Durham University

Durham E-Theses

Investigation of the processes required for the automation of stitchmarking in shoe manufacture

Nigel R Tout

How to cite:

Tout, Nigel R (1989) Investigation of the processes required for the automation of stitchmarking in shoe manufacture. Doctoral thesis, Durham University.

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a https://etheses.durham.ac.uk/id/eprint/6487/ is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full Durham E-Theses policy for further details.

Academic Support Office, The Palatine Centre, Durham University, Stockton Road, Durham, DH1 3LE
e-mail: e-theses.admin@durham.ac.uk Tel: +44 0191 334 6107
https://etheses.durham.ac.uk

Abstract

Investigation of the Processes Required for the Automation of Stitchmarking in Shoe Manufacture

by Nigel R. Tout

PhD Thesis 1989

This thesis describes a novel approach to the high speed automatic stitchmarking of shoe upper components by integrating an electrographic printer with a shape recognition system. A critical review of recognition system parameters selects the currently known shape parameters which are most suitable for use in a high speed recognition system with the large number of different shoe components found in a typical shoe factory. These are compared with the parameters actually used in the previously developed recognition system to be used for stitchmarking. A discussion of printing technologies suitable for marking shoe materials with computer generated patterns follows. It is concluded that an electrographic printer has the best combination of characteristics. There follows a description of experiments demonstrating xerography on shoe upper materials, and the design of a system integrating a low–cost laser printer to the recognition system which proved the concept of continuous automatic stitchmarking. With this performing satisfactorily, the system was converted to use a high speed printer requiring the use of an advanced graphics processor for handling the data transformations and interfacing with the printer. Modifications to the printer for operation with shoe materials are described, together with the need for a special toner. A full description of the resulting stitchmarking system is given, followed by details of analyses of its performance. Individual chapters are devoted to the accuracy, recognition efficiency, and the timing of the system. A potential bottleneck in determining the orientation of certain difficult shapes is identified, and faster methods for dealing with these are specially investigated. This concludes that the best approach is to optimise the present method and accelerate the calculations by using a more advanced microprocessor. A discussion of the general running of the machine includes details of problems which occurred with the modified printer mechanism and how these were overcome. Finally, suggestions are made for incorporation in an improved system capable of handling larger shapes with a tenfold speed increase.

The copyright of this thesis rests with the author.
No quotation from it should be published without
his prior written consent and information derived
from it should be acknowledged.

Investigation of the Processes Required for the Automation of Stitchmarking in Shoe Manufacture

by

Nigel R. Tout

**A Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy**

School of Engineering and Applied Science

The University of Durham
1989

- 3 APR 1990

DECLARATION

The work contained in this thesis has not been submitted eleswhere for any other degree or qualification and unless otherwise referenced it is the author's own work.

Copyright ©1989 by Nigel R. Tout
The copyright of this thesis rests with the author. No quotation from it should be published without N.R. Tout's prior written consent and information derived from it should be acknowledged.

ACKNOWLEDGEMENTS

This research project was funded jointly by the ACME (Applications of Computers to Manufacturing and Engineering) directorate of the SERC (Science and Engineering Research Council) and BUSM Ltd. (formerly the British United Shoe Machinery Co.).

The help given by personnel at Durham university was greatly appreciated. Included are Mr. T.Nancarrow for assistance with the computing equipment, Mr. J.Greensmith for assistance with electrical equipment, and Mr N.Dunning for invaluable mechanical work. The project was executed under the supervision of Dr. C.Preece and Dr. J.E.L.Simmons, to whom I am grateful for their guidance and encouragement. At BUSM, the collaboration of Mr. R.Yardley during the hectic phase with the exhibition machine was especially valuable. Especially, I would like to thank Mr. D.Reedman of BUSM, whose constant enthusiasm and encouragement promoted the smooth execution of the project

CONTENTS

LIST OF PLATES

LIST OF FIGURES

LIST OF TABLES

LIST OF COMPUTER PROGRAMS

ABBREVIATIONS AND NOMENCLATURE

ASCII	American standard code for information interchange.
A3	International paper size, 420 × 297 mm.
A4	International paper size, 297 × 210 mm.
BUSM	The collaborating company (formerly British United Shoe Machinery Co.).
CAD	Computer aided design.
CAND	The number of database shapes with root area and principal 2nd. moments within tolerance of a scanned shape.
CCD	Charge coupled device.
CRDPRS	The number of coordinate pairs to generate stitchmarking.
DOD	Drop on demand.
DOTCLK	Dot clock produced by the ionographic printer.
F	Lens focal length.
GCOEFF	'Goodness' coefficient, a measure of the elongation of a shape, $= (MMAX - MMIN)/(MMAX + MMIN)$.
GSP	TMS34010 graphics system processor.
HSYNC	Horizontal synchronisation signal.
IDIF	The count, when comparing the features of an unknown shape with a database shape, of the number of radii which do not match.
LED	Light emitting diode.
LCS	Liquid-crystal shutter.
LSI	Large scale integration.
M	Lens magnification.
M_{pq}	Moment of area of a shape of order x^p and y^q.

MMAX	Maximum principal second moment of area.
MMIN	Minimum principal second moment of area.
NCURVS	The number of stitchmark lines to be drawn.
PE-1620	Perkin Elmer 16-bit mini computer.
PRNT	Signal to the printer to initiate printing.
PVC	Poly vinyl chloride.
RAM	Random access (read/write) memory.
RCLCK	Return clock to the ionographic printer.
SDATA	Serial printing data to the ionographic printer.
SDB	TMS34010 software development board.
TTL	Transistor–transistor logic.
TMS34010	Texas Instruments graphics system processor.
U	Lens to image distance.
V	Object to lens distance.
VRAM	Video RAM (read/write memory).
VSYNC	Vertical synchronisation signal.
Z8000	Zilog 16-bit microprocessor.

CHAPTER 1

INTRODUCTION

1.1 Shoe upper manufacture

Shoe uppers are the parts of shoes above the soles which are shaped to cover the foot. The shoe upper components which are manipulated and assembled to make a 3–dimensional upper are initially cut from flat sheets, either of leather or synthetic material. Plate 1.1 shows the components which make up a typical ladies shoe upper.

The manufacture of shoe uppers, especially from leather, is characterised by the processing of components in small batches. This results from the large range of sizes of shoes, each of which requires its own size of shoe components, and the need for left and right of each. Rapidly changing fashions with their diversity of parts mean that there is not much scope for introducing inflexible automation into the upper manufacturing processes. This results in the processes of upper manufacture being performed usually by manually operated machines where the operators' dexterity reduces the need for retooling before each batch.

The only area where automation has made much impact is in decorative and some joining stitching, where automatic stitching machines are used. In these the pattern to be stitched is stored in computer memory, and is called up manually as required. However, the shoe components still have to be manually loaded into specially made jigs for each design.

Plate 1.1 *Components of a typical ladies shoe upper.*

1.2 The need for automation in shoe upper manufacture

There can be severe problems in operating a shoe factory manually using operator skill to eliminate the need for constant retooling. These are due to

- the lengthy times to train operators the necessary skills,
- variability of output quality with fatigue of the operator, and
- absenteeism causing loss of the skilled operator.

In order to make shoes of a certain style the following steps typically have to be performed. The 'clicker', the person who cuts out the components from the sheet material, is given a list which details all the parts required to make the size range and the quantity of each component. When cutting components from leather it is a part of the clicker's skill to select the appropriate part of the skin of leather for each type of component. Thus the upper components that need to be hard wearing are cut from the tougher parts of the skin, and the other parts are cut from the remaining parts of the skin. Also, the larger components are usually cut out first, with the smaller ones being cut from the material remaining. This results in the parts not being cut in a regular order, which requires the clicker to keep counting the number of each part in order to see when enough have been cut. When sufficient of any part has been cut they are bundled into a batch with an elastic band and labelled. The batches of components are transported to each station where they have to be unbundled, operated on, and rebundled, as depicted in fig. 1.1, [1]. When these initial operations have been performed on the components they are rebundled into kits of parts sufficient to make a pair of shoes prior to the assembly operations.

Working in this way leads to

- a large logistics problem in moving the components to the correct station at the correct time,
- a very large stock of work in progress, and

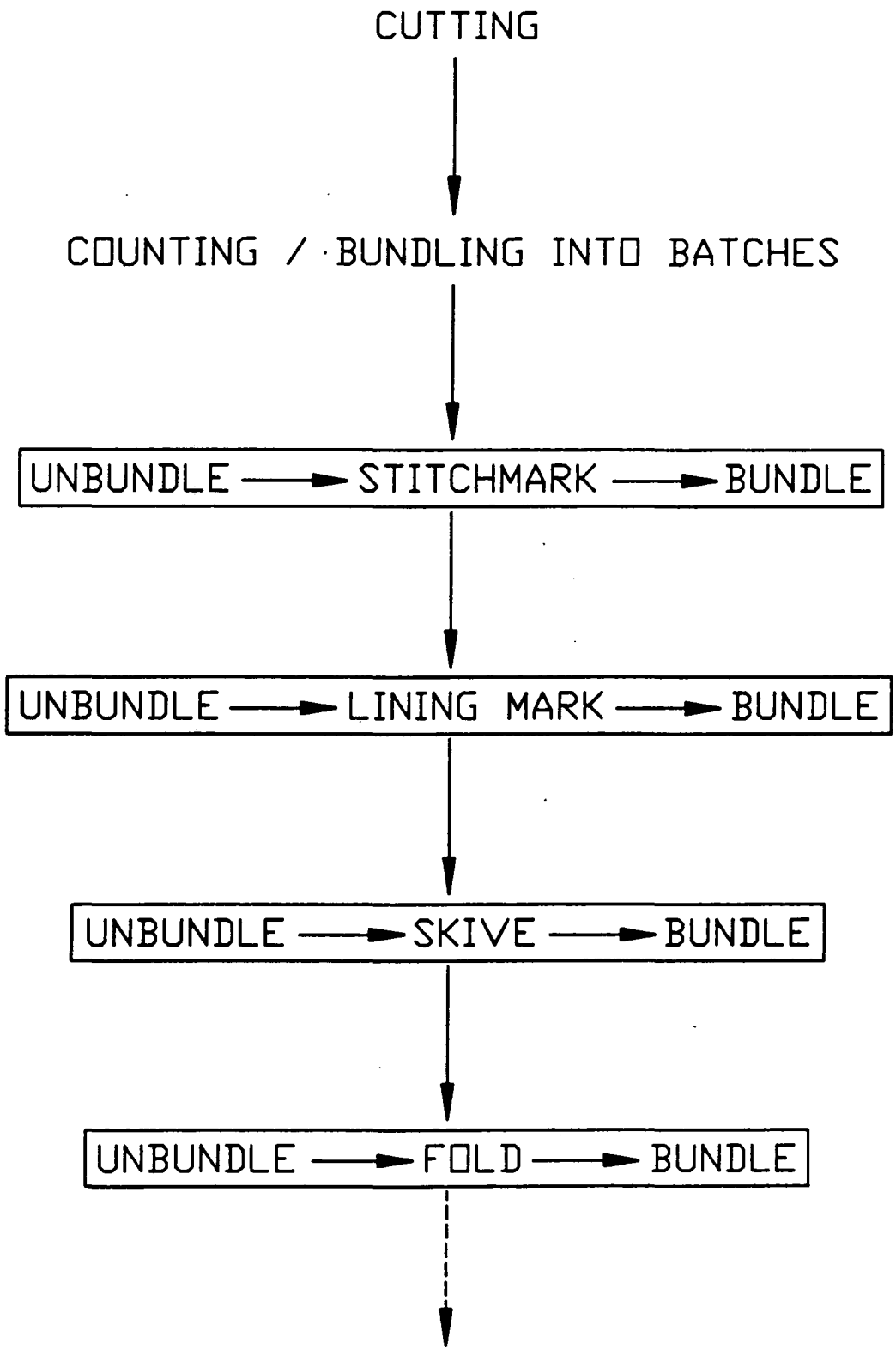


Figure 1.1 *Part of the progress of a batch of shoe components during upper manufacture.*

- a lot of time wasted in unbundling and bundling the components.

The basis of this project is the automation of the manufacture of shoe uppers by using one or more vision systems linked to individual operating stations, with conveyors transporting the shoe components from operation to operation. The system would have to be capable of accepting components in any order, and the automatic operations would have to be software controlled, so that no retooling would be required.

Since this would require radical changes in the operating procedures of shoe factories, and since the research required to produce a complete system is considerable, initially stand–alone machines which perform just one operation are planned.

1.3 Stitchmarking

Of the possible operations that could have been automated, it was decided that 'stitchmarking' would be the first. Stitchmarking is the application of marks, which may be temporary, to shoe upper components as guides for stitching, folding, pleating, joining, or the attachment of accessories. A related process that the machine might also perform is 'lining marking', which is the printing of size, style, and batch information on the linings of shoes.

