sin \theta - \ell \cos \theta = \rho \tag{5.10}$$

$$w \cos \theta - \ell \sin \theta = \gamma \tag{5.11}$$

In either event, feature element ' ψ ' is obtained through a modified expression as

$$\psi = \frac{\ell + w}{\ell - w} = \frac{p + \Gamma}{|\rho - \gamma|} \tag{5.12}$$

As evident from Equations (5.7) and (5.12), ' ψ ' defines a translation-orientation independent feature. Table 5.1 depicts the discrimination power of the above feature for various compact sets.

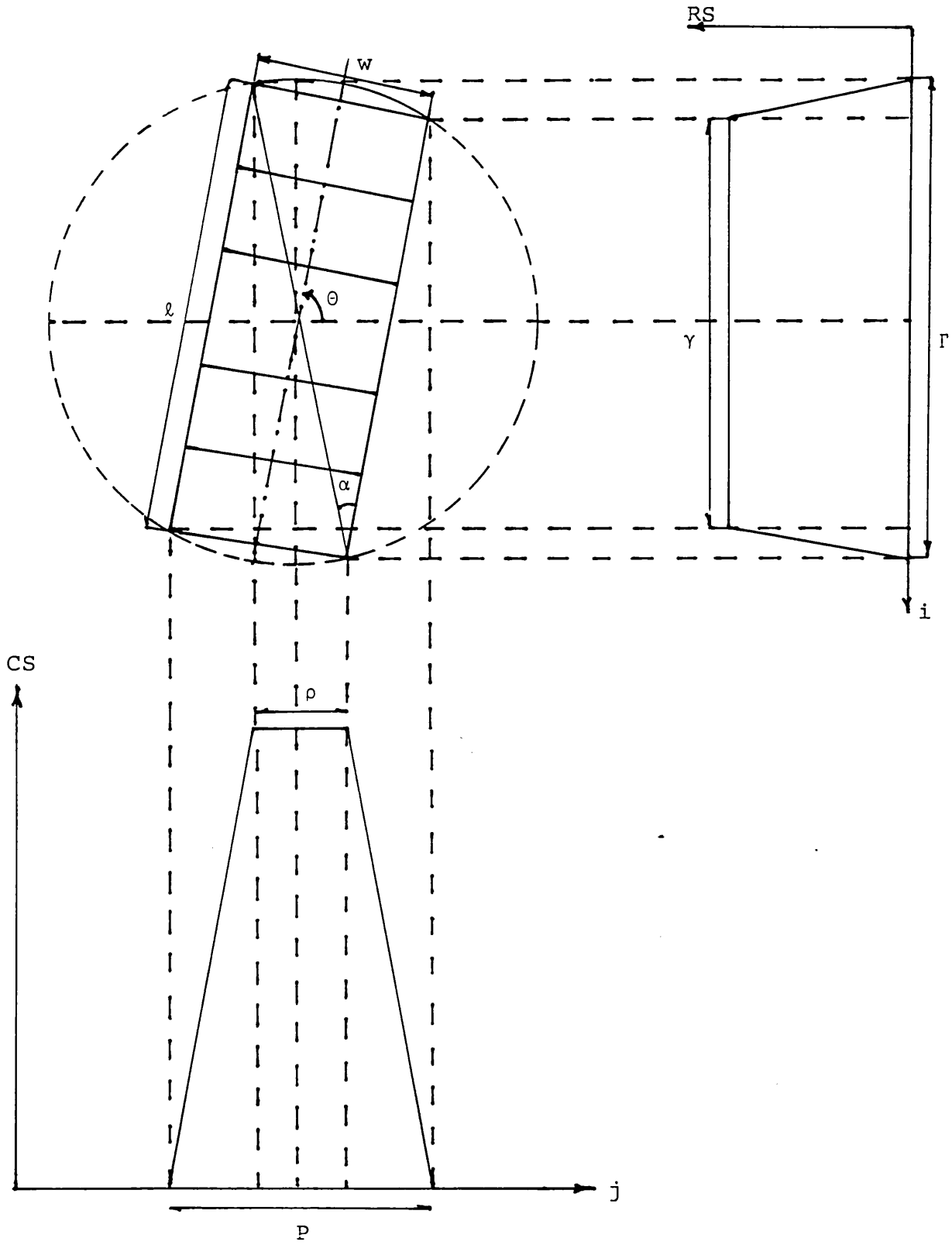


Fig. 5.3 The geometry of projection set for a compact component set corresponding to Eauation (5.12)

No. of components	ψ
1	1.5 - 1.7
2	2.2 - 2.8
3	3.0 - 5.6
4	9.8 - 16.8
5	27.1

Table 5.1 Values of ψ for different classes of compact sets

Identification in this problem rests on a sequential classification scheme with the feature vector

$$\phi = \begin{pmatrix} \sigma \\ \psi \end{pmatrix}$$

5.4.1.2 Preprocessing and feature extraction The preprocessing leading to extraction of the first feature element, σ, which also locates the the component set within the scene is delineated below.

Notations:

a. suffixes f and l, and f' and l' denote the first and the last points in RS and CS projection vectors due to a pattern (i.e. a component set).

b. I_c and J_c denote the coordinates of the median centre of the component set.

c. The span of a projection vector is defined as the

smaller of the two in the pair of

$(I_c - f)$ and $(l - I_c)$

or

$(J_c - f')$ and $(l' - J_c)$

The following steps constitute the preprocessing sequence:

Step1. Remove isolated row or column sum elements in either of the projection vectors.

Step2. Compute as

$$\sigma = \frac{1}{2}\left(\sum_{j=f'}^{l'} cs_j + \sum_{i=f}^{l} rs_i\right)$$

Step3. Compute the median centre of the binary pattern from

$$\sum_{i=f}^{I_c} rs_i = \frac{1}{2}\sigma$$

$$\sum_{j=f}^{J_c} cs_j = \frac{1}{2}\sigma$$

Step4. Determine the projection span (according to definition 'c' above). Update f, l, f', l' to conform to new span.

Step5. Go to step 2 until prespecified number of iterations have been performed, (two iterations in the case of the implemented routines), otherwise

exit.

The output of the above will be a refined σ as well as the median centre of the total set.

The second feature element, ψ, is computed from the refined projection vectors, resulted in from the preprocessing associated with measurement of σ.

5.4.1.3 Classifier training

Let

$\{ \Sigma_i^{(k)}, \Psi_i^{(k)}: k=1,\ldots,M\}$ represent the two prototype feature element for the M classes of concern; and

$\{ \Sigma_i^{(k)}, \Psi_i^{(k)}: i=1,\ldots,n\}$ denote the values of feature elements for each class of the training set.

Prototype feature element computation is based on the following:

$$\Sigma^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \Sigma_i^{(k)}, \qquad K = 1, .., M$$

or equally

$$\Sigma^{(k)} = \text{Median } \{\Sigma_i^{(k)} : \; i = 1, .., n \}$$

and similarly

$$\psi^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \psi_i^{(k)} , \quad k = 1, .., M$$

or

$$\psi^{(k)} = \text{Median } \{\psi_i^{(k)} : i = 1,.., n\}$$

5.4.1.4 Classification There are two stages to the classification. The initial stage is governed by the following decision rule:

$$x \notin \omega_i \quad \ldots \quad \text{if} \quad |\Sigma^{(i)} - \sigma| \neq \min_{k=1,..,M} |\Sigma^{(k)} - \sigma|$$

with x representing the incident component set.

At the second stage

$$x \in \omega_k \quad \text{if} \quad |\psi^{(k)} - \psi| \leq \xi^{(k)}, \quad K=1,..,M$$

where $\xi^{(k)}$ indicates the permissible deviation of the respective feature element for class ω_k.

It should be borne in mind that the ultimate objective is to verify the compactness of a component set which is known, through initial stage of classification, to comprise 5 upright components.

5.4.2 PERCEPTION OF ORIENTATION

From Eq.(5.3)

$$\sin 2\theta = \left(\frac{P + \Gamma}{\ell + w}\right)^2 - 1$$

$$\theta = \begin{cases} n\pi + \frac{1}{2}\sin^{-1}\left[\left(\frac{P+\Gamma}{\ell+\omega}\right)^2 - 1\right] \\ \\ (n+\frac{1}{2})\pi - \frac{1}{2}\sin^{-1}\left[\left(\frac{P+\Gamma}{\ell+\omega}\right)^2 - 1\right] \end{cases}$$

with $n = \emptyset$

$$\theta = \begin{cases} \frac{1}{2}\sin^{-1}\left[\left(\frac{P+\Gamma}{\ell+\omega}\right)^2 - 1\right] \\ \\ \frac{\pi}{2} - \frac{1}{2}\sin^{-1}\left[\left(\frac{P+\Gamma}{\ell+\omega}\right)^2 - 1\right] \end{cases} \qquad (5.7)$$

Eq.5.7 expresses an existent ambiguity, pertaining to the adopted method of computation of the orientation. The ambiguity in this instance is removed by comparison of P, and Γ.

$\theta < \pi/4 \qquad \Gamma < P$

$\theta = \pi/4 \qquad \Gamma = P$

$\theta > \pi/4 \qquad \Gamma > \Gamma$

Fig.5.5 provides evidence of another ambiguity. As evident, the projection vectors are insensitive to compact component sets assuming orientation angles of θ and $(\pi-\theta)$. Hence removal of this ambiguity demands additional information, not contained in the projection set.

Here the additional information is sought in the relation of the co-ordinates of the first component pixel (I_p, J_p) detected in the course of the raster search, with those of

the median centre, (I_c, J_c).

The significance of (I_p, J_p) and (I_c, J_c) relation becomes evident by illustrations of Fig.5.5.

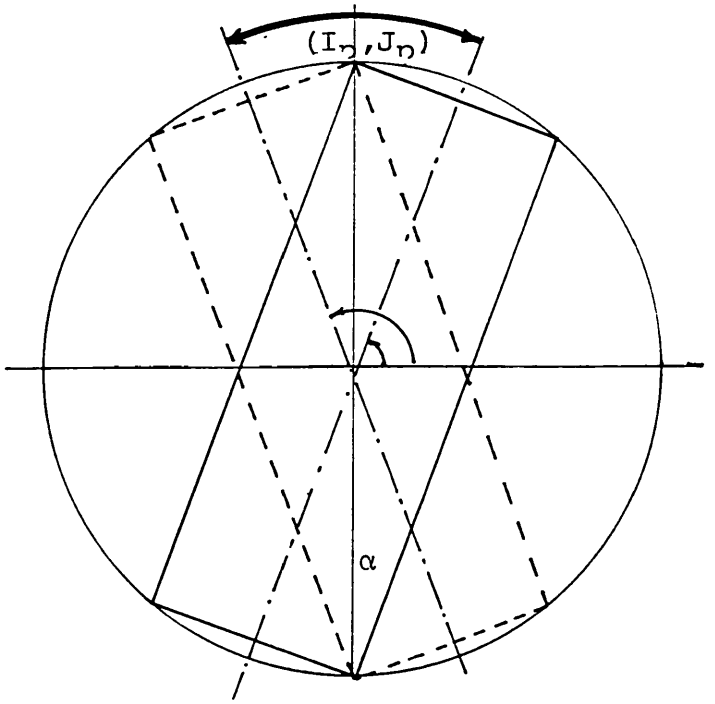
If the unambiguous orientation angle is denoted by θ', then application of the following rule removes the ambiguity.

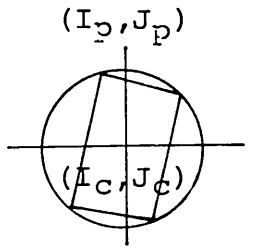
$$\begin{aligned}
&\theta' = \theta \text{ if } J_p > J_c \text{ and } P^{(k)} > T^{(k)} \\
&\theta' = \pi - \theta \text{ if } J_p < J_c \text{ and } P^{(k)} > T^{(k)} \\
&\theta' = \theta \text{ if } J_p < J_c \text{ and } P^{(k)} < T^{(k)} \\
&\theta' = \pi - \theta \text{ if } J_p > J_c \text{ and } P^{(k)} < T^{(k)}
\end{aligned} \qquad (3.8)$$

where T denotes a threshold ratio corresponding to orientation of the component set as shown in Fig.5.5(a), and k=1,...,5 indicates the number of components in the set.

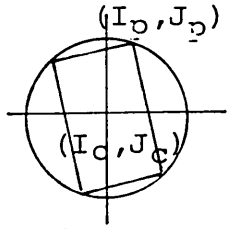
The interest is predominantly focussed on the case when k=5. The preceeding rule indicates that the algorithm can be stretched to compute orientation angles for compact sets that consist of a lesser number of components.

Obviously, the algorithm in its current form would be noise prone, and to decrease such vulernability the following heuristic criterion is enforced.



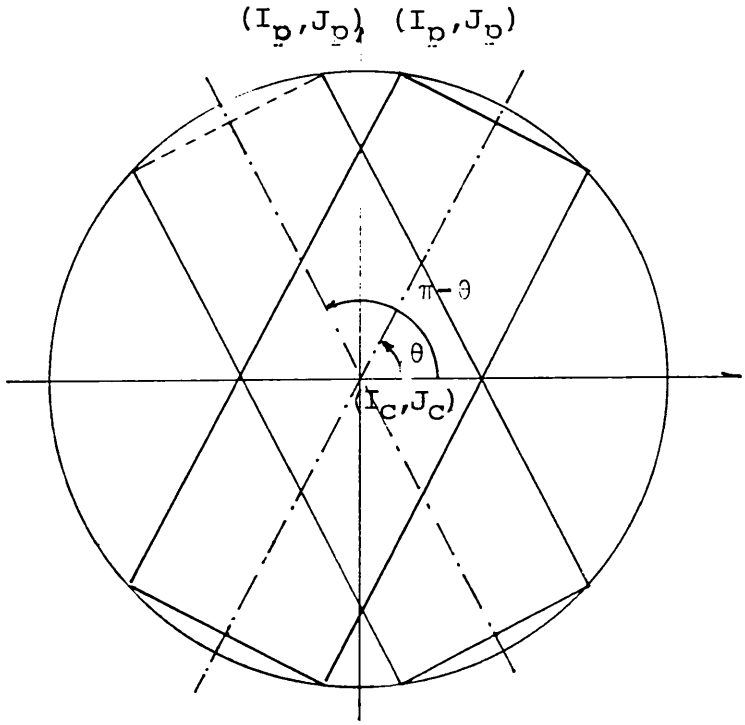


b) $P < T$ and $J_p < J_c$



c) $P < T$ and $J_p > J_c$

a) Limiting conditions for which $P = \ell \sin \alpha + w \cos \alpha = T$ (a constant)



d) $P > T$

Fig 5.5 Ambiguity in orientation for the θ and (π-θ) case

(I_p, J_p) is validated if

$$\sum_{i=I_p}^{I_p+2} \sum_{j=J_p}^{J_p+2} a_{ij} \geq 4$$

5.5 REMARKS

(i) The problem addressed in this chapter was a hypothetical problem, nonetheless, the solution reaffirms the view that vision problems still remain to benefit from unique solutions.

(ii) As perception of orientation constitutes one of the requirements, mere transformation of the incident image data into the projection set will not suffice. Additional information for implementation of the noise counter measure, of expression (5.8), has to be retained. Run length coding of the incident data, where the coordinates of successive transitions are stored (as in chapter 4) is a viable option. The computational requirements based on M68B09 microprocessor, for a 256 x 256 image are:

$(S)_{total}$ = Run-legth coded data + Projsection set + Work space

= 1Ø Kbytes + 2 x 256 Bytes + 1 Kbytes

≒ 12 Kbytes

$(t)_{total} \doteq 2$ sec.

(iii) The translation-orientation independent feature (Ψ) introduced in this work will exhibit excessive variability when the compact component set manifest aspect ratios in close proximity of unity.

(iv) The component sets can alternatively be viewed against a darker background, or via rear lighting. This necessitates initial application of a sequence of expand (PBIXP), and shrink (PBISH) commands, as appropriate prior to extraction of the projection set.