With the planned complete automation of the manufacture of shoe uppers there will be little or no need for stitchmarking, since the stitching together would be performed fully automatically. However, the automated three dimensional assembly of shoe uppers is still many years away and would not be adopted universally. There will hence be a continuing demand for stitchmarking, and also undiminished demand for lining marking. Stitchmarking was also perceived as an operation which could be fairly easily automated, since there are a variety of printing technologies available. Also, it would be a useful operation for determining the accuracy of the position and orientation location functions of the recognition system.

As with the other processes of shoe upper manufacture, stitchmarking is basically a manual operation. The methods employed are

i) using silver coloured 'carbon' paper with a hand operated letterpress

press,

ii) using a water based ink impregnated pad with a hand operated letterpress press,

iv) using a silver ink ball–point pen to sketch by hand along slots in templates or,

v) using devices which prick or indent the surface of the components. These devices are mounted within the knives which are used to cut out the components.

The pricking method of marking has the disadvantages of damaging the shoe components, the devices within the cutting knives can become damaged, and the knives cost much more with the devices.

The stitchmarking lines on the components are either covered up by other components, stitched over, or removed. Water based stitchmarking can be cleaned off with a water based cleaner, and the others with a mild solvent cleaner.

All of the printing processes require the unbundling, loading, marking, unloading, bundling operations outlined in the previous fig. 1.1 [1] , and the need for retooling for each batch. The tooling also has to be made, maintained, and stored.

1.4 Automatic stitchmarking

An automatic stitchmarker needs to recognise the design, size, position and orientation of each shoe component and determine the appropriate stitchmarking. A printing mechanism is required which will mark leather and synthetic shoe components and is compatible with the recognition system. It is also almost certainly necessary to transport the shoe component from recognition to printing.

In a very few cases the same shoe component is given more than one stitchmarking pattern. There are examples of shoe factories [2] where, to minimise the number of cutting knives, the same components are used in more than one design of shoe, if possible. In such cases identical components

may require different stitchmarking on different occasions. For automatic stitchmarking there must either be means of informing the machine which stitchmark pattern to apply during the current period of production, or the use of one component in more than one design of shoe, requiring different stitchmarking, would have to be prohibited.

Initial work on vision systems required for the automation of shoe upper preparation started in 1975 at the City University, London. The basic recognition procedure was proved there, although the recognition process was unrefined and the computer system was slow [3] . The system resulting from this work was transferred to the research department at BUSM Ltd. (formerly the British United Shoe Machinery Company), the collaborating company, where the recognition process was refined and faster computers were used.

1.5 Specification

A specification for the operation of the proposed stitchmarking system resulting from this project in terms of recognition and printing was drawn up in consultation with BUSM. This was quite rigorous, since in the words of a BUSM report [1] 'stitchmarking is perhaps the most important operation [in shoe upper preparation] in that it sets the standard for the finished upper ...'. Although the specification was for a research machine, when this was completed it was expected that the way would be clear to designing a machine to be put into production which would have the high throughput of up to 3 to 4 components per second and maybe printing across a wider path.

The specification that was drawn up was :-

i) **Speed** – To have an average throughput of one component every 5 to 10 seconds, including recognition, transfer, and marking. The processes are to be designed so that they can be overlapped in the future to give greater speed.

ii) **Size range** – The width of the printer is to be at least 210 mm (i.e. A4 width). A width of 300 mm (i.e. A3 width) would be preferable if suitable equipment using the chosen printing technology can be

located. This preference should not be allowed to influence the choice of technology, or to delay the project.

iii) **Materials** – The printing process should be capable of marking at least a reasonable range of tanning finishes and, preferably, man–made surfaces without damaging them. The ink should be delible. The transfer mechanism should be capable of handling a range of substances, from glove and lining leathers to stiff working–boot leathers.

iv) **Interfacing** – The transfer and marking systems should be designed to interface with the existing shape recognition hardware and software fundamentals wherever this presents no great disadvantage.

v) **Accuracy and repeatability** – The sum of all registration errors arising from recognition, transfer, and marking should not exceed +/− 0.25 mm.

vi) **Data input** – The database of shapes and their stitchmarking will be assembled either by using the machine in a 'Teach' mode, or by downloading the information from a shoe CAD system.

vii) **Process extension** – Conceptual consideration will be given to means of overcoming speed and size constraints, and means of dealing with process output. (e.g. sorting, stacking, passing to another automatic operation, &c).

1.6 Layout of the thesis

The recognition system that was to be used with this research project had been developed over many years at City University and BUSM by various workers. One major requirement was to assess the suitability of the resulting recognition system for identifying shoe upper components as a major component in the automated system. Towards this goal Chapter 2 is a critical survey of the ways in which such a recognition system could be implemented. Chapter 3 surveys printing mechanisms which could be interfaced with the recognition system and would be capable of stitchmarking, and selects the most suitable.

The initial experiments in automatic stitchmarking and the evolution of the eventual system are described in chapter 4. The resulting automatic stitchmarking system is described in chapter 5. The remaining chapters provide an analysis of the performance of various aspects of the system. Chapter 6 assesses the accuracy of the stitchmarking process and chapter 7 the recognition efficiency. A timing analysis is detailed in chapter 8 and the attempts to overcome one potential bottleneck are described in chapter 9. Finally, chapter 10 describes the performance of the printing mechanism and the operation of the machine as a whole.

SI units are used throughout the text. However, Imperial units are included as well for items to which they are commonly applied, such as a toothed belt with 5 teeth per inch, or a printing density of 300 dots per inch.

CHAPTER 2

SHOE COMPONENT RECOGNITION PROCESSES

2.1 Introduction

This chapter presents a brief critical review of methods of identifying shoe components by considering the characteristics of their shapes. It concludes that the methods evolved at City University [3] and BUSM [4] for recognising shoe components are basically sound.

It is necessary to identify each component, and determine its location and orientation, so that the appropriate stitchmarking for that component can be applied in the correct position. Recognition of a component consists of assigning it to a class, where each class represents those shoe components with the same size and shape which will receive the same stitchmarking. This recognition involves an analysis of the shape of the outline of each component under test. The shoe components that the system will have to handle will be pieces of leather or synthetic shoe material, with a uniform thickness of up to about 2.5 mm. They will be fairly opaque, with a closed outline, but may have some holes and open areas within them. Also, they may be floppy or stiff, and although they may be slightly curled or creased, they will need little force to hold them flat.

Commercially available 'ready to run' vision systems of reasonable price are of too general application to be optimal for specific applications. A system tailored to the recognition of shoe components is likely to be faster and have a higher recognition efficiency.

The functions required for a suitable recognition system are shown in fig. 2.1. Firstly the shoe component has to be sensed. The output from the sensor is usually preprocessed by, for example, amplifying, filtering, or digitally converting. Some form of segmentation is usually performed: this being the reduction of the data to that of use for recognition. For example, discarding all except edge points. From the reduced data, features useful for the recognition of the shape are extracted. Recognition is performed by matching the features of the shape under test to those of a knowledge base. Once the shape has been recognised the appropriate response for that shoe component can then be performed.

The system will be treated in two parts, as shown in fig. 2.1. The first is the data acquisition section, and the second is the shape recognition section.

2.2 Data acquisition

In order to identify a shoe component, data about its outline must be acquired in a form suitable for subsequent mathematical analysis. Optical methods of data acquisition are the most appropriate, being highly developed, fast and accurate, though other methods, such as the use of tactile sensors, are conceivable. Since the shoe components are thin, opaque, and are characterised by their outline, they can be treated as two–dimensional shapes for imaging purposes. To simplify the task of acquiring data, the machine will be allowed to deal individually with isolated components, thus removing the complications of imaging and recognising multiple and overlapping components.

Factors which must be taken into account when designing a data acquisition system include the speed, accuracy, complexity, reliability, cost, susceptibility of the system to the environment, and possible 'rough' handling in a shoe factory. Environmental effects which can cause problems include temperature, dust, and local electrical or magnetic fields.

2.2.1 Optical data acquisition

Suitable optical data acquisition techniques are based on two methods,

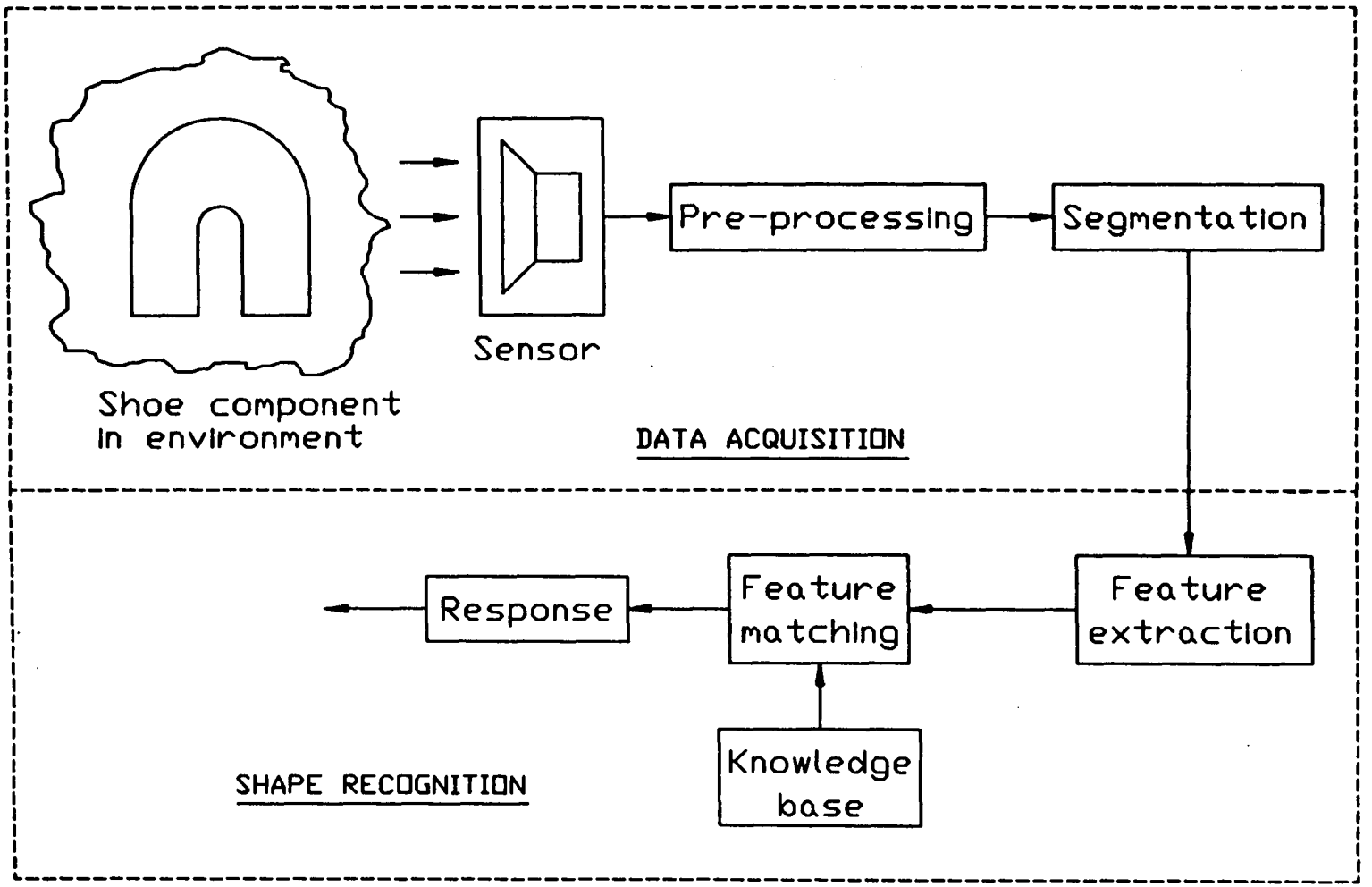


Figure 2.1 *Sub-functions required of the recognition system.*

scanning and imaging. Scanning is the observation of the object under test by focusing attention on one point of the object at a time, where this point is moved in a regular manner over the surface of the object. Optically, scanning can be performed by using (i) a fixed object and a moving sensor, (ii) a moving object and a fixed sensor, (iii) a fixed object, a fixed sensor and a moving point of light, or (iv) a combination of these. The best combination of mechanical simplicity and optical resolution is obtained by using a laser as the light source and scanning it with a rotating polygonal mirror. However, such a system has the disadvantages of a small depth of field, the need to maintain the rotating mirror to a high standard of accuracy, and the high cost of such systems [4].

Data acquisition by imaging requires the use of a device to produce an image of the object under test and a transducer to convert the image to an electrical signal. The imaging device must produce an accurate image with sufficient resolution to give the required accuracy in the position and orientation of the shoe components, and to be able to differentiate between adjacent sizes.