CHAPTER 6

A VISION BASED TECHNIQUE FOR AUTOMATIC TOOL PRESETTING, IDENTIFICATION, AND WEAR MEASUREMENT IN CNC LATHE

6.1 INTRODUCTION

The aim of the investigation presented herein is automation of tool operations via introduction of a visual link, in CNC lathe.

A new vision based technique is introduced to automate the tool set-up process. The proposed technique comprises a vision system and a procedure. The technique is then extended to tool identification and certain wear measurements while, essentially, retaining the same vision system.

The above contributions remain theoretical as the work does not extend to actual vision system realisation and its incorporation in a CNC lathe. Otherwise, methods, computational algoritms, as well as various possible configuration for the vision system, their expected performance and the nature of interface to the CNC system are included, in such a way to provide a clear guideline for subsequent implementation of the technique in a CNC lathe.

6.2 CRITICAL EXAMINATION AND RELEVANT OBSERVATION OF THE PROCESS

The following observations are pertinent to the formulation of the technique.

a. The tasks of

(1) tool set-up,

(2) tool identification, and

(3)tool wear measurement

can be performed sequentially as ordered above, (i.e. such a sequence is not incompatible with the objective of the overall operation).

b. The motion of the tool is confined to the horizontal plane P, as depicted in Fig.6.1, through axial and lateral movements along x and z axes.

c. The tools, in their plan view, present a summit point that lies at a minimum distance from the z axis. (The technique can also cater for instances where the tool presents several or even a line as a series of summit points).

d. Every tool assumes a known orientation in the tool

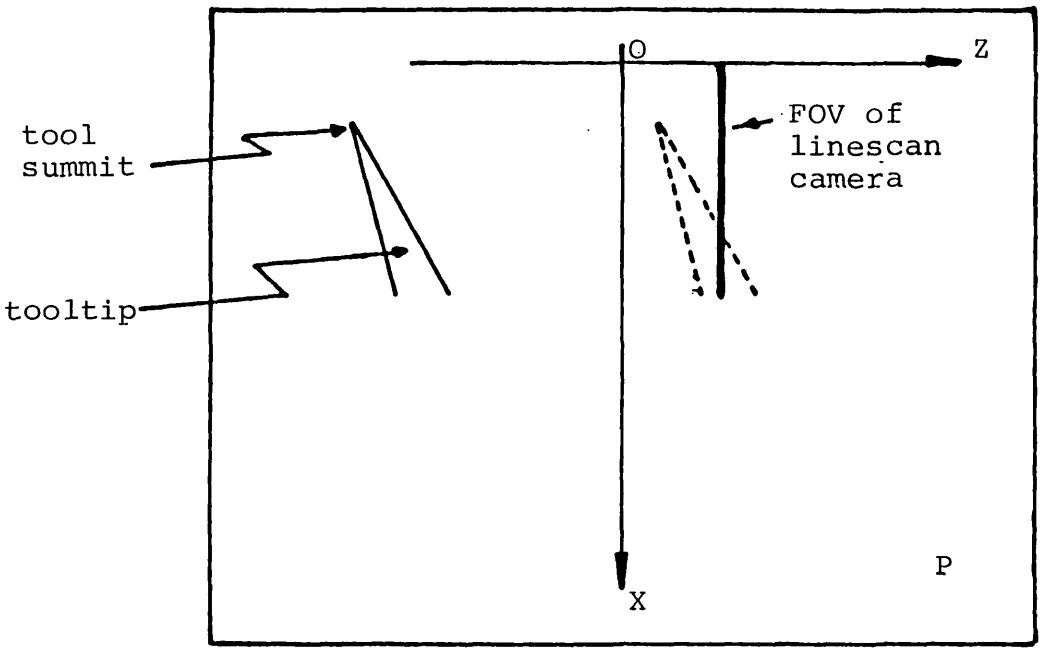


Fig. 6.1. Horizontal plane 'P' containing the motion of the tool.

holder and retains this orientation throughout the operation cycle.

e. The location and identity of the tools on the supporting turret is predetermined and known to the system (i.e. to the supervisor/ master program).

As the desired tool is placed in the tool holder the tool recognition problem reduces to verification of identity of the chosen tool.

f. A worn tool may be viewed as one that has lost material beyond an acceptable limit in the vicinity of the cutting edge, unless material build-up has occured at the cutting edge. (The presented technique is principally concerned with the tool wear that is manifested through its plan view, but also addresses the flank wear detection and the "built-up edge" phenomenon.)

6.3 OVERVIEW OF THE PROPOSED SYSTEM

Throughout this chapter, the vision system is considered in the context of a sensor and a processor.

A solid state line scan camera forms the sensor. Selection of such a camera was motivated by the need for a very high resolution.

A 2048-element linear array will provide adequate resolution to locate the tool to an accuracy of 10 μm in a Field Of View (FOV) of 20 mm. Further reduction in FOV is only a pre-requisite for attainment of higher accuracy, as viability of visible light becomes precarious when the desired resolution approaches its wavelenght.

The camera is installed overhead, and image acquisition is envisaged to take place under controlled conditions leading to a binary threshoded image.

The processor may assume different configuration on the basis of the prevailing time constraint and the nature of the operation. When the time constraint is not stringent a microcontroller is deemed best to serve the purpose.

Whether the processor should be a microcontroller configuration or a sequential logic circuit form can best be discussed after a description of the operations involved.

6.4 TOOL SET-UP

Tool set-up refers to the process of establishing the precise location of the tool with respect to a datum.

The significance of tool set-up can best be judged by the importance attached to the precision of machined products, as any inaccuracy in tool set-up will be directly reflected on the workpiece quality.

6. Vision in automation of CNC turning centres

A variety of methods are currently available for tool set-up, but they invariably rely on intervention of the human operator and in some cases additional sophisticated equipments. Daneshmend in his survey of existing methods in [33] reports that the position transducers used in setting-up of tools, produce readout (or feadback in closed loop systems) based on the position of the tool holder rather than the tool itself. Such methods necessitate accurate presetting of the tool within the tool holder prior to the actual exercise of tool set-up. These methods extend from template viewed by unaided eye to the use of microscope. The attainable accuracies here vary considerably, typically 200 μm in the former case to 10 μm in the latter.

Position transducers are inadequate for this purpose as their information is solely based on the position of the tool holder rather than the tool itself. Furthermore the more accurate of these methods are time consuming as they need to be performed outside the turning centre, and in large turning centres where the cost of machine down-time is the pre-eminent factor, their disadvantage is apparent.

Touch trigger probes have reportedly claimed some success. They have been known to produce a detectable signal for as low a deflection as 10 μm. Such a figure cannot necessarily be attributed to their accuracy as they respond even when met inadvertantly by an erroneous point of the tool.

The author was recently informed of a the IPL (Integrated

Photomatrix Limited) project "Tipscan", which aims to automate certain functions in machine-tools. To this date no information pertaining to the nature of operation and algorithms therein has been published*, and direct communication from IPL [34] reveals and verifies that it has not yet concluded its conceptual phase.

4.1 Method of set-up

The steps necessary for effecting tool presetting aspect of the technique are:

Step1. The tool is positioned to one side, say to the left, of the FOV. [coarse alignment can be achieved by sliding the tool holder to an extreme point to meet a micro switch or alternatively by the vision system at the expense of resorting to additional optics (mounted, say, on a motorised turret alongside the precision optics).]

Step2. The tool's summit is brought to the periphery of the camera's FOV, as depicted in Fig.6.1. It is then moved laterally past the camera at uniform speed, until it leaves the FOV.

Let the whole scene be represented by L

$$L = \{L_j : j=1,\ldots n\}$$

where

$L_j = \{a_{ij} : i=1,\ldots,m\}$ constitutes a single frame, (with m=2048).

* Verified by direct enquiry from IPL (June 1983)

The objective of this step is to compute the tool summit's ordinate δ_{min} .

There are two alternative approaches available in computation of δ_{min} .

a. **Through direct measurement of successive** δ_j**'s,** where δ_j denotes the first tool point in frame L_j . Then

$$\delta_{min} = \min_{j = 1,\ldots,n} \left[\delta_j \right]$$

Assignment of the first logical 1 to the tool body would render the method excessively susceptible to noise. Therefore as a practiacal measure, image data due to successive line scans are convolved with the same operator as that met in chapter 4, i.e.