Two types of transducer are available; these are television camera tubes and solid state devices. Television camera tubes comprise a large group of devices which can be divided into two basic types. One type has a light sensitive screen and a means of scanning an electron beam across this screen. The other type has a stage which transfers the electron image from a light sensitive screen onto an intermediate electrode, which is then scanned by an electron beam. Different devices of each type differ in the type of sensitive screen, the type of intermediate electrode (if present), and the use of magnetic or electrostatic scanning for the electron beam. These differences give the devices different operating characteristics. Although they are fairly cheap, television camera tubes suffer from being fragile, having limited tube life, and needing adjustments for drift and aging.

These effects are minimal in solid state imaging devices. Such devices are made using integration techniques to produce arrays of light sensors in a single line or multiple lines on a silicon slice, together with circuitry for connecting each sensor in turn to a single or limited number of outputs. The

light sensitive element can be a photodiode, phototransistor, or a depletion layer formed beneath an electrode. Typically, element spacings of 7 to 100 μm are being produced, in linear arrays of up to 6000 elements, and area arrays up to 400 $\times$ 600 sensors. To produce an image of an area with a linear array requires relative movement between the subject and the array so that successive strips of the subject are imaged on the array. However, this can be an advantage if the objects being imaged have to be moved anyway.

The resolution of a solid state array is determined basically by the sensor spacing, though it is also affected by the wavelength of the illumination. Short wavelengths do not penetrate a sensor very deeply before generating an electron–hole pair, so the charge released is collected in that sensor. However, long wavelengths penetrate deeper into the sensor, and the charge produced may move sideways during its journey to the collecting electrode on the surface of the sensor. This can allow the charge to be collected at an adjacent sensor site, and cause degradation of the resolution. For this reason, solid state camera systems often incorporate infrared stop filters if the illumination contains long wavelengths.

The main advantages of solid state arrays, when compared with television camera tubes, are small size, absence of high voltages, resistance to damage from bright lights, resistance to physical shock, low sensitivity to the effects of magnetic fields, and a fixed and known sensor position.

For recognising the shapes of shoe components and determining their positions and orientations for stitchmarking, the resolution obtainable from the current solid state area arrays is too coarse. Linear arrays provide a higher resolution along one line, and by relative movement past this line can give a higher resolution across an area. Work has shown that digitising shoe components with a 0.2 mm grid gives the required accuracy [3], which over the width of 400 mm needed to image the largest shoe component would require a linear array with about 2000 elements.

A new type of imaging transducer which would have certain advantages comprises a linear array of high intensity light emitting diodes, a precision rod lens array, and a linear array of photo sensors, which are either

charge coupled or amorphous silicon devices. These contact image sensors are designed for use in document image inputting devices, and produce an exact one to one correspondence with a document by relative movement of the document in contact with the device. Devices have been described with 4,864 sensors along a length of 300 mm., to give a pixel spacing of 0.06 mm. The use of one of these devices would remove the need for a lens, with its problems of distortion and alignment, though the length currently available is less than that required for imaging shoe components, and enquiries with the manufacturer disclosed that they are not yet readily available.

2.2.2 Signal conditioning

For analysing the outline of the image of a shoe component the shape has to be abstracted from the background. This is performed most conveniently by processing the analogue output of the imaging transducer using a threshold level. The resultant signal can have parts of the image corresponding to the shoe component at one binary value, and the background at the other. The signal can be converted into a binary form by using a simple, fixed threshold circuit. However, an automatic threshold may be needed if the illumination of the image drifts or there is variation across the scan. It is possible to have a different threshold for each pixel by storing values for each in read only memory, and reading them out through a digital to analog convertor synchronously with the image.

The level of the threshold will affect the apparent size of the image of the shoe component due to partially illuminated pixels at the boundary, as shown in fig. 2.2. If the threshold is set at 50% of the voltage difference between the outputs of wholly illuminated and wholly non–illuminated cells then cells A and D will be registered as illuminated and cells C and E will be registered as non–illuminated. Cell B, which is 50% illuminated, may be registered in either category, depending on noise and non–linearities in the system. If the threshold is moved towards the voltage of non–illuminated cells, then cells like A and D will start to change their category, and likewise with cells like C and E if the threshold is moved the other way. The setting of the threshold hence includes or excludes pixels at the boundary.

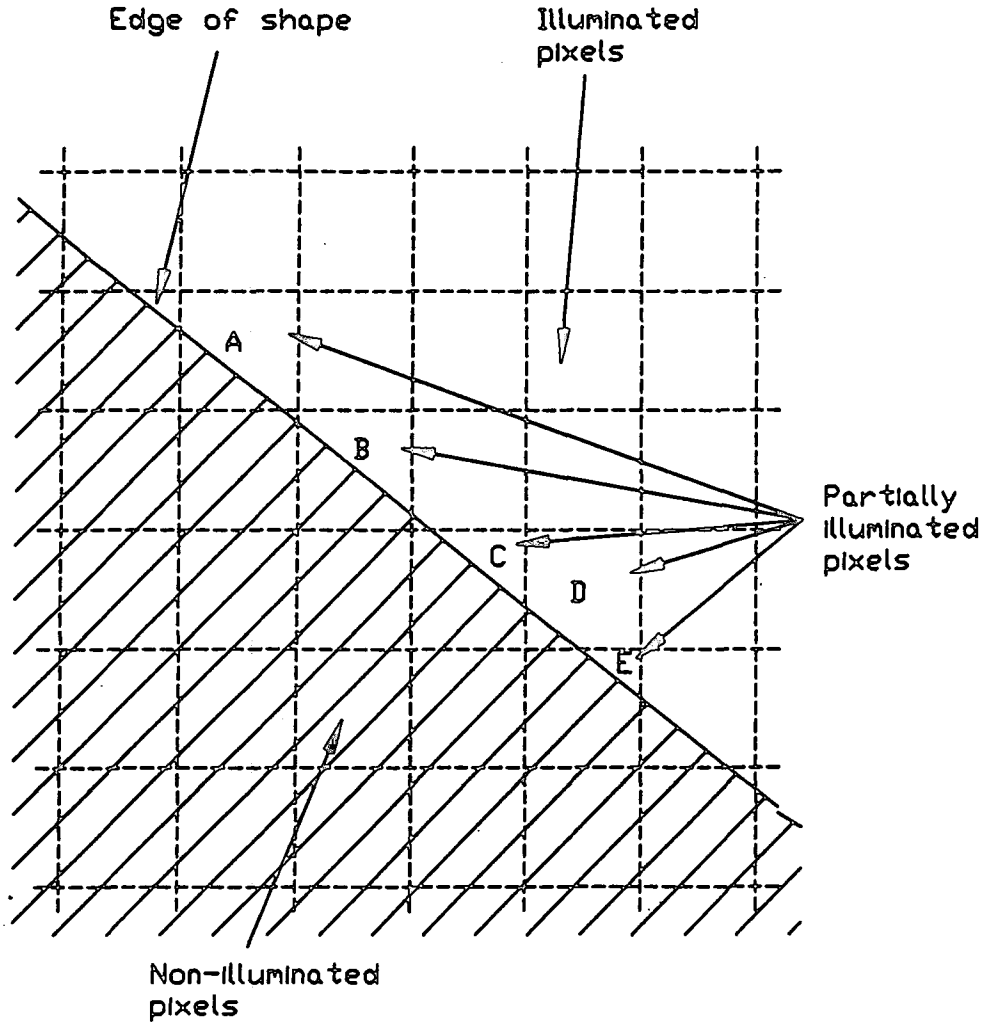


Figure 2.2 *The digitisation of the outputs of pixels at the edge of a shape.*

2.2.3 Lenses

Since, apart from the contact image sensor, these image transducing devices have sizes ranging up to only about 40 mm., an optical system is needed to produce a faithful reduced size image of the component under test. Such an optical system will comprise a source of illumination for the shoe component and a lens to form the image on the transducer.

All lenses suffer from aberrations and lack of fidelity in image production. Compound lenses are designed for particular applications so that the balance of defects is adjusted to maximise the performance of the lens for that application. For example, standard camera lenses are designed for the subject to be at a large distance, and flatness of field is not of high priority, whereas enlarging lenses are designed for operation at small distances with high flatness of field. It is important that a suitable lens is chosen, and is used within its capabilities.

2.2.4 Illumination

The shoe component under test must be illuminated to give an image with the best possible differentiation between the component and its surroundings, so that its outline can be easily detected. The wavelength of the light should ideally be matched to the spectral sensitivity of the transducer. Photodiodes have a spectral sensitivity that covers the visible region, but peaks in the near infrared. This closely matches the spectral emission of incandescent lamps, but these generate a lot of heat and require reflectors and diffusers to produce an even illumination. Rod illuminators are available which are particularly useful when using backlighting with linear sensor arrays. With these a high–intensity tungsten halogen lamp is focused onto the end of a specially designed transparent rod. The light travels along the rod, and is scattered out of the rod's side by a rough strip along its length.

Fluorescent tubes can be used, although the matching between tube emission and solid state arrays is less efficient. Also, so that there is no flicker on successive scans, a high frequency inverter should be used to provide the power supply for fluorescent tubes, rather than the 50 Hz mains.

Since the range of surface colours and finishes of shoe components can range from glossy silver to matt black, it is difficult to choose a background colour to give sufficient contrast to differentiate the component's outline when using front lighting. Methods to give the maximum contrast possible include using a background covered in retroreflective plastic, as used for road signs, combined with axial lighting, or using a background with a fluorescent surface together with ultraviolet illumination. However, there is always high contrast when shoe components are back lit, since they are usually opaque, and even if translucent they scatter such a lot of the light, that only a small portion reaches the sensor through them.

With both front and back illumination consideration has to be given to minimise the effects of dust and dirt from the shoe components which might interfere with the imaging.

2.2.5 Segmentation

Segmentation is the extraction of related data from the image, from which it is easier to calculate features of the shapes within the image. A typical segmentation operation is edge detection which reduces the information to just the outlines of objects. This can be performed by looking for gradients within the image or by convolving a mask.

2.3 Recognition of silhouette shapes

The imaging system will convert the image of a component under test into a set of electrical signals which give a representation of the component digitised on a square grid as in fig. 2.3. The objective now is to use this electrical signal as the basis for classifying the component, i.e. the shape has to be recognised as that of a known component.

2.3.1 Recognition techniques

It is possible to match images directly, but, due to the large amount of information they contain, this takes a long time. In general, recognition techniques attempt to reduce the amount of data that has to be handled,

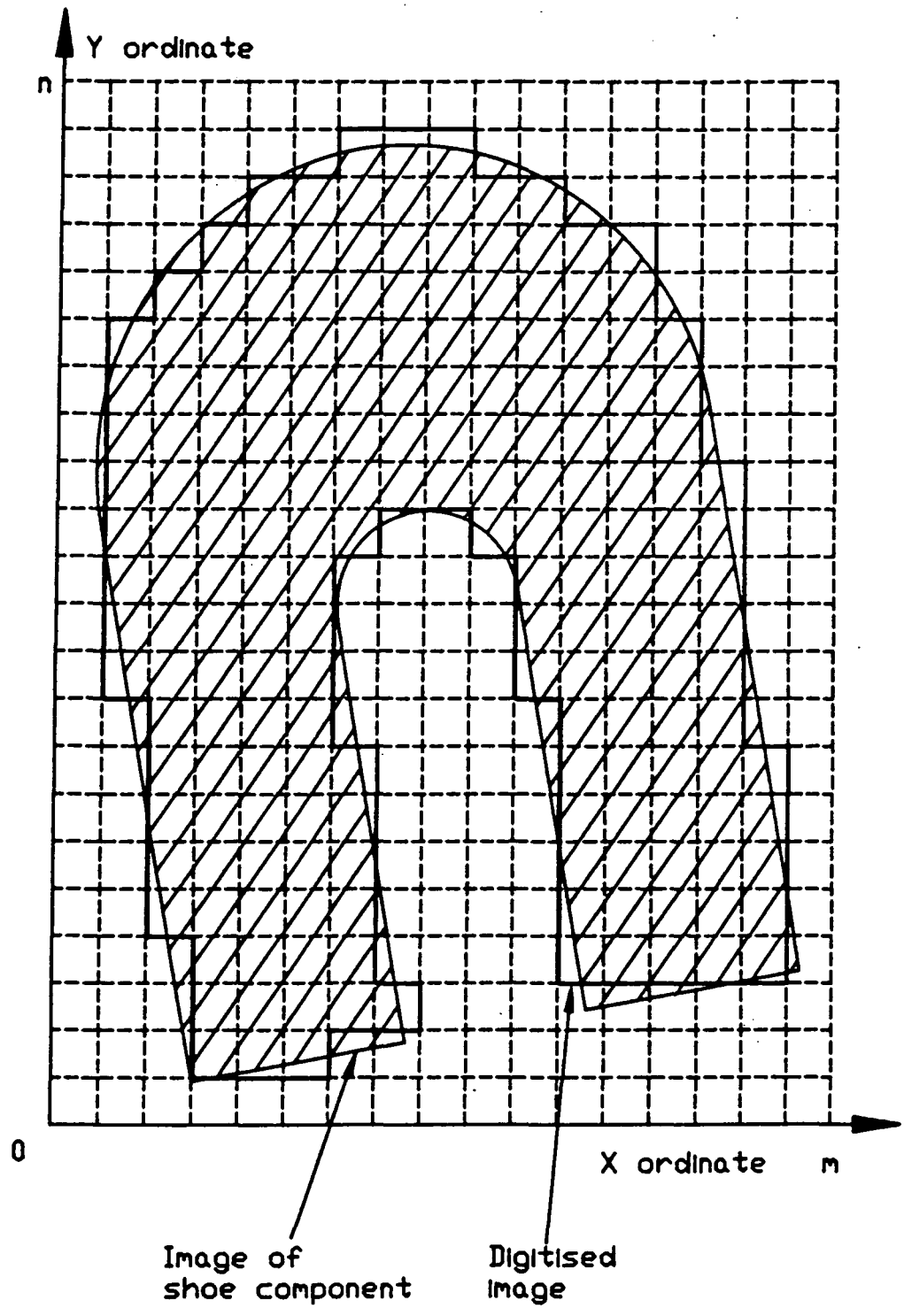


Figure 2.3 *Digitisation of the image of a shoe component on an $m \times n$ pixel grid.*

and convert it to a more convenient form by extracting features of the shape. Some data is discarded, and it will no longer be possible to fully reconstruct the shape from that which remains. However, the data that is retained in these features must be sufficient to distinguish between all the different classes of shapes that the machine is likely to hold at any time in its knowledge base.