δ_j is validated

iff $\sum_{i=\delta_j}^{\delta_j+2} a_{ij} = 3, \qquad j=1,\ldots,n$

b. **From the orthogonal projection set-** The scene data are initially transformed into a row sum projection vector (see chapter 5)

$$RS = (rs_1, rs_2, \ldots, rs_m)$$

As a noise counter measure, the isolated row sum elements are removed.

The first non-zero row sum element, rs_f, will reveal the ordinate of the tool tip, with

$$f = \delta_{min}$$

[$rs_f > 1$ is indicative of summit edge.]

Step3. To find the abscissa of the summit point, the tool is returned through the same path (i.e. laterally) to the FOV of the camera. Here, the lateral motion of the tool is assumed to be governed by the actuating signal 'v(t)', where:

$$v(t) = \begin{cases} 1 & \delta_j - \delta_{min} > 0, \quad \delta_j - \delta_{j-1} < 0 \\ -1^* & \delta_j - \delta_{min} > 0, \quad \delta_j - \delta_{j-1} > 0 \\ -1 & \delta_j - \delta_{min} = 0, \quad \delta_j - \delta_{j-1} = 0 \\ 0 & \delta_j - \delta_{min} = 0, \quad \delta_j - \delta_{j-1} < 0 \end{cases} \qquad (6.1)$$

Thus the tool comes to a halt with its tool summit point referenced at a point whose abscissa is the same as that of the line scan camera and its ordinate is equal to δ_{min}.

6.4.2 Speed of operation and processor configuration

Defenitions:

---

* v(t) = -1 is indicative of reversal in tool motion.

S = memory requirement

S_f = storage for frame buffer

S_{wk} = memory work space

t = time to complete a process

t_f = processing time per frame

t_a = time to establish the abscissa of the summit point, as in Step 3 above

T_L = integration (line scan) time of linear array

ρ = resolution

n = lateral span of the scene (in pixels)

b = span of the scene of intereset

Image due to a full scene comprises 2Ø48 x 2Ø48 pixels. Storage of the full frame or any major proportion of it is neither viable nor necessary.

Automatic tool set-up via Step2(a), above, lends itself to implementation by a sequential logic circuit, as depicted in Fig.6.2. This has the advantage of performing the processing almost as fast as data can be acquired. From 6.4.1

$$(t)_{\delta min} = n (T_L + t_f)$$

$$(t)_{set-up} = (t)_{\delta_{min}} + t_a$$

$$Ø \leq (t)_a \leq (t)_{\delta min}$$ [according to where the summit occurs within the span]

$$t_{\delta min} \leq (t)_{set\text{-}up} \leq 2 \cdot t_{\delta min}$$

With

$$b = 2\emptyset \text{ mm} \quad \text{and,} \quad T_L = 4 \text{ msec,}$$

$$n = \frac{b}{\rho} = 2\emptyset\emptyset\emptyset$$

$$t_f = \emptyset, \text{ [as processing is performed within } T_L \text{]}$$

$$8 \text{ sec} \leq (t)_{set\text{-}up} \leq 16 \text{ sec}$$

As evident, when processing is performed via the logic circuit, the predominant factor in stretching the tool set-up period is the integration time of the linear array.

In instances when the time constraint is substantially relaxed, a microcontroller can be used instead of the logic circuit of Fig.6.2. Adoption of a microcontroller is attractive as its versatility allows it to be used in the subsequent stages of tool identification and wear measurement. Fig.6.3 shows the overall configuration of such a processor. As evident transfer of image data when adopting a microcontroller is to take place via a DMA arrangement.

Computation of successive S_j's proceeds immediately after the acquisition of respective L_j's.

The memory and time requirements are as folows:

$$(S)_{set\text{-}up} = S_f + S_{wk}$$
$$= 2 \text{ kbytes} + 1/2 \text{ Kbyte}$$
$$\doteq 2 \text{ x } 2\text{-Kbyte RAM chip}$$

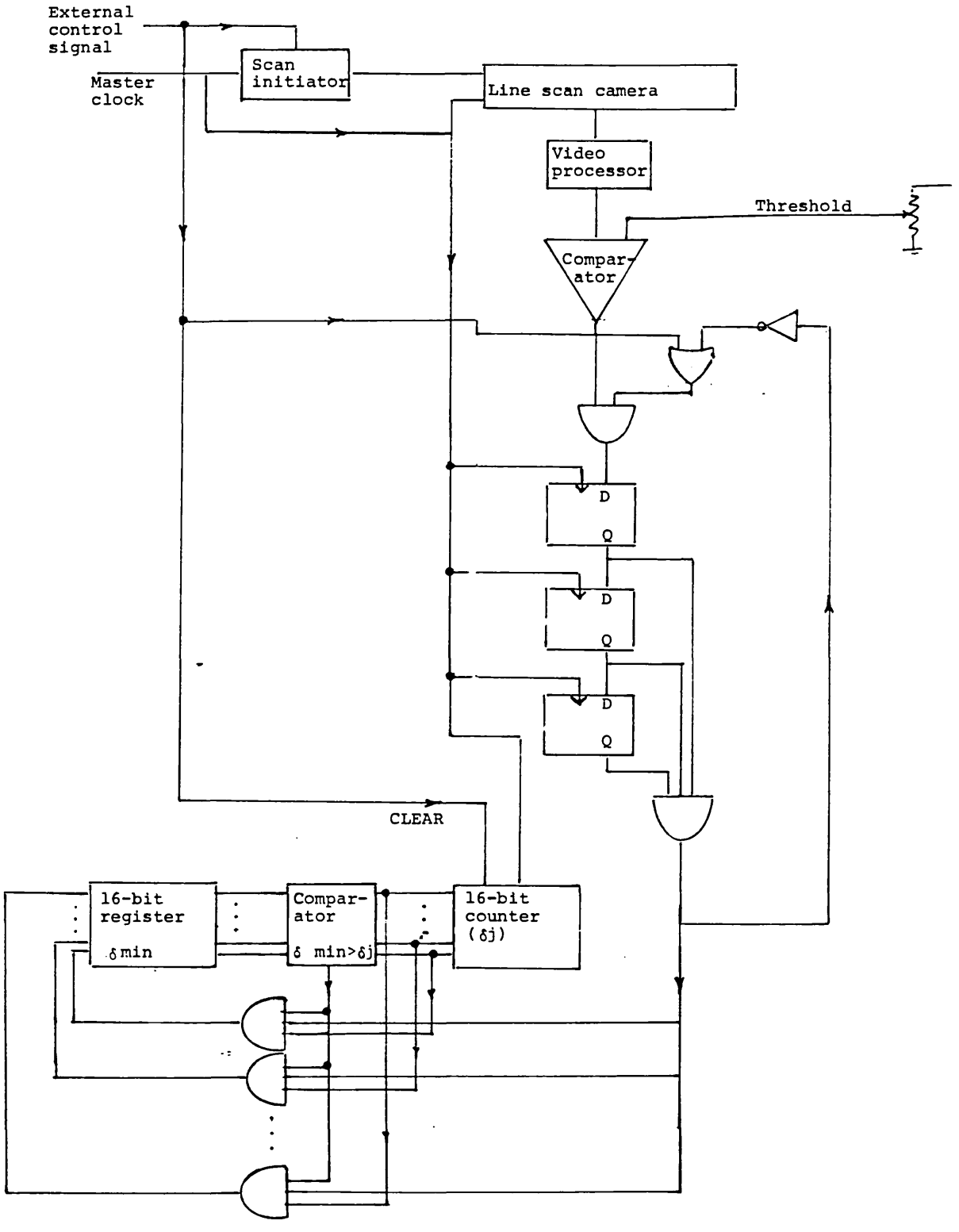


Fig. 6.2. A depiction of logic circuit implementation of the processor.