Compared with other areas of shape recognition the recognition of shoe components has some simple aspects and some more difficult ones. For example, it is simple compared with the recognition of complex 3–dimensional items which can present multiple viewpoints, since shoe components are 2–dimensional and have simple outlines. Also, the pattern recognition required is essentially deterministic since all of the shoe components belonging to any class will be identical, compared with, for example, the identification of handwritten characters where the same character from differing sources can vary greatly.

However the identification of shoe components does present a problem in the large number of different shapes which will have to be distinguished. At any one time, a shoe factory may have 50 designs of shoe active, i.e. either being produced or likely to be produced. Each design may have 18 half sizes and 20 different components per pair, giving a total of 18,000 different component shapes. Different components may be very similar, differing only in scale with an area difference of 2%, or be mirror images of each other.

Recognition techniques can be grouped according to whether they are external and use only the boundary, or are global and use the interior as well [5] [6]. External scalar methods analyse the essentially one–dimensional function describing the boundary. Internal spatial analysis methods operate on the spatial distribution of the object's interior elements. Internally oriented techniques are preferred since they are less prone to noise in the shape edge. This is important when attempting to recognise shoe components, since the edges may be ragged due to protruding leather fibres.

2.3.2 Feature selection and extraction

The features that are extracted from the shape should be quick to

compute and highly efficient for recognition. They also need to be invariant with the shoe component's position and orientation, and insensitive to noise in its outline caused by a ragged edge. Unfortunately, no general theory exists to allow us to choose which features are relevant to a particular project, so the design of feature extractors is empirical and uses many ad hoc strategies. The more specific the objects to be recognised, the more efficiently can the features be chosen. Shoe components have a wide range of shapes, so the choice of features to use is not an easy one.

Since the edge completely specifies the shape, a method of edge coding could be used, such as Freeman coding [7], which would reduce the amount of information to be handled. However, this still leaves a lot of information to be handled, and it is computationally intensive to match shapes with differing orientations.

Parameters specific to the shapes to be recognised have been used, such as number of corners, holes, lobes, etc., or the distances between such things. These are useful if the range of objects is very narrow, as may be the case in some industrial situations, but shoe components are too varied for them to be of much use. Another technique which has been used, but is not much use in recognising shoe components is the ratio of the sides of a rectangle in which the shape can be enclosed. This of course varies with the ways in which a rectangle can be placed round the shape.

The area within the silhouette of a shape is widely used in recognition systems, since it is a global feature and thus is fairly immune to edge noise. It is also easily approximated by counting the pixels within the silhouette, determined by summing over the image area $m \times n$, as shown in fig. 2.3.

The area, A, is approximated by

$$A = \sum_{1}^{m} \sum_{1}^{n} f(x, y)$$

where $f(x, y)$ takes the value 1 within the silhouette, and 0 outside it.

In contrast to the enclosed area of a silhouette, the perimeter (p), as determined by counting the number of edge points of the silhouette, and

parameters which include it such as the shape factor $(p^2/(4A))$ and peround $(4\pi(A/p^2))$ [8], are unsatisfactory since they are very sensitive to edge noise, and to the orientation of the shape with respect to the digitisation grid. Some correction can be applied for the latter, [9], but this requires a lot of computation and is still not very satisfactory.

The lengths of radii from the centroid to the shape edge can be used reliably if they are long compared with the magnitude of any edge noise. They can be used in polar coding techniques, where a set of radii is measured [10], or in statistical techniques, where for example the maximum, minimum, or RMS mean radii are determined.

Moments of area are further global features which are widely used. The moments of area of a two dimensional shape are defined by

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx.dy$$

For shapes binarily digitised by a square grid, the moments can be approximated by summations of the coordinates of the pixels over the image area. Again the function $f(x,y)$ has the value 1 within the silhouette of the shape, and 0 outside it. The zero order moment $(p = q = 0)$ gives the area of the shape, as already described.

The two first order moments (with respect to the absolute coordinate system) for the $m \times n$ image area are given by

$$M_{10} = \sum_{1}^{m} \sum_{1}^{n} x f(x,y)$$

$$M_{01} = \sum_{1}^{m} \sum_{1}^{n} y f(x,y)$$

The first order moments are used to define the centroid of a shape, since they both have the value zero when calculated relative to the centroid, as further discussed in section 2.3.5.

The three second order moments are approximated by

$$M_{20} = \sum_{1}^{m} \sum_{1}^{n} x^2 f(x,y)$$

$$M_{02} = \sum_{1}^{m}\sum_{1}^{n} y^2 f(x,y)$$

$$M_{11} = \sum_{1}^{m}\sum_{1}^{n} x.y f(x,y)$$

It is more useful if the second moments are translated to be relative to the centroid of the object by using the parallel axis theorem. If the reference system is then rotated around the centroid of the shape, it will be found that the second moments with respect to the reference system will be maximum and minimum at a certain angle. This angle corresponds to the principal axis of the shape, and can be used for specifying the shape's orientation, as outlined in section 2.3.5. The corresponding principal 2nd. moments of area have been widely used as recognition features, [11].

According to Pavlidis [5], although the first few moments convey significant information for simple shapes, they fail to do so for more complicated ones. Also, there is a decrease in performance with higher order moments, since crucial information is lost [12]. It is possible to devise combinations of moments with values which are invariant to rotation [13], which are hence called moments invariant. These have found some use in recognition systems, [14] [15].

Since the boundary completely specifies the shape, it is possible to parametise its contour as a function of time by tracing around it. The resulting function can then be expanded into a Fourier series, and the Fourier coefficients can be used as features [16].

2.3.3 Considerations about features

The set of features selected for recognising a shape is termed a vector of the shape. Since there is no general formula for selecting the features of a vector to give adequate performance, they are usually chosen using empirical methods from a large set of candidate features. Assuming that the large set gives an acceptably small probability of recognition error, then there may be a subset of features which will still give satisfactory performance with the advantage of less computation.

There are several techniques that have been used for choosing the best subset [17]. The optimal feature set is found by examining all possible combinations of the candidate features. However, due to the large number of possible combinations this is usually too expensive computationally.

A suboptimal feature set can be chosen by starting with one feature and adding further features while testing the performance of the set. It is possible to test all of the features singly before hand, and start with those which show the best performance. However, the features chosen are not necessarily independent, and may not form the smallest set giving the required performance, which is why the technique is suboptimal. The opposite approach is to discard features from the large set while testing the performance, and stop when the limit of performance is reached. It is also possible to combine the techniques by, for example, adding two new features and discarding an old one till the required performance is obtained.

It is necessary that the vectors of all the different classes of shapes differ from each other by at least one feature, otherwise it will not be possible to distinguish between them all. That is, there must be separation. From coding theory, the number of features by which the different shapes are separated is termed the minimum distance. The higher the minimum distance, the more immune is the recognition system to errors in the measurement of features. One source of errors is due to electrical noise within the imaging system which affects the measurement of the features. The consequences of this noise can be reduced by using tolerances, so that a shape's vector need only match a class's vector within tolerance.

Rejection occurs if there are errors in one or more features of a shape, such that the vector no longer matches that of the correct, or any other, class. Substitution occurs if the errors cause the vector of a shape to match that of an incorrect class. Substitution is potentially worse, since this may result in shoe components and machinery being damaged, whereas a rejected component can be rescanned. The number of features and the tolerances used affect the rejection and substitution rates [3].

2.3.4 Teaching phase

The vectors for classification are usually obtained by presenting the recognition system with one or more samples of each class during a 'teach' phase. The recognition system extracts the features of this sample and can perform a statistical analysis, if necessary, before storing them in a knowledge base. A statistical analysis is of less importance with shoe components than with many other objects since all shoe components of the same class have nominally identical shapes.

2.3.5 Feature matching

There are three basic techniques for comparing the features of the shape under test with the features of the candidate shapes in the knowledge base.

The first method is to use the binary decision tree technique. This is a serial matching approach, where each feature acts as a filter. The first feature of the scanned shape is compared with the corresponding feature of a candidate shape from the knowledge base. If the features do not match then that candidate shape is discarded, and the next is tried. If they do match, then the process is repeated with further features of both vectors until all candidates have been discarded, and the shape remains unrecognised, or a candidate with a completely matching vector is found, in which case the shape has been recognised.

If the shapes in the knowledge base are arranged in some order, then it is possible that only a certain grouping of them need be candidates for matching. This drastically reduces the computation required for recognition, but there may be problems with the rejection rate of this technique.

The second method is to use the nearest neighbour technique, where the shape under test is allocated the class whose vector is closest in the Euclidean vector space. This is a parallel matching approach, since all of the features are used simultaneously. For large knowledge bases of candidate shapes, however, the computation required for this method is enormous. It is also not appropriate for recognising shoe components since this is largely

a deterministic problem, and in theory an exact match can be found.

The third method also takes a parallel matching approach. It uses a technique based on coding theory [4] [18], where each feature of a vector has equal value and a certain number of mismatches between vectors is permitted for recognition. This makes allowance for the small probability that any feature may have been measured out of tolerance, due to random errors, no matter how large the tolerance is. However, it means that ideally all of the shapes in the knowledge base will be candidates for matching, and these may number many thousands.

2.3.6 Determination of position and orientation

To determine the position of the shoe component a unique point in its shape has to be chosen. The point most widely used for two–dimensional shapes is the centroid [19] [20]. The coordinates of the centroid (X_c and Y_c) of a shape within the digitised area $m \times n$ are approximated by the equations

$$X_c = \frac{1}{A}\sum_{1}^{m}\sum_{1}^{n} x.f(x,y)$$

$$Y_c = \frac{1}{A}\sum_{1}^{m}\sum_{1}^{n} y.f(x,y)$$

$$A = \sum_{1}^{m}\sum_{1}^{n} f(x,y)$$

where A is the enclosed area of the silhouette, and again $f(x,y) = 1$ inside the shape, and 0 outside it.

These calculations involving summations can be speedily performed by digital computers. Due to the influence of the large area within the outline of the shape, the centroid's position is insensitive to a noisy edge.

However, the determination of the orientation of a silhouette is a much more difficult problem, to which there is no solution which is satisfactory generally. For stitchmarking it is not the absolute orientation of the scanned shape which is required, but the orientation relative to the class prototype in the knowledge base. This is because the data for stitchmarking is likely to be stored relative to the details of the class prototype.

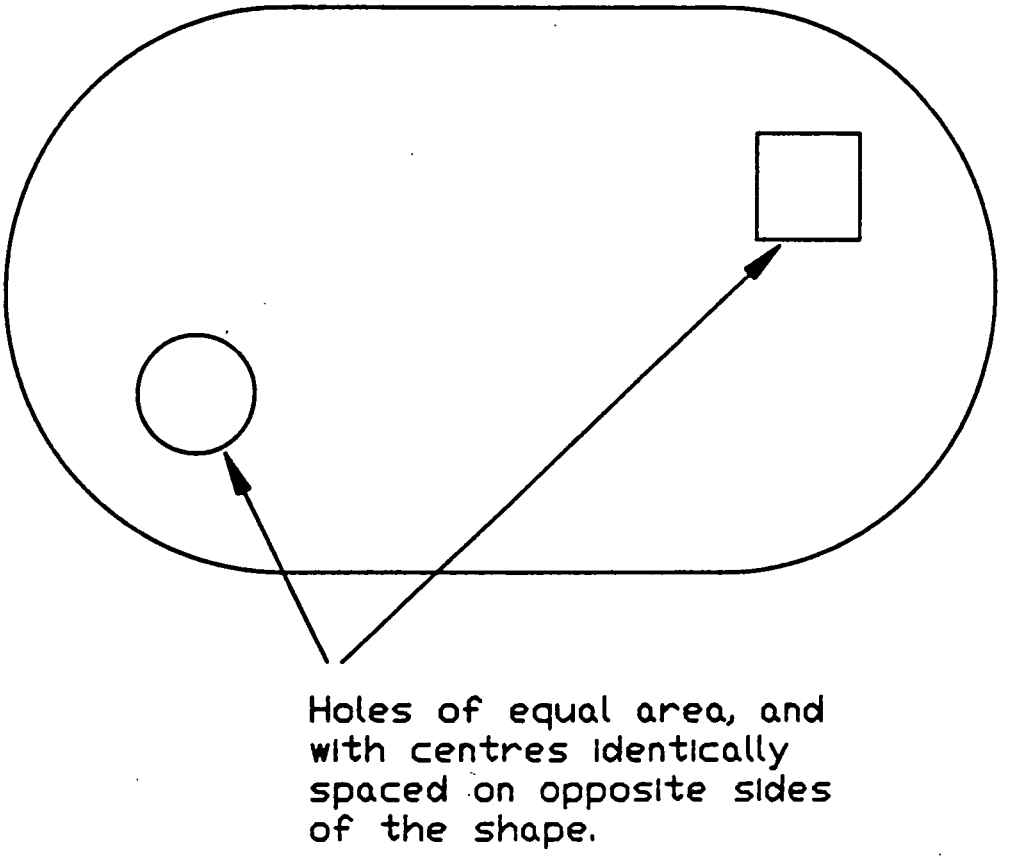


Figure 2.4 *Shape with ambiguous orientation.*

Ideally, the orientation would be specified by a unique axis through the silhouette, which could be easily and accurately determined for all shapes. However, problems arise with shapes which have a point of near symmetry, regular polygons for example, and with shapes like fig. 2.4 [3] which has two holes of equal area, but different shape, at identical but opposite positions.

The starting point for determining a unique axis through the silhouette is usually the centroid, which will already have been found when determining the shape's position. Many methods have then been used, or have been suggested, for specifying a characteristic axis of the shape.

One of the most basic methods is to find the direction of a prominent point of the shape. This works well with some industrial components [22], but there is no prominent point that can be accurately specified for all shoe components, apart from the centroid.

Another widely used method is to find the longest radius from the centroid to the boundary. This was the original approach for determining the orientation of shoe components used by Koulopoulos [3]. However, as he points out, the longest radius may not be unique, as shown in fig. 2.5, and may not be localised at one point on the boundary, as shown in fig. 2.6.

A related approach is to find the direction of minimum or maximum projection length. This method has been used for determining the orientations of objects to be gripped by a robot [22], but, again, for shoe components the minimum or maximum projection length is not necessarily unique and may not be localised at one point on the boundary.

There are several methods based on circles intersecting the boundary of the shape. These determine the coordinates of the intersecting points, the angles between radii which pass through the intersecting points, or the lengths of the radii to the intersection points. The basic method uses circles centred on the shape's centroid. This has been used for determining the orientation of industrial components [23], but would need a way of choosing the circle size to cater for all sizes of shoe components. It would also be prone to noise caused by ragged edges, and would have problems when the edge is almost an arc centred at the centroid, or comes to a corner which just

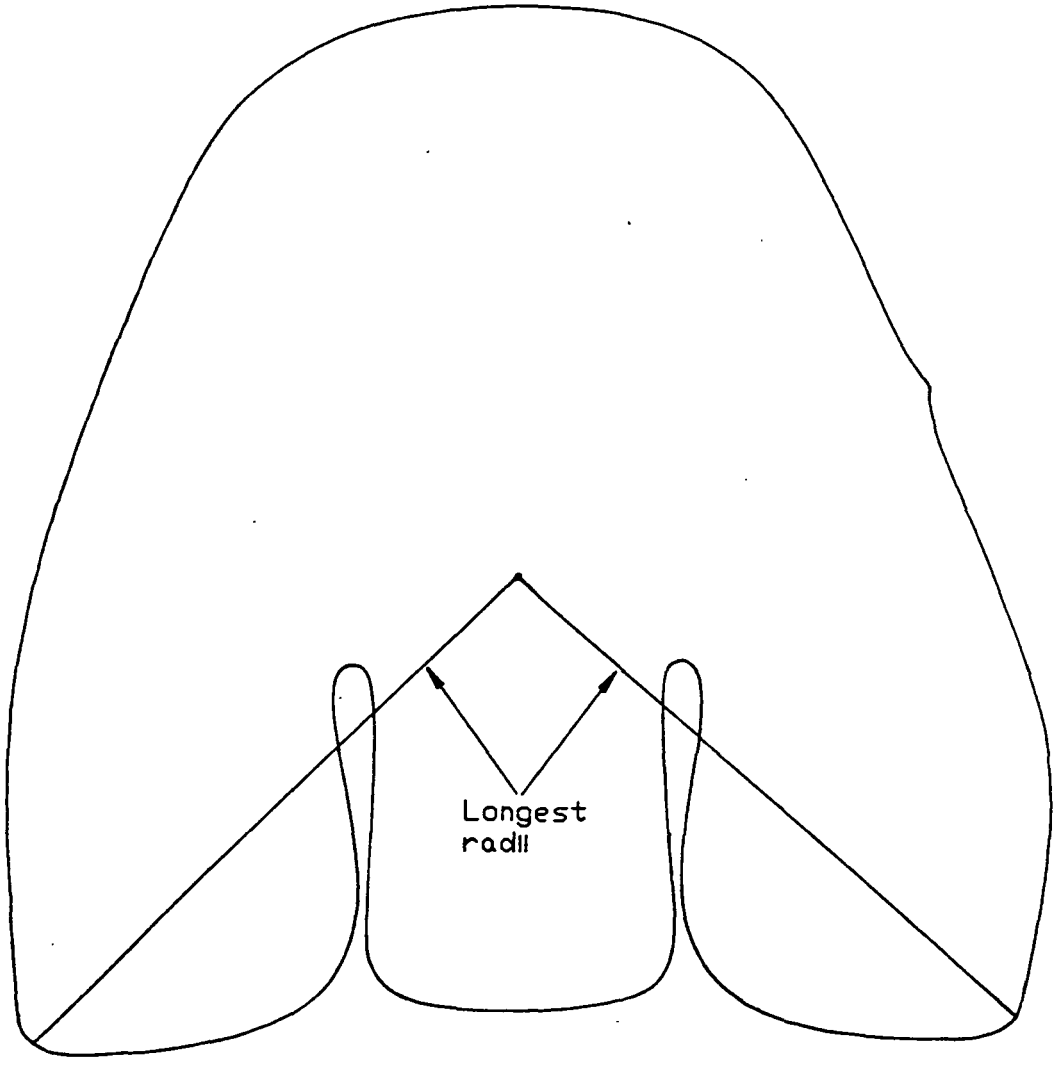


Figure 2.5 *Shape where the longest radius is not unique.*

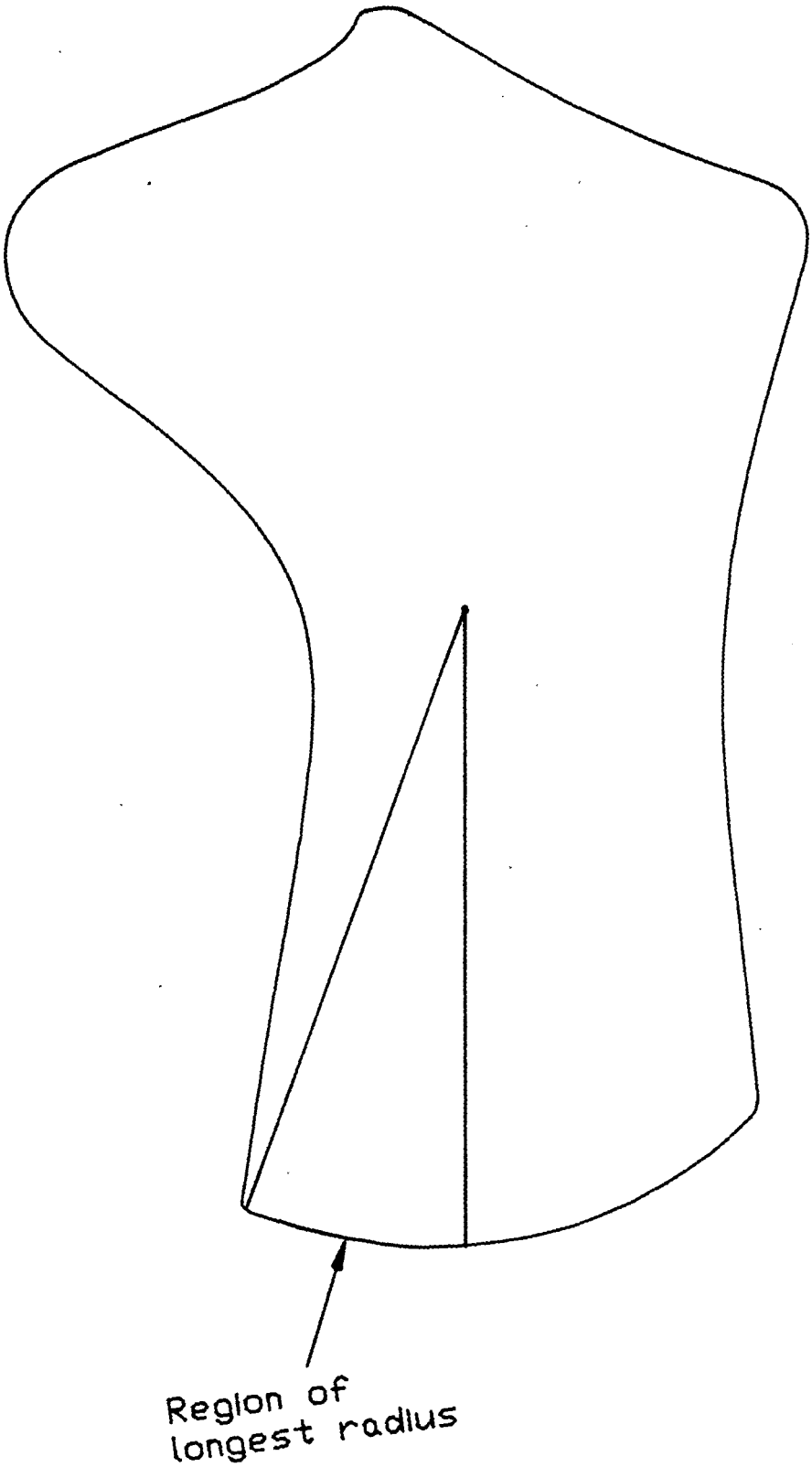


Figure 2.6 *Shape where the longest radius is not localised.*

touches the circle.

Another method uses a family of circles with centres spaced around a circle centred on the shape's centroid. A robotic vision system using this method has been described [24] [25], but again a way is needed of choosing the circle size to cater for all sizes of shoe components. And again, it will be prone to noise caused by ragged edges, and will have problems with some types of edges.

The axis of maximum symmetry has been used for determining the orientation of industrial parts [26] [27], and may be useful with some types of shoe component. However, complications arise with some shapes, such as regular polygons, which may have more than one axis of maximum symmetry.

The corner points of shapes have been used for orienting parts. These can be found using various techniques, such as convolving a mask on the edge points of the silhouette, using the medial axis transform to find the skeleton of a shape [28], and projecting straight lines along the outline. However, corner points do not appear to be too useful since shoe components do not necessarily have well defined corners, and fancily cut ones may have dozens.

A characteristic axis that has been proposed and used successfully for gripping parts by robot [29] [30] [31], is the principal axis of the area enclosed by the silhouette. This is obtained by determining the shape's second moments of area, see section 2.3.2., and provides an accurate method for long thin shapes. However, it does not give the orientation reliably for certain shapes, such as those tending towards regular polygons, where the second moment does not vary much with orientation. One advantage of this method, though, is that the second moments may already have been calculated for use as recognition features.

A related technique is to find the principal axis of the edge points of the silhouette by using the second moments calculated from the edge points only. This method has been investigated for determining the orientation of shoe components [32], but was discarded as not reliable enough. One problem with it is that it is highly influenced by the noise of a ragged edge.