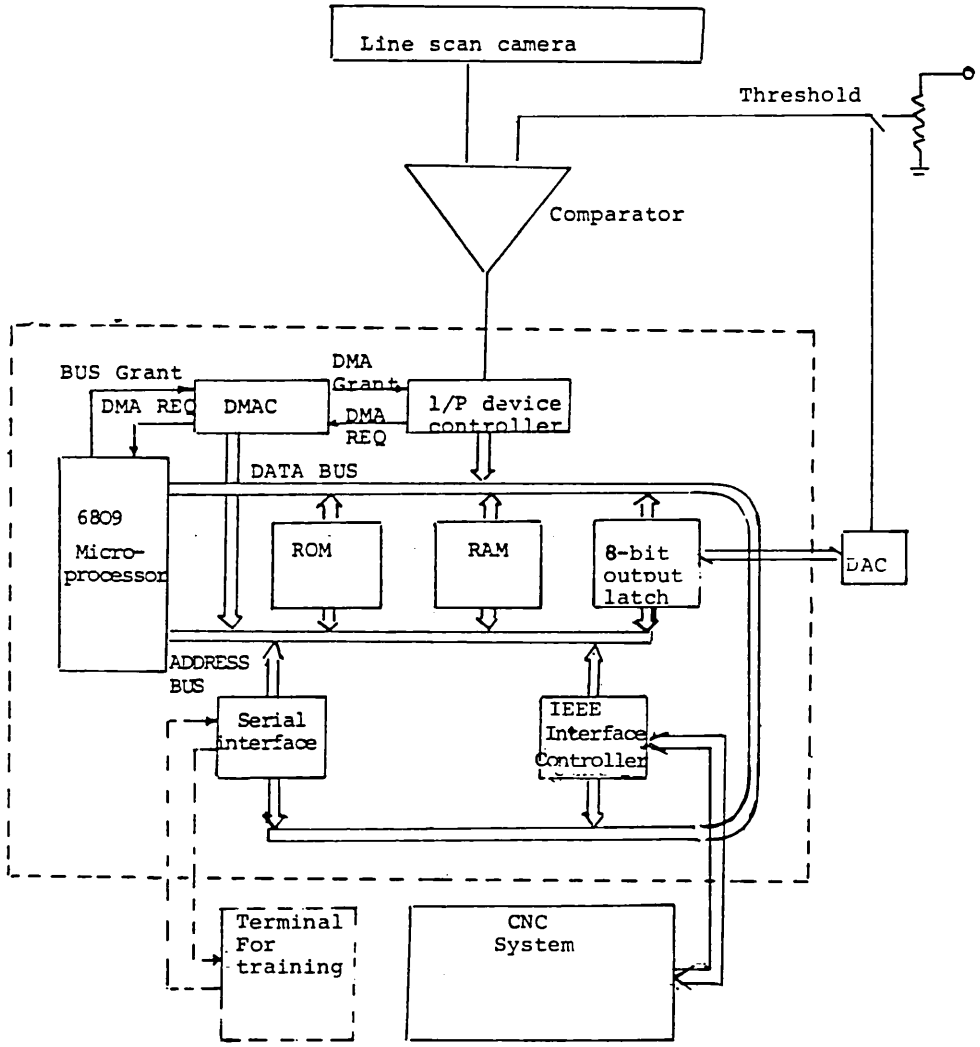


Fig. 6.3. Microcontroller configuration.

= 4 Kbytes

T_L = 4 msec [as before]

t_f = 12 msec = 3 . T_L [based on MC68B09 micro processor]

$$(t)_{8min} = n (T_L + t_f)$$
$$= 4 n T_L$$
$$= 32 \text{ sec}$$

Thus

$$32 \text{ sec} \leq (t)_{\text{set-up}} \leq 64 \text{ sec}$$

Tool set-up via the alternative route of Step2(b) is best suited for microcontroller implementation, exactly in the same way as explained above. In this case the processor requirements become

S_{wk} = 1/2 Kbyte + 2 Kbytes (storage of row sum vector)

$S = S_f + S_{wk}$

= 2 Kbytes + 2½ Kbytes

≒ 6 x 2-Kbyte RAM chip

= 12 Kbytes

$t_f = 3 T_L$ as in the previous case

Hence

$$32 \text{ sec} \leq (t)_{\text{set-up}} \leq 64 \text{ sec}$$

6.5 TOOL IDENTIFICATION

Tool identification is a vital aspect of machine-tool operation, as incorrect choice of tool can have have reprecussions even beyond that of a damaged workpiece.

6.5.1 Method of identification

The tool recognition problem, as indicated earlier, can be relaxed by virtue of the fact that we merely aim at verifying the identity of the specimen (i.e. the loaded tool) against that of the desired tool. Fig.6.4 shows typical tool shapes.

6.5.1.1 Feature selection The proposed features fall into two catagories:

a. **Those extracted directly from the incident binary data.** The width of the tool at predefined distances from the summit (β_i), as depicted in Fig.6.5, are selected as successive feature elements, such that

$$\beta_i = \sum_{j=J_{ki}}^{J_{ki}+q_{ki}} a_{ij}, \quad i \varepsilon \eta_a^{(k)}, \quad k = 1,\ldots,M \qquad (6.2)$$

where

$\eta_a^{(k)}$ = the set of distances from summit point at which β_i's are measured, when the tool of interest is of class ω_k .

To increase the discriminant power of these feature elements β_i's are expanded into a 3-tuple measurement

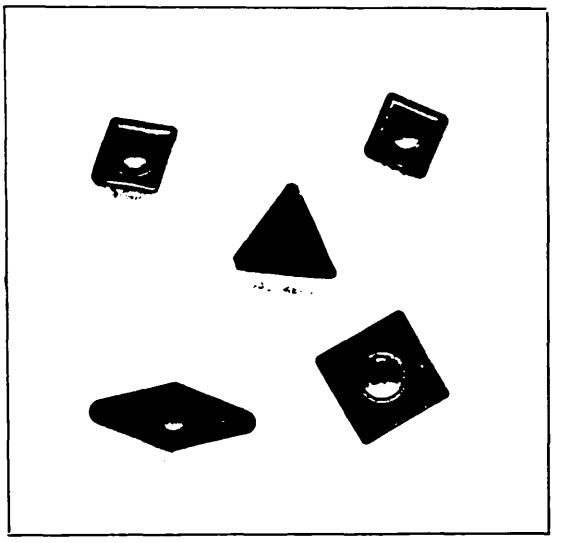


Fig. 6.4. Typical tool shapes

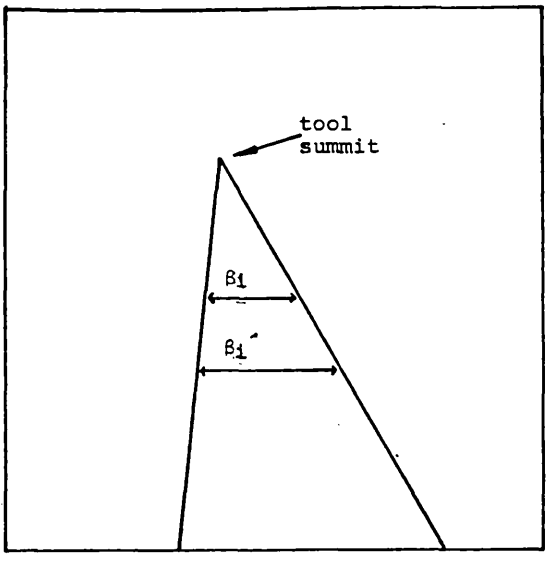


Fig. 6.5. Proposed feature elements for discrimination of tool geometry

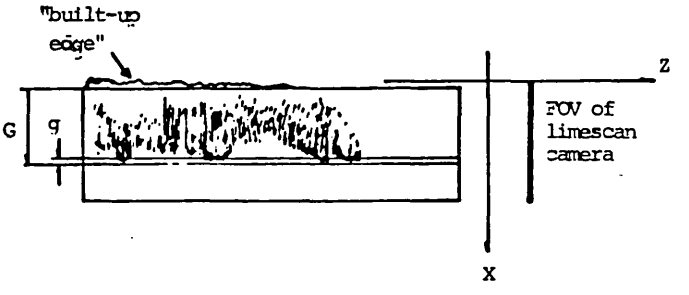


Fig. 6.6. View exhibiting flank wear along with associated parameters

$$\phi_i = \left\{ \sum_{j=J_{ki}-3}^{J_{ki}-1} a_{ij}, \beta_i, \sum_{j=J_{ki}+q_{ki}+1}^{J_{ki}+q_{ki}+3} a_{ij} \right\} \qquad (6.3)$$

A point worthy of note is that the composition of the feature vector for different tool shapes may vary with different tools. Moreover, new additions to the tool catalogue may necessitate augmentation of the existing feature vectors by new elements. This is to ensure that the feature vectors can still discriminate between the various tools in the catalogue.