It is possible to find the relative orientation of the shape under test

to the class prototype by cross correlating their edge point positions, but this requires a great deal of computation. The computation can be reduced by using polar coding techniques to give a set of radii for the component under test, and performing a circular correlation technique with the radii of the class prototype [10] [26]. Ideal shapes would have a signature with random–like properties, so that cross correlation would produce a pulse at the orientation where the two shapes coincide. Nearly circular shapes will have very shallow peaks, and shapes with axes of symmetry will have more than one peak. There will again be difficulties with the shape shown in fig. 2.4 if the polar coding of the boundary ignores the internal structure of the shape. Aside from these difficulties, this technique will be able to deal with many shapes for which the principal axis is not suitable.

2.4 Examples of recognition systems

There are recognition systems in the literature which could be used for the basis of a shoe component recognition system.

De Coulon and Kammenos [10] used a TV camera for viewing 2 dimensional industrial objects prior to gripping them. Edge detection was by a 3×3 window and the position was determined using the centroid. The relative orientation of the objects relative to the class prototype was determined using a cross correlation technique.

Armbruster et al [24] used a solid state area array, with hardware for producing the sums required to determine the centroid coordinates. The features calculated were area and coordinates of the boundary obtained by a polar coding technique using a family of circles. The system was used for examining 3–D industrial components, the features of the stable positions of which were stored in the knowledge base.

Zurcher [27] determined the position and shapes of up to 10 silhouette objects a second passing on a conveyor belt for industrial automation applications. A TV camera provided a 256×256 array of pixels, but he noted that although it was cheap, its principal defects were geometrical distortion and thermal drift. The system calculated the position of the centroid and

used polar coding to determine the orientation by finding the axis of greatest symmetry. The set of radii from polar coding was used for matching the shape under test with reference sets for recognition. All of this was performed by hardware, using TTL LSI.

Vuylsteke et al [31] describe a vision system for flat objects which was based on a video camera, where the area and the first and second moments were calculated very quickly by hardware.

A system has been described [33] which gripped mechanical parts on a conveyor belt moving at 300 mm/sec. This used a television camera with the belt illuminated by a high speed flash so that the image was not blurred. The parts to be gripped had not to overlap, and they were detected by searching for blobs.

Cronshaw [34] describes a conveyor based system for recognising the type, position, and orientation of chocolates, and directing a robot to decorate them appropriately. The recognition system was based on a 256 × 256 pixel digitised TV picture of the silhouette plan of each chocolate. The chocolates were decorated downstream of the scanning stage. Features extracted included area, perimeter, centre of area, radii signature, and corners. It handled a small number of shapes, with a recognition time of about 0.25 secs.

A system demonstrated by the French 'Centre Technique Cuir' for recognising and stacking shoe components apparently uses a TV camera and a conveyor belt, though further details have not been disclosed, [35].

2.5 Conclusions

To obtain a stable, high resolution image, requires the use of a solid state linescan device, and consequently the use of a lens for imaging. The relative movement which is required between the shoe components and the imaging system can be obtained by carrying the components on some sort of transportation device, such as a conveyor belt or a drum. Backlighting of the components will produce the highest contrast image.

For recognition, global features will be the more reliable since the component edges may be noisy with leather fibres. Area and moments are

the major features that should be used. Polar coding, using radii from the centroid to the edge, is more prone to edge noise, but should prove useful if used appropriately.

The centroid of the shape will determine the component's position, and the principal axis can be used to quickly determine the orientation of long thin shapes. It may be necessary to use some other method, such as a correlation technique, to decide the orientation of those shapes whose principal axis cannot be accurately determined.

These features most appropriate for matching and the parameters for determining the position and orientation of shoe components are basically the same as those which evolved on the recognition system at City University and BUSM, which was to be the basis of the stitchmarking machine. This confirmed that, initially, the recognition system ought to be used in the form that it arrived from BUSM. It will be described more fully, together with the modifications required for stitchmarking, in chapter 5, and assessments of various aspects of its performance will be discussed in chapters 6, 7, and 8.

CHAPTER 3
PRINTING TECHNOLOGIES FOR STITCHMARKING

3.1 Introduction

This chapter examines those printing technologies which are suitable for stitchmarking shoe components and could be controlled by a computer, and gives reasons for the decisions made in eventually choosing an electrographic printer for the automatic stitchmarking machine.

The established methods of stitchmarking are manual, making use of either hand operated letterpress type stencils, an example of which is shown in plate. 3.1 [36], or by drawing through slots in card templates placed over the components. Neither of these is suitable for computer control. A marking technology is required which can be controlled by a computer so that components arriving at random can be marked appropriately after being recognised. The necessary speed of operation rules out any processes requiring mechanical changes to the printing mechanism in order to print different shoe components. Also the marking has to be visible on all types of shoe upper material, has to remain in place during the manufacture of the upper, and has to be removable without damaging the surface when the shoe is completed.

The printing mechanisms which can be controlled by computer fall into one of two groups, small area markers and full–width markers. The first includes those which can only mark a single point, or a few points simultaneously, so lines would have to be drawn by sequentially tracing the device across the shoe component. The second group can print a long line

This bench mounted machine marks uppers with guide lines for fancy stitching, perforating, positioning underlays, etc.

Types of work
All upper components, using water soluble or ribbon inking ('to order').

Output
Up to 300 pieces per hour.

Operation
The upper component is located in a guide board on the base of the machine. The spring-balanced die holder returns after marking, and contacts an ink pad which re-inks the die.
The machine can be used with either treadle or hand operation.

Benefits
1 The die holder can take dies up to 31cm x 23cm (12¼" x 9").
2 The machine can be fitted for either water soluble or ribbon inking ('to order').

Plate 3.1 *Example of a hand-operated stitchmarking machine* [36].

of points simultaneously, so that by using relative motion between the shoe component and the print line the whole component could be printed in one movement.

The full–width printers are the more promising due to their shorter time to print each component and the lack of any complicated scanning mechanism. However, they are generally more expensive, so both groups of printing mechanisms will be examined.

3.2 Small–area markers

Small–area markers use a device that can only mark at one point, or a small number of points, at any moment. Therefore, to draw lines, the device either has to be scanned raster fashion over the component, switching on when it is over a spot to be marked, or it has to be used as an X–Y plotter, and trace along the stitchmark lines. In both cases the relative motion can be obtained by moving the marking device or the component being marked.

The devices which fall within this category include pens, dot matrix impact print heads, and ink jet devices. The simplest of these is the pen. This could be a ball point or fibre pen, especially one of the high quality, long life variety used in X–Y plotters. They have the advantage of being cheap, and using simple drive mechanisms, but the pens dry up, wear out, pick up dirt, and may get very hot if used intensively at high speed. One other disadvantage is that since they work by contact, some method would have to be used to hold the component securely while the pen moved over its surface. A demonstration of stitchmarking using a ball point pen had been shown at BUSM. Here the pen moved backwards and forwards in the X direction, while the shoe component was held down by idler rollers onto a table which moved under the pen in the Y direction. However, this was too slow to meet the required specification.

Dot matrix print heads consist of linear arrays of 7 to 30 fine wires which can individually be given an impulse against an inked ribbon suspended close to the surface to be printed. It would be possible to use a dot matrix print head to print onto the shoe upper component, though the print head

would have to be moved in a raster pattern over the component, and the component would have to be held firmly during printing. A non–permanent ink would also have to be used.

Moving to non–contact printing devices, it would be possible to use a fine airbrush in a mechanism which X–Y plots the lines. Disadvantages with this method are that the ink has to be dried, the pen requires cleaning, and does not draw a fine and distinct line.

A related process that can draw much finer lines uses ink jets. These devices issue a stream of ink droplets which is directed against the surface to be printed. The position of impact of the droplets is controlled either mechanically, by moving the jet, or by deflecting the droplets electrostatically.

Attempts to use a deflected ink jet for recording date back to at least Kelvin in 1873, but printers using this technique have only become available since the mid 1960's [37]. These devices have just one ink jet nozzle from which the ink droplets emerge with a charge obtained from a high voltage electrode, as shown in fig. 3.1. Deflecting electrodes alter the trajectory of the droplets on their way to the surface to be printed, so they can be targeted over a small area. These print heads are usually continuously cycling devices, where droplets are issued continuously, but those not needed are collected for recycling before reaching the surface. However, they may also be 'drop on demand' (DOD) types, where ink droplets are only ejected when required. These devices are typically used for printing 'sell by' dates and batch numbers on items passing on conveyor lines.

Problems were foreseen with the ink of continuous cycling devices becoming contaminated with the dirt and debris of shoe factories, and causing blocked jets. Devices which make use of the deflection and collection method of operation cannot be used in situations where they are subject to accelerations, so must be kept stationary while the surface being printed moves perpendicularly to the direction of deflection. There may also be inaccuracies in droplet positioning due to non–linearities associated with the deflection of the ink droplets.

Non–deflection ink jet mechanisms are of the 'drop on demand' (DOD)

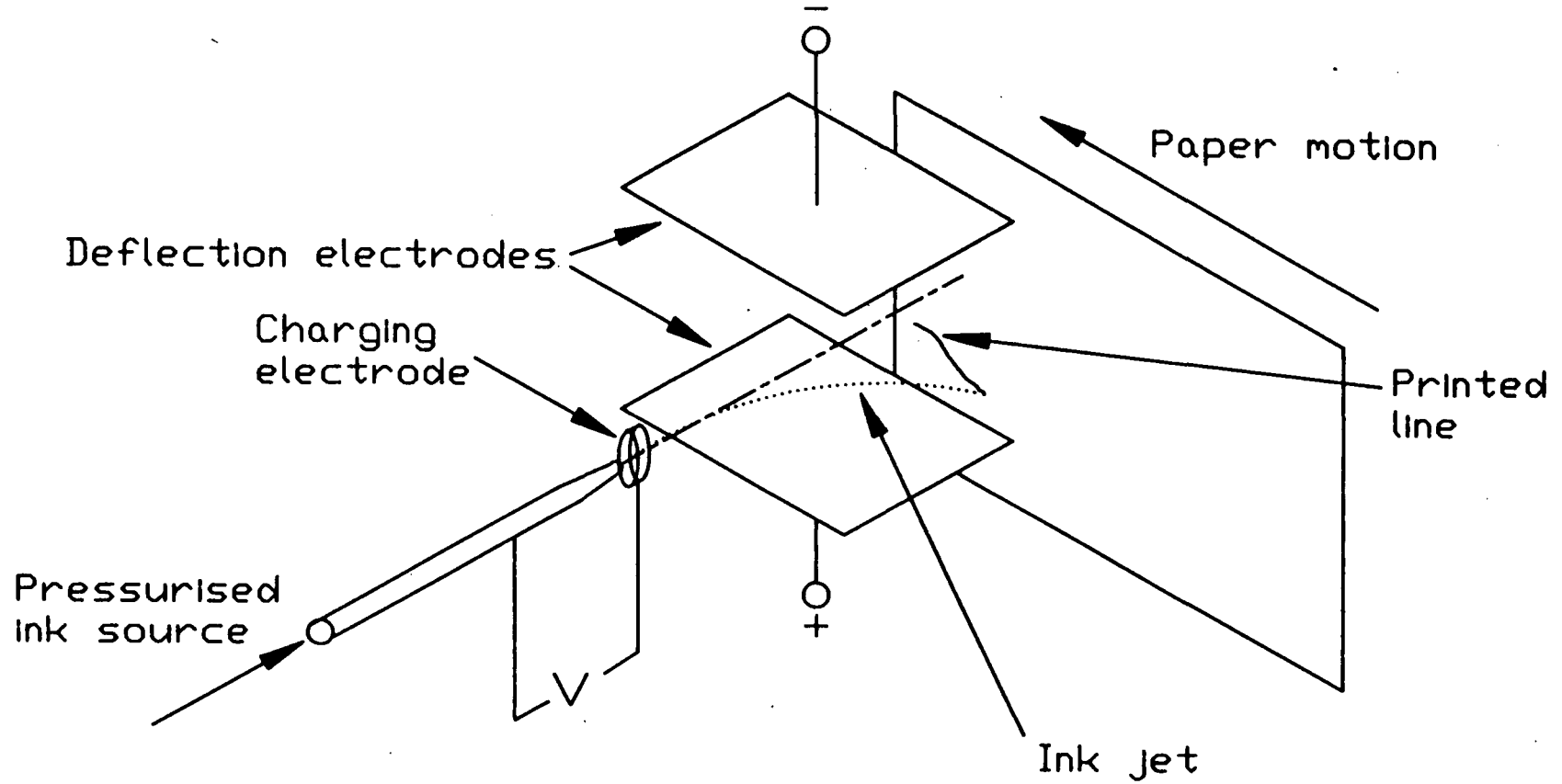


Figure 3.1 *Schematic diagram of a deflection ink jet mechanism.*

type, and may use electrodes to gate or modulate the stream of droplets, but must use relative motion to produce printed images. Some print heads have only a single nozzle and must be moved over the image to be printed. Others, such as dot matrix print heads, have a line of up to about a dozen closely spaced parallel gated jets, each one of which can print dots straight ahead. Such a head could be moved in a raster pattern over the component to be printed, printing dots when necessary, much as happens when printing characters or graphics on paper. This method has been demonstrated for printing patterns onto textiles [38]. These non–deflecting heads are inherently more accurate than the deflecting type since there are none of the linearity problems associated with the deflection systems.