This feature can prove effective in discrimination of many tool shapes but not all. Extraction of subsequent feature elements could require tool repositioning, to allow β_i measurement at i>m.

b. **The alternative feature elements are extracted from the orthogonal projection set, forming a series of 2-tuples:**

$$\int_l (rs_i, cs_j) : i,j \varepsilon \eta_b^{(k)} \int^l$$

where

$\eta_b^{(k)}$ = the set of i's and j's at which the row sum

and column sum vectors are measured as elements of the feature vector. This will be peculiar to the tools of class ω_k .

The projection vectors due to each class of tool exhibit minimal variability as the tools retain a fixed orientation and also they appear at a prespecified position when identification is in progress. However, there may remain a need to augment the feature vector with additional elements when "non-unique" projection vectors relate to more than one realisable tool shape. The latter features can in many instances be extracted from the projection sets of repositioned tools.

6.5.1.2 Classification Classification is based on the following non-linear distance metric. When the feature elements are those expessid in Eq.(6.2),the distance measurement reduces to

$$|\Phi_i^{(k)} - \phi_i| = \begin{cases} 1 & |B_i^{(k)} - \beta_i| > T_i^{(k)} \\ 1 & \sum_{j=J_{ki}-3}^{J_{ki}-1} a_{ij} > 2 \\ 1 & \sum_{j=J_{ki}+q_{ki}+1}^{J_{ki}+q_{ki}+3} a_{ij} > 2 \end{cases}$$

$$|\Phi_i^{(k)} - \phi_i| = \begin{cases} 0 & |B_i^{(k)} - \beta_i| \leq T_i^{(k)} \\ 0 & \sum_{j=J_{ki}-3}^{J_{ki}-1} a_{ij} \leq 2 \\ 0 & \sum_{j=J_{ki}+q_{ki}+1}^{J_{ki}+q_{ki}+3} a_{ij} \leq 2 \end{cases}$$

where

Φ_i represents the 3-tuple defined in (6.2) for the prototype tool, and

B_i the respective width of the prototype tool.

Verification of the tool's identity is based on the following decision rule

$$D(\phi) = D_t \text{ iff}$$

$$\Delta(\underline{\Phi}^{(t)}, \underline{\phi}) = \sum_{i \varepsilon \eta_b^{(t)}} |\phi_i^{(t)} - \phi_i| = 0$$

Otherwise the tool insert is rejected as not belonging to the desired class.

Through a training phase the following set of values are established.

$$\{B_i^{(k)}, J_{ki}, q_{ki}, T_i^{(k)} : i \varepsilon \eta_a^{(k)}; \quad k = 1, \ldots, M\}$$

For the alternative feature set of expression (6.3), let the following notation prevail:

$\Phi_i = [rs_i]$ prototype

$\phi_i = [rs_i]$ tool insert

$\Psi_j = [cs_j]$ prototype

$\psi_j = [cs_j]$ tool insert

then

$$|\phi_i^{(k)} - \phi_i| = \begin{cases} 1 & |\Phi_i^{(k)} - \phi_i| > \tau_i^{(k)} \\ 0 & |\Phi_i^{(k)} - \phi_i| < \tau_i^{(k)} \end{cases} \qquad (6.7)$$

$$|\Psi_j^{(k)} - \psi_j| = \begin{cases} 1 & |\Psi_j^{(k)} \psi_j| > \tau_j^{(k)} \\ 0 & |\Psi_j^{(k)} - \psi_j| < \tau_j^{(k)} \end{cases}$$

$$i,j \varepsilon \eta_b^{(k)}$$

The identity of the tool insert is verified

$$\text{iff} \sum_{i \varepsilon \eta_b^{(k)}} |\phi_i^{(k)} - \phi_i| + \sum_{j \varepsilon \eta_b^{(k)}} |\Psi_j^{(k)} - \psi_j| = 0 \qquad (6.8)$$

6.5.2 Processor configuration and performance

Identification when pursuing either alternatives is based on processing of successive frames immediately after their acquisition. This approach allows adoption of the same controller as in the tool set-up case, Fig.6.3.

To comply with requirement of the previous arrangement, each frame of 2048 pixels is transferred to the RAM via the DMA arrangement, although the resolution with which the scene for identification is examined need not exceed 40 μm.

When pursuing the first alternative, the processing requirement may be summed up as follows:

$$(S)_{ID}{}^{*} = S_f + S_{wk}$$
$$= 2 \text{ Kbytes} + 1/2 \text{ Kbyte}$$
$$\doteq 2 \text{ x } 2\text{-kbyte RAM chip}$$

$$n = \frac{b}{\rho} = 500$$

$T_L = 4$ msec

$t_f = 4$ msec $= T_L$ [The 4 msec allowance for processing of each frame is ample, yet necessary for the camera to operate in a continuous scanning fashion.]

---

* ID: Identification.

$$(t)_{ID} = n\ (\ T_L + t_f\ )$$
$$= 2\ n\ T$$
$$= 4\ sec$$

Obviously, tool repositioning for extraction of additional β_i's will double the estimate.

Identification from projection set can equally be performed with the above controller arrangement. In this case

S_{wk} = 1/2 Kbyte + 4 Kbytes [The latter, 4 Kbytes, is for retaing the projection set]

$$(S)_{ID} = S_f + S_{wk}$$
$$= 2\ Kbytess + 4\tfrac{1}{2}\ \ Kbytes$$
$$\doteq 4\ x\ 2\text{-kbyte RAM chip}$$
$$= 8\ Kbytes$$

$$t_f = 4\ msec = T_L$$
$$(t)_{ID} = n\ (\ T_L + t_f\ )$$
$$\doteq 4\ sec$$

The only advantage here is that the information content in projection vectors is more than the previous alternative, and in certain cases may obviate the necessity of tool repositioning.

Thus tool presetting and identification can share an

identical hardware.

6.6 TOOL WEAR MEASUREMENT

Worn tools adversely affect the precision of machined products. Avoiding the use of worn tools by frequent tool replenishment, contributes to highly undesirable machine down time. There are a variety of methods that have been developed for wear measurement [53]. They range from estimation of a tool's life, based on its past performance, to measurement of machined parts or monitoring of feed force during the machining process. However, none of these methods has achieved a widespread use.

Having acquired a vision system to automate the tool set-up and identification processses, it is of interest to know the extent to which it can be of use in performing wear measurement.

6.6.1 Method of detecting tool wear

Wear detection is attempted when tool presetting and identification have been accomplished. Moreover, wear measurement is not conducted with a view to quantify wear, but to reveal the class membership of the specimen as far as belonging to either classes of worn or unworn is concerned.

6.6.1.1 The wear that is manifested throught the tool's plan view With a precise notion, of its position the tool is sited at a position peculiar to its identity. It is then moved past the line scan camera as before with the view to perform area measurement in the vicinity of the cutting edge. The measured area is then compared with that of the prototype tool and if discrepant by more than a prespecified value the specimen is dismissed as worn tool. i.e

$\chi^{(k)} \varepsilon \omega_{wt}$* iff

$$\sum_{i,j \varepsilon \eta_{wt}^{(k)}} a_{ij} \geq T_{wt}^{(k)}$$

where $\eta_{wt}^{(k)}$ represents the set of test pixels in the vicinity of the cutting edge and T_{wt} is a dedicated decision threshold for the tool class ω_k .

6.6.1.2 Flank wear and "built-up edge" detection

That facet of the tool which bears the flank wear remains obscured from the overhead camera discussed so far. Hence a second line scan camera with lower element population of 128 elements is deemed necessary not only to view the flank wear but also to inspect the "built-up edge" that is manifest from the same side, as depicted in Fig.6.6.

The area of interest as depicted, in Fig.6.6, is

---

* w_t: Worn tool based on wear detectable from the plan view.

confined to a strip of width 'g' at a depth 'G-g' where 'G' is the critical wear depth which renders the tool worn. 'g' extends to a few pixels.