A new process which is related to the DOD ink jet printer is the wax jet mechanism, where the ink is in the form of a coloured wax [39]. When a dot is required to be printed high intensity heat causes a small blob of the wax to be melted and ejected from a nozzle onto the receiving paper, where it solidifies. This overcomes the problem of the drying time of a liquid ink.

From this discussion it can be seen that the main drawback of using single–point or small–area printers is that either the printing device or the shoe component, or both, have to be physically moved around. This is a slow process compared with the desired eventual cycle time of the machine, section 1.5., whether a raster or X–Y plotting technique is used. The time taken to print a component raster fashion depends on the area over which the print head can produce printing at any moment in time. It would also depend on the maximum speed and acceleration that can be given to the printing device or the shoe component, and the length of the component along which printing is required. The time taken to X–Y plot the stitchmarking on a shoe component depends on the length of the lines to be drawn, and again on the maximum speed and acceleration that can be given to the printing device or the shoe component.

Although it was conceivable that with careful design of the movement mechanism a single point device could have been used to meet the specification, it appeared very difficult to increase the speed to meet any future development requiring the printing of several components a second.

The main advantage that they do have is that there is a wide range devices available with a variety of inks and colours.

3.3 Full–width marking

This section deals with those devices which can mark simultaneously in many places along a line. If the component to be stitchmarked were moved perpendicularly past the line, then all the stitch marking could be applied in one pass.

3.3.1 Ink jet and thermal print mechanisms

The ink jet devices which fall into this category can again be divided into those which make use of deflection of the ink jets, and those which do not [40].

It is possible to link together in a line, as shown in fig. 3.2, several of the single point deflection type ink jet heads already discussed. Each head would print across twenty to thirty millimetres of the width, so that the maximum deflection of one jet butted up to the minimum deflection of its neighbour. However the heads would have to be assembled and calibrated accurately, and each would need its own drive electronics and ink supply, so increasing the cost.

The other type of ink jet array is distinguished by having a nozzle for every printing point across the width to be printed. A simple method of implementing this technique, although too coarse for stitchmarking, is the Millitron process, which has been used for several years for printing patterns on carpets and furnishing fabrics [41]. Dye is squirted continuously, to prevent clogging, through a line of needles arranged at a density of 4 per cm to a catch basin. Each needle has a computer controlled air jet, which can deflect the dye onto the fabric being printed.

In another type of mechanism, used for document printing, the ink is forced through a linear array of fine nozzles, where it is broken up into droplets by using ultrasonic energy. Each nozzle has its own charging electrode, and droplets not needed for printing are deflected into a gutter for

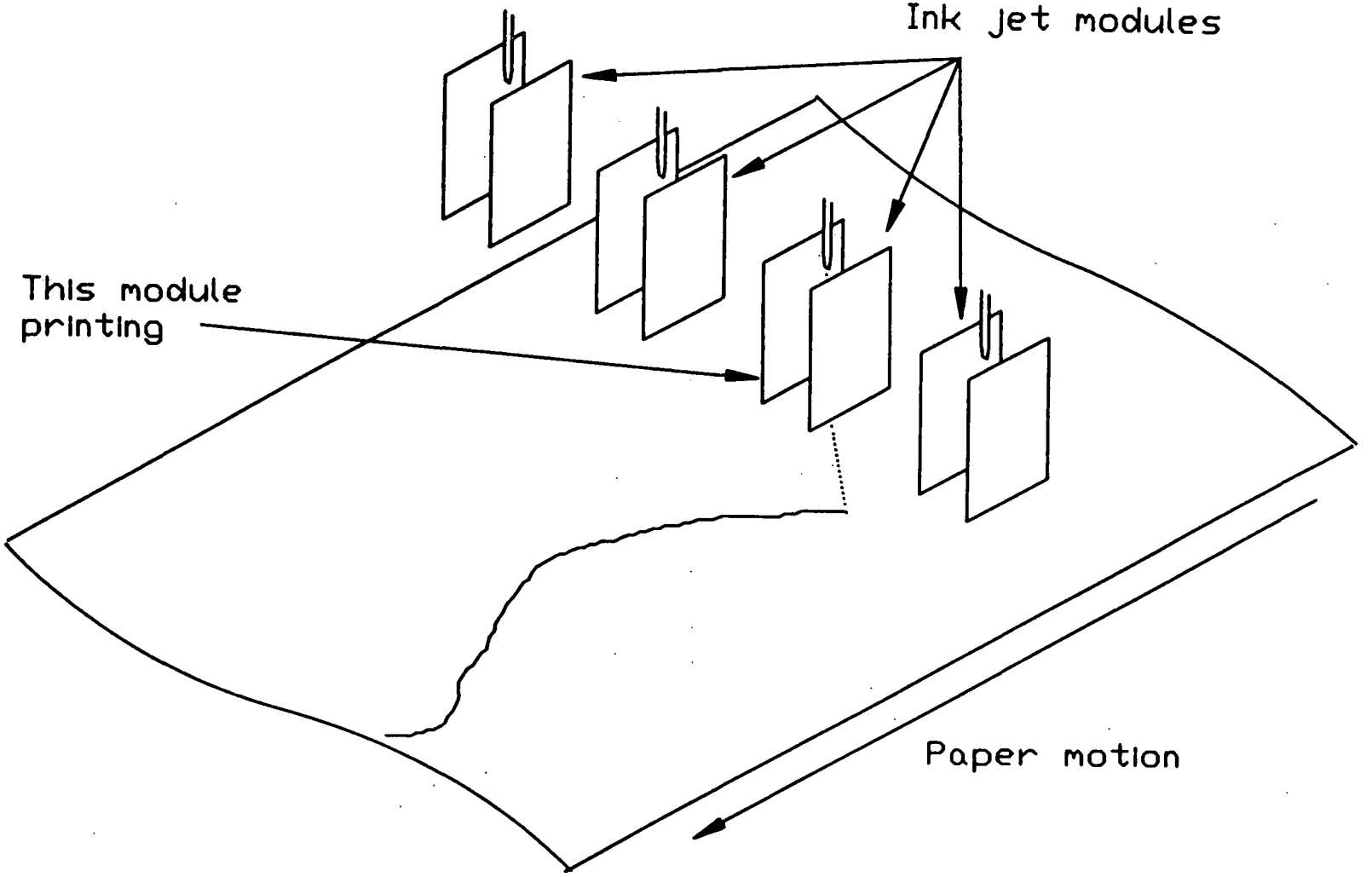


Figure 3.2 *Schematic diagram of a linear array of deflection ink jets.*

recycling, while those required are allowed to pass straight ahead onto the surface to be printed. The accuracy of droplet placement is mainly determined by the alignment of the individual undeflected jets. Diconix Inc. have a commercially available printer using this technique. Their Dijit 2800C system has a 270 mm wide print head with a density of 4.7 jets per mm (120 jets per inch), which is capable of printing paper passing at up to 4 m/sec. This looked very attractive, however the price quoted for the print head alone was 'about £10,000' which severely reduced its rating for stitchmarking.

A variation on this type of print mechanism is the bubble jet printer. Within the print head of this mechanism the ink is heated locally within each ink capillary. This causes rapid vaporisation of the ink, formation of a bubble, followed by collapse of the bubble and the issuing of an ink droplet. A prototype high speed printer with a linear array of 2000 nozzles has been reported, but is not yet commercially available [42].

It can be appreciated that although full–width ink jet printers are very fast and could meet the speed and probably the accuracy requirements for stitchmarking, there is only a small range of equipment available and this is expensive. Also it was predicted that ink jet printers with their multiplicity of fine jets would be very susceptible to the dust and dirt of shoe factories.

Another type of full–width printing mechanism which is commercially available is the thermal printer. These have a bar of resistance elements in contact with a waxed carrier which touches the surface to be marked. The image is produced by melting dots of wax which are transferred from the carrier to the surface being printed. Typical mechanisms can print up to 16 dots/mm, and up to 500 dot lines per second [43]. A limitation of this technique for stitchmarking is that it is a contact technique so the shoe component would have to be held firmly whilst it rubbed against the waxed carrier. Also the printing speed is limited by the time constants dictated by the thermal mass of the heating elements [44].

3.3.2 Electrographic print mechanisms

A form of high speed full–width printing which has advanced greatly

in recent years and looked very promising as a suitable mechanism for stitchmarking is electrographic printing [45] [46] [47] [48] [49] [50]. Electrographic printing can be defined as any process in which a latent electrostatic charge image is formed on the surface of a dielectric material and subsequently made visible by a toning process.

The simplest implementation of electrographic printing uses charging electrodes in contact with or very close to the dielectric surface, with a conducting ground plane behind [45]. When a potential difference is applied between the electrodes and the ground plane an electric charge having the same shape as the cross section of the electrode is deposited on the surface of the dielectric. This image can be made visible by a developing system, where pigmented toner particles are introduced into the vicinity of the charged areas, to which they become attached. The image is usually made permanent by heat fusing the toner to the surface. The process is very simple, and has the capability of high printing speeds, but attempts to produce a viable electrographic printing process in the first decades of this century were not commercially successful.

One difficult problem is that the dielectric medium used to receive the charge image must have a time constant sufficiently long so that the charge does not dissipate before it can receive the toner. The resistivity of common untreated paper it too low to support a charge for more than a few microseconds [45], and hence it has to be given a coating of a dielectric material for use with such a process. This paper is used with electrostatic plotters, which, for example, can plot 1106 mm wide with a dot density of 15.7 per mm (400 dots/inch) at a speed of 10.5 mm/sec [51]. In the case of stitchmarking, leather also has a resistivity similar to that of plain paper, so could not be used directly to retain a charge image for long enough for toning.

This problem, and others which had prevented a viable electrographic printing process on plain paper, was only overcome with the efforts of Chester Carlson in the late 1930s and 1940s on the Xerographic process which proved commercially successful. The solution is to use an offset process employing a dielectrically coated drum on which the charge image is formed and toned.

The toned image is then transferred from the drum onto ordinary paper, and the drum is reused. These processes are extensively described by Schaffert [49].

In the Xerographic process the dielectric coating is photoresistive, and light causes portions of an all over charge on the drum to be discharged. The charge image remaining where the light has not reached receives the toner. In the original process, still used extensively in office copiers, the light image is produced on the drum by a lens from original copy. This type of printer has been used for printing patterns on fabric [41], but the need to produce a complete original for imaging onto the drum makes photocopiers unsuitable for printing stitchmarking. However, since 1975, when IBM Corporation announced the first machine [52] [53], laser printers and their derivatives have become available which can print images generated by computers.

Laser printers have recently become very popular as high quality computer hard copy devices, and there has been a large reduction in prices at the low specification end of the market. In addition to laser printers, LED (Light Emitting Diode) printers and LCS (Liquid Crystal Shutter) printers must also be included, since they only differ from laser printers in the light source and its modulation.

All members of the laser printer family use a process based on Xerography with an imaging drum coated with a photoconductor. An image of electrical charge is produced by writing on the completely charged drum with a scanning laser beam, a linear array of LEDs, or by modulating a light source with a liquid crystal array. The photoconductor resistivity decreases where the light strikes and most of the charge is discharged to the earthed metal underneath. The resistivity of the photoconductive surface which remains in the dark gives a long enough time constant for the charge to remain while the image passes through a developing station, where toner is applied. The toner is a very fine, coloured, thermoplastic powder, and is usually suitably charged so that it adheres to the drum in those areas which have been discharged. A corona discharge behind the substrate being printed is used to transfer the toner from the drum to the substrate as it passes under the drum. The toner powder is then permanently fused onto the paper using

heated rollers, radiant heaters, pressure, or a combination of these.

Laser printing can be very fast, for example the Siemens 2300–2 laser printer will print up to 345 mm wide at a paper speed of 0.74 m/sec, however it is very expensive. Cheap laser print engines are available for slow, low–throughput applications, printing at 11.8 dots/mm (300 dots/inch) across a 210 mm width. However, these print at speeds as low as 0.02 m/sec and are designed for light duty work of a few thousand prints per month. There is much activity in the area of office laser printers, though there are few manufacturers of the print engines. Developments in the area of high speed, wide printers are, however, much slower.