Segregation of the intact from the worn area, when inspecting the flank wear remains a critical process, if binary imaging is to be pursued.

The specific nature of the illumination and viewing technique for this phase remains to be perfected in the actual installation. The proposed approach aims at capitalising on specular reflectance of the intact surface as opposed to its diffusion when wear extends to the critical depth. Here the illumination source must be highly collimated, e.g. a laser, and so aligned to project only a fine bright spot onto the critical strip, of Fig.6.6. The second linear array must naturally be so aligned to intercept the reflected radiation.

There, the problem remains to be the automatic alignment of the light source for different tool geometries. [Realignment of the camera can be avoided by recourse to a sufficiently wide FOV, at most at the expense of a 256-element instead of the initial 128-element array.] Such an arrangement would necessitate inclusion of an additional control circuit for rotation of the illuminant beam (say, via a rotating mirror) through prespecified angles, peculiar to identity of the loaded tools.

Assignment of tools to the class of worn or intact in flank wear inspection would use σ_f as feature, where

$$\sigma_f = \sum_{i=G-g}^{g} \sum_{j=1}^{n_f^*} a_{ij}$$

and $\chi \varepsilon \omega_w$ iff

$$\Sigma^{(k)}\dagger - \sigma_f > T_{wf}^{(k)}$$

otherwise it is usable.

The 'built-up edge can be detected in the same manner as the flank wear by examining a similar strip adjacent to the upper bound of the tool profile, as shown in Fig.6.5 Distinguishing the built-up edge (i.e. material built-up due to work piece) from the background may require another dedicated threshold which can be derived as in previous cases.

6.6.2. Hardware aspect in wear measurement

If the 68Ø9 based controller, used in the earlier for tool set-up and identification, is also applied to this task, the memory and time requirements will be as follows:

a. Wear detectable from plan view " w_t"

Let Span of cutting edge, b = 2Ø mm

Depth over which area is measured, d = 4Ø µm

---

* $\pi_f = \frac{b}{\rho}$

† $\Sigma^{(k)}$: σ_f for prototype tool

$$\rho = 10\ \mu m$$

$$\eta_{wt} = \frac{b}{\rho}\,\frac{d}{\rho}$$

$$= 8000 \text{ pixels}$$

$$(S)_{wt} = n\, S_{\eta_{wt}} + S_{wk}$$

$$= 8 \text{ Kbytes} + 1/2 \text{ Kbyte}$$

$$\doteq 5 \text{ x } 2\text{-Kbyte RAM chip}$$

$$= 10 \text{ Kbytes of RAM}$$

$$T_L = 4 \text{ msec}$$

$$(t) = n\, T_L + t_f$$

$$= 2000 \cdot 4 \text{ msec} + 20 \text{ msec}$$

$$\doteq 9 \text{ sec}$$

b. Flank wear

With $g = 50\ \mu m$, $b = 5mm$

and $\rho = 10\ \mu m$

$$\eta_{wf} = \frac{g}{\rho}\,\frac{b}{\rho}$$

$$= 2500 \text{ pixels}$$

$$(S)_{wf} = \eta_{wf} + S_{wk}$$

$$= 2\tfrac{1}{2} + 1/2 \text{ Kbytes}$$

$$= 3 \text{ Kbytes}$$

$$\doteq 2 \text{ x } 2\text{-Kbyte of RAM chip}$$

$$= 4 \text{ Kbytes}$$

$$(t)_{wf} = n\ T_L + t_f$$
$$= 500 \times 4 \text{ msec} + 20 \text{ msec}$$
$$\simeq 2\tfrac{1}{2} \text{ sec}$$

Built-up edge

Similar estimates as in the flank wear case apply here.

6.7. SIMULATION

The viability of the technique, discussed so far, has been assessed using the development system of chapter 2. Simulation of the line scan camera was achieved by data acquisition through only a single column.

To compensate for the high resolution not attainable in the development system, recourse had to be made to larger models instead of the actual tools.

In the absence of a motorised platform for traversing the tool model across the FOV, the model was moved manually in the tool set-up case. But as uniform motion of the tool is a prerequisite for successful tool identification and its wear detection, data were acquired through a number of successive digitising columns, (i.e. the desired relative motion was achieved through movement of the line scan camera rather than the specimen).

The simulation did not run to the extent of flank wear and built-up egde detection as special lighting arrangement and viewing technique had to be adopted.

6.8 REMARKS

(i) The investigation presented in this chapter is intended to provide sufficient guidelines for realisation of a vision system within a CNC lathe , primarily to automate all tasks short of the flankwear and "build-up edge" detection.

(ii) The processes whose automation were undertaken in this chapter are all marked for their high measurement accuracy requirement.

The 10 μm accuracy aimed for the tool presetting, practically touches the limit of visible light effectiveness as a measurement tool. Such an accuracy can only be secured through highly controlled illumination yielding favourable tool-background contrast leading to binary imaging.

(iii) Although uniform tool motion is mandatory in the identification and wear measurement case, it is not imperative for presetting.

CHAPTER 7

SUMMARY AND CONCLUSIONS

The central theme of this thesis has been, that the complexity of a robotic vision system grows with that of targets and scenes that it must confront. Moreover, flexibility in the sense of coping with extensive catagories of targets in a variety of supporting scenes can render the industrial vision system prohibitively expensive. Thence, to expedite design and development of "special purpose" vision systems a methodology was adopted. This was based on the use of interactive image analysis for prototyping robotic vision systems. To this end, an interactive image analysis package was designed and implemented. Pursuing this methodology, the investigation was extended to specific industrial problems.

The main achievements of this research may be summarised as follows:

- Design and development of an interactive image analysis package, for prototyping (particularly low-cost) industrial robotic vision systems (chapters 2 and 3).

- Design and development of a vision system for recognition of engine bearing cap sets and automation of their handling process, in the automotive industry (chapter 4).

7. SUMMARY AND CONCLUSIONS

- Theoretical extension of the Wald's Sequential Probability Ratio Test to a parametric classifier (section 3.5.5.1).

- Derivation of a translation, orientation, size invariant feature as well as derivation of expressions for computing the orientation of a compact engine bearing cap set from their respective orthogonal projections (chapter 5).

- Formulation of a new vision-based techhnique for automating tool operations in CNC lathe (chapter 6).

The significance of these contributions has been discussed in the respective chapters.

Each of the above areas presents a growth potential for expansion. The scope of CAVDEP is extremely wide, and its modular structure lends itself to easy expansion with minimal impact. Its prime deficiency stems from the quantisation levels of the digitiser (4 bits). This confines its operation to high contrast scenes. Replacement of this unit with at least a 6-bit digitiser is suggested as an immediate step.

The evaluation module can particularly benefit from addition of routines which evaluate preprocessing operators. This entails introduction of commands which degrade the acquired image by injection of pseudo random signals.

7. SUMMARY AND CONCLUSIONS

The installed vision system at Ford can readily benefit from the novel method of automatic position alignment, discussed in section 4.2.1.3. The gantry arm is, currently, aligned by being referenced against micro switches, requiring more than a minute. The proposed method will provide the same information on real-time basis, without interfering with the operation of the gantry. Furthermore, the capability of this system can also be extended to computation of the pallet orientation by recourse to tapered markers, rather than the current pallet location markers. The method will be similar to positional alignment of the gantry, with the pallet moving to present two such markers successively to the camera.

REFERENCES

1 Gonzales,R.c. and Wintz,p.Digital image processing. Addison-Wesley, 1977.

2 Slater,P.N. Remote sensing optics and optical systems. Addison-Wesley, 198Ø.

3 Stafford, R.H., Digital television- bandwidth reduction and communication aspects, Wiley, 198Ø.

4 Hunt,B.R."Digital image processing". in Oppenheim, A.V.(Ed.). Application of digital signal processing. Prentice-Hall, 1978.

5 Mucciardi,A.N. and Gose,E.F. "A comparison of seven techniques for choosing subsets of pattern recognition properties", IEEE Trans, Comput., vol, C-2Ø, pp, 1Ø23-1Ø31(SEP. 1971).

6 Duda,R.O. and Hart,P.E. Pattern classification and scene analysis. Wiley, 1973.

7 Mason,C.J.W. and McFall,C.H. "Some experience with the method of potential functions", i.e.e n.p.p.l Conference on pattern recognition, Teddington, England, 1968.

8 Rosenfeld,A. and Kak,A.C. Digital picture processing.

Academic Press, 1982.

9 J. Sklansky (Ed.), "Pattern recognition introduction and foundations", Hutchinson and Russ, 1973.

1Ø Seman,N.G. "Robots are molding a new image". in W.R. Tanner (Ed.), Industrial Robots. Robotics International, 1981.

11 Tanner,W.K. "Industrial robot today" in W.R. Tanner (Ed.). Industrial robots. Robotics International, 1981.

12 Batchelor,B.G., Marlow,B.K., Smith,B.V.D. and Werson, M.J. "A research laboratory for automatic visual inspection". Proc. 5th Int. Conf. on automatic inspection and product control, Stuttgart, pp. 13-33. (198Ø)

13 The Manual of Photography, revised by Jacobson,R.E., Focal Press, London, 1979.

14 Beynon,J.D.E. "Optical self scanned arrays". The Radio and Electronic Engineer, Vol. 49, No. 1Ø, pp. 493-5Ø2 (October 1979)

15 Wekler,G.P. "Operation of pn junction photodetectors in a photon flux integrating Mode", IEEE J. Solid-State circuits, Vol. SC-2, No.3, PP.65-73, (1976).

16 Bracewell,R.N. The Fourier transformation and its applications. Mcgraw-Hill, 1978.

17 Batchelor, B.G. "Interactive image analysis as a prototyping tool for industrial inspection". IEE J. of Computers and Digital Techniques, Vol. 2, No. 2, pp. 61-73 (April 1979)

18 Batchelor, B.G. and Brumfitt, P.J. "Command language for interactive image analysis", IEE Proc., Vol.127, Pt. E, No. 5, pp.203-218, (Sep. 1980)

19 Mott,D.H. "Developing and evaluating image processing algorithms". Sensor Review, pp. 78-81 (April 1981)

20 Prewitt, J.M.S. "Object enhancement and extraction" in Lipkin, B.S. and Rosenfeld, A. (Ed.), Picture processing and psychopictorics, pp.75-149, Academic Press, 1970.

21 Kitchin,P W and Pugh, A. " Processing of binary images" in Pugh, A. (Ed.), Robotic vision. IFS Publication, pp.21-42, Nov. 1983.

22 Hu, M.K. "Visual pattern recognition by moment invariants", IRE Trans. Inform. Theory, Vol.IT-8, PP.179-187(Feb.1962).

23 Hall, E.L., Computer image processing and recognition, Academic Press 1979- Page 476.

24 Andrews, H.C., Introduction to methematical techniques in pattern recognition, Wiley, 1972, Page 186-188.

25 Ullmann, J.R., Pattern recognition techniques, Butterworth, 1973.

26 Maali, F. and Besant, C.B. "Machine vision in recognition of engine components", Proc. 1st Int. Conf. Automated material Handling, London PP.161-172 (Apr.1983).

27 Pavlidis, T."Computer recognitionof figures through decomposition", Information and Control Vol. 12, pp.526-537, (1968).

28 Chang, S.K. "The reconstruction of binary patterns from their projections, commun. Ass. Comput. Mach. Vol. 14, PP.21-25 (Jan 1971).

29 Chang, S.K. and Chow, C.K. "The reconstruction of three-dimensional objects from two orthogonal projectionsand its application to cardiac cineagiography", IEEE Trans. Comput., Vol. C-22, No.1, PP.18-28 (Jan. 1973).

3Ø Shliferstein, A. and Chien Y.T. "Some properties of image-processing operations on projection sets obtained from digital pictures",IEEE Trans on Comput. C-26, No.1Ø, PP.958-96 (Oct.1976).

31 Ryser, H.J. "Combinatorial mathematics, Wiley, 1963- Chapter 6, Matrices of zeros and one.

32 Wang, Y.R. "Characterisation of binary patterns and their projections", IEEE Trans. Comput., PP.1Ø32-1Ø35 (Oct.1975).

33 Maali F., Daneshmend L., Besant C.B., Pak.A. "A vision-based technique for automatic tool set-up, identification, and wear measurement in CNC turning centers", Tech. Rep. No.RC1/Nov. 1982.

34 Communication from IPL to L.Daneshmend, Ref DJM/JMB/511 Dated: 24 June 1983.

35 Batchelor, B.G. "Pattern classification and data analysis" in Batchelor B.G.(Ed.), Pattern recognition ideas in practice. plenum, 1978.

Appendix A

PROCESSING IMAGERY ON THE ICCC MULTI- MAINFRAME COMPLEX

The tedious nature of assembly programming initially led to processing of the acquired imagery on the ICCC multi-mainframe complex. Image data still would be obtained through CAVDEP via the magtape unit supported by the PDP11 minicomputer system, as shown in Fig.2.1. This complex comprises a tightly coupled CDC Cyber 170 and a Cyber 720 further supported by a Cyber 174. The latter is a dedicated graphic facility. Each of these computers has 131 Kwords of 60-bit word of memory, yet only a minute fraction of this storage is available to programs that are designed to run interactively. Currently only 25 Kwords of memory are made available to interactive programs. This limit is raised 32 Kwords after 1800 Hr. This complex is supported by a large array of peripherals, commonly found in similar facilities. The principal advantage of the particular mainframe facility is the extensive support of graphical and library packages available to it. This latter feature is not necessarily shared by all mainframe computers.

To meet the taxing constraint of those catagories of "jobs" that qualify to run interactively, the routines were organised around a procedure file which is invohked by

entering its designation along with the identity of the image file. The use of a specific library routine allows parameters to be passed between the routines (which are written in Fortran) and the procedure file through three of the registers that the mainframe system supports. Interrogation of the registers at a certain node in the procedure file will lead to execution of the subsequent routines of the procedure file.

Appendix B

DISTANCE METRICS IN CLASSIFICATION

The distance between two N-dimensional metrics $\underline{\Phi}$ and $\underline{\Psi}$ grows with their dissimilarity, and can be described by a variety of metrics. The mertics vary in sensitivity and computational cost, and are selected to suit individual applications.

The City Block metric of chapter 3 and the widely used Euclidean metrics are special cases of Minkowski's distance metric [35]

$$\Delta_n(\underline{\Phi},\underline{\Psi}) = \left[\sum_{i=1}^{N} |\Phi_i-\psi_i|^n\right]^{1/n} \qquad (B.1)$$

(B.1) reduces to a City Block metric with n=1, i.e.

$$\Delta_1(\underline{\Phi},\underline{\psi}) = \sum_{i=1}^{N} |\Phi_i-\psi_i| \qquad (B.2)$$

When employing the latter in a minimum distance based classification scheme, the relative dominance of the successive feature elements must be established. In this work, when no added emphasis is attached to a specific feature element, then the distance function is computed as

$$\Delta_1(\underline{\phi},\underline{\psi}) = \sum_{i=1}^{N} \left| \frac{\phi_i - \psi_i}{\frac{1}{M}\sum_{k=1}^{M} \phi_i^{(k)}} \right| \qquad (B.3)$$

[ϕ and ψ here represent the prototype and sample feature vectors respectively.]

When there are a multitude of prototypes to each class, then their mean or simply their median is selected as in (B.3).

The Euclidean metric is derived when n=2 in (B.1), as

$$\Delta_2(\underline{\phi}, \underline{\psi}) = \left[\sum_{i=1}^{N} |\phi_i - \psi_i|^2 \right]^{\frac{1}{2}} \qquad (B.4)$$

Classification by any variation of the minimum distance scheme using the Euclidean metric still will employ a linear decision function.

Recourse can equally be made to non-linear distance metrics, as in chapter 6. Here

$$\Delta_{NL}(\underline{\Phi}, \underline{\Psi}) = \begin{cases} p & \Delta(\underline{\phi}, \underline{\psi}) > T \\ q & \Delta(\underline{\phi}, \underline{\psi}) < T \end{cases} \qquad (B.5)$$