A laser printer could be used to print a computer generated stitchmark pattern, though the major concern about using laser printers for stitchmarking in a shoe factory environment was the uncertainty about the working life of the photoconductive coatings. These are based either on selenium or organic resins, which are fairly weak physically, so may wear quickly when used with shoe materials and could be damaged by staples or tacks which might pass through. They may also be prone to contamination by compounds in shoe components, such as plasticisers in PVC, and would have to be cleaned carefully if contaminated with adhesive. The LED and LCS printers, with their fixed light source positions, are potentially more accurate than the true laser printer, where the positioning accuracy depends on the accuracy of scanning by a rotating prism. This is illustrated by figs. 3.3 & 3.4, which show the printing misalignment exhibited by the Dataproducts 2600 A3 laser printer [54]. Fig. 3.3 shows the distortion to a horizontally scanned line due to misalignment of the mechanically scanned laser with the print drum, and fig. 3.4 shows misalignment between the same printed dot on successive scan lines caused by jitter in the printing synchronisation.

For stitchmarking, these electrographic processes have the advantages that they are very compatible with digital electronic systems, they can print images onto surfaces which are moving past at steady speeds, and they do not have the fine nozzles of ink jet printers to become blocked. However, problems in using electrographic printers for stitchmarking include the small range of toner colours which are available and the need to fuse the toner

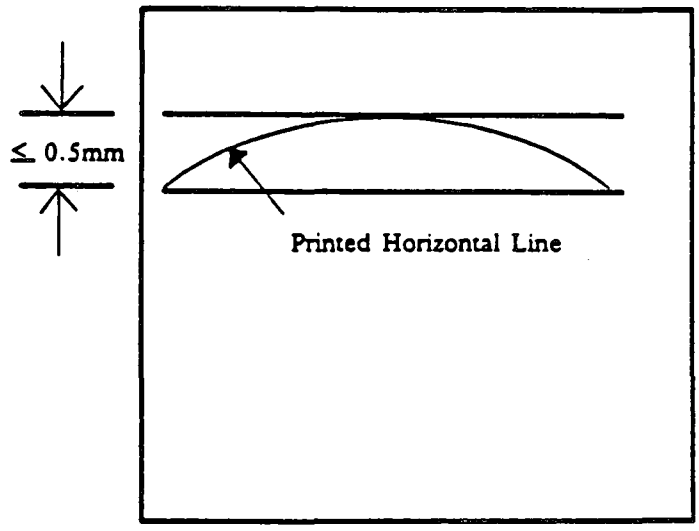


Figure 3.3 *Horizontal scan skew of the Dataproducts 2600 laser print engine* [54].

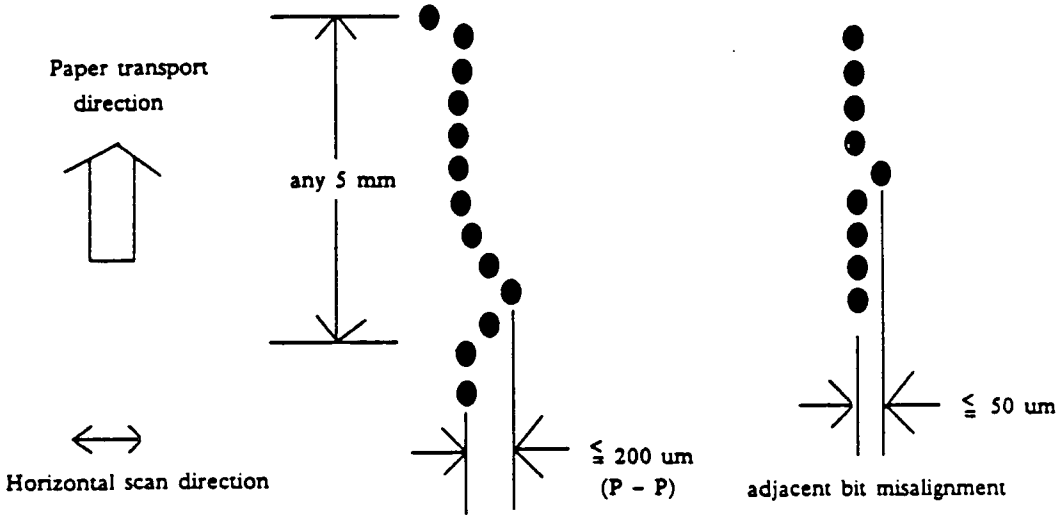


Figure 3.4 *Image-phase misalignment of the Dataproducts 2600 laser print engine* [54].

to the printed surface. Toners are available in black, and often the primary colours, and although special colours can be produced the electrical and/or magnetic specifications of the toners are very strict, so this is not a facile thing. The toner is usually fused permanently by heated rollers or radiant heaters, though pressure is sometimes used in addition or as an alternative. For stitchmarking, the toner must not be permanently fused, so the fusing system would have to be adjusted.

An alternative to the laser printer is the magnetographic printer. These produce a permanent magnetic image within the surface of a steel drum by using a bank of recording heads, where each head is just over 0.1 mm thick [55]. The toner has magnetic properties and adheres to the magnetic image on the drum. It is transferred to the paper by using a transfer corona, and is fused by heat in the normal way. This process has the advantage that it is very simple and the image drum is very robust. The major supplier of these printers, Bull Peripherals, of France, produces print engines to print up to 345 mm wide at a paper speed of up to 0.3 m/sec. However, the high price of these engines can be appreciated from the cost of the magnetic recording head modules which has been quoted at 'approximately £4000 per 100 mm of printing width'. This reduced the attractiveness of the magnetographic process for stitchmarking.

A type of electrographic printer which has the advantages of robustness and high speed is the ionographic printer, which has been developed by Dennison Manufacturing Company in conjunction with Delphax Systems, of Canada and the USA respectively. This high speed printing process, which is described by Rumsey & Bennewitz [56] and depicted in fig. 3.5, again employs an offset method of electrography, but this time with an aluminium drum which has a very hard anodised dielectric coating, of about 40μm thick. An image in electrical charge is formed directly on the surface of the anodised layer by a closely spaced ion–cartridge which generates modulated jets of negatively charged ions from a multitude of fine holes along its length. The image is toned using a now almost ubiquitous magnetic brush development unit, which in this case uses a conductive, single component, magnetic toner. The use of this toner has two advantages. Firstly it requires only a simple,

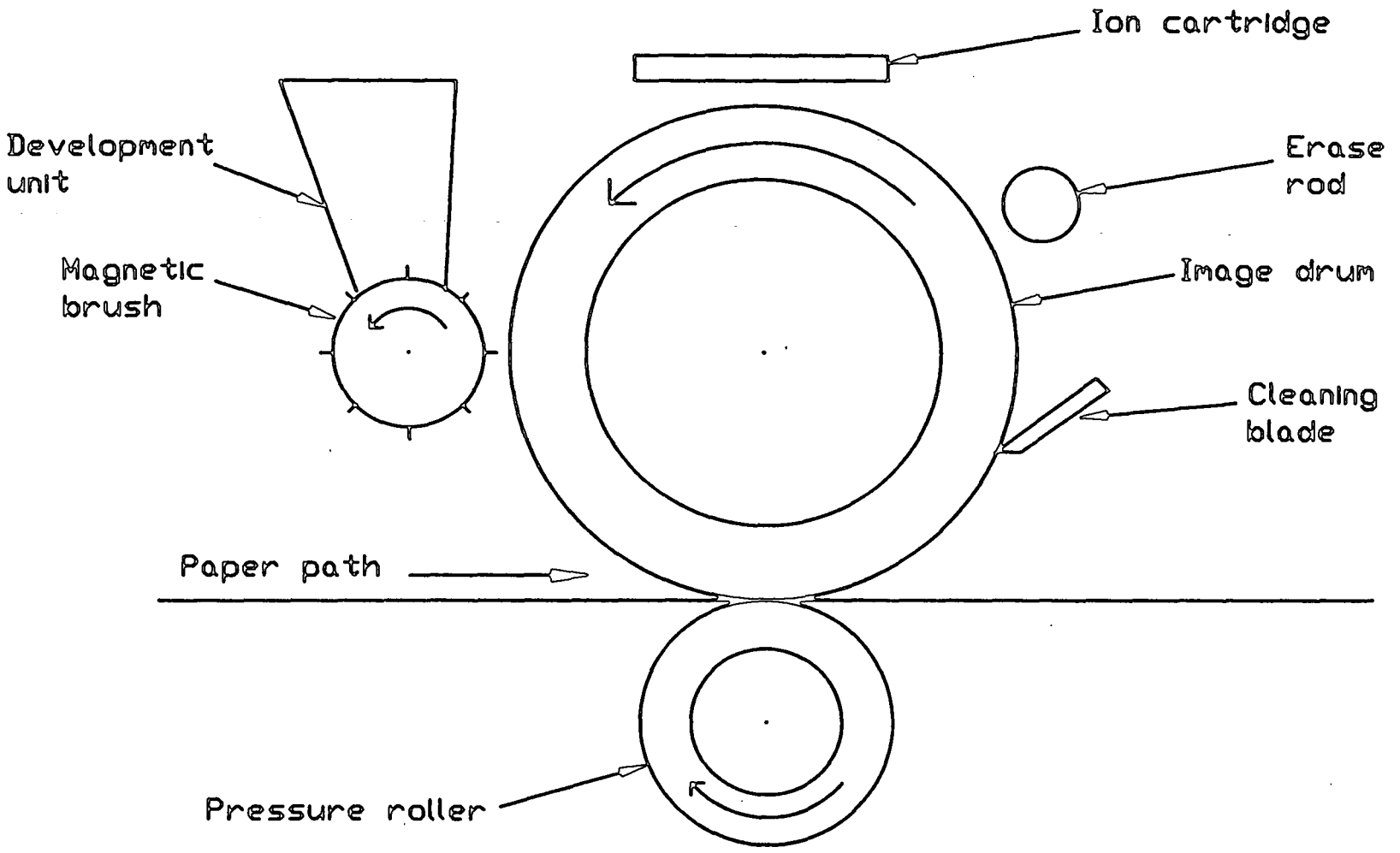


Figure 3.5 *Schematic cross section of the Delphax ionographic print engine.*

robust developer unit suited to high speed operation. Secondly, the toner has a longer useful life in the developer unit than dual component toners, where the carrier particles gradually lose their effectiveness as they become covered with impacted pigment. The toner brushed against the charged areas of the drum becomes inductively charged and is held to those areas by electrostatic attraction.

The ionographic printers commercially available employ the less common pressure fusing technique to fuse the toner to the paper. This uses a pressure roller underneath the imaging drum, between which the paper being printed passes. The high pressure squeezes the toner powder into the paper fibres, which has the effect of transferring and fusing the toner simultaneously. The force between the drum and the pressure roller on one print engine is specified as 800 Kgf across a width of 210 mm. This is obviously so high that shoe components would be damaged. However, it was thought possible that the engine could be operated at a lower force that would not damage the shoe components, but would still transfer the toner and fuse it sufficiently for stitchmarking. There was also the possibility that a more plastic toner might be available which fused under a lower pressure. If this lower pressure working were not successful, then the pressure roller could be removed and a transfer corotron could be used [49] together with a heat fusing toner.

Even with the problems of the small range of toner types available and the use of pressure for transfer and fusing, the advantages of this mechanism over laser printers appeared to be quite significant. These were the extremely hard wearing drum surface, the printing speed, the mechanical simplicity, the accuracy of charge placement on the drum from the ion cartridge, and the moderate price for the various components. Print engines are available from Delphax Corp. for printing up to 210 mm wide, at speeds of 0.15 to 0.38 m/sec, and the manufacturer has reported that a 345 mm wide printer is being developed. The price for a complete printer to print 210 mm wide at a speed of 0.15 m/sec. was quoted as about £6,000.

3.4 Printers used for similar purposes

There is some information available on a few examples of printing

technologies being used for computer controlled printing onto apparel materials.

Keeling [40] reports that Cambridge Consultants Ltd. is developing full–width ink jet printers made from several deflection devices in a linear array. These will be used for printing on a variety of materials, including fabrics, under computer control. They will allow unlimited scope for complexity of the printed design with the capability of instantaneous change from one to another. Although not yet commercially available, a prototype capable of printing at 1 m/sec with a resolution of 4 points/mm has been demonstrated.

Use of xerographic techniques for printing designs on fabrics have been reported [57] [58] [59]. Results using conventional copier based equipment have been good, though work to give better wear performance of the toner is continuing.

3.5 Conclusions

Full–width printers are mechanically simple and are able to print continuously with the components travelling at constant speed. These give the advantages of reduced maintenance, simple control, and high throughput, compared with single point or small area print mechanisms.

Ink jet printers are very fast and have a range of different coloured inks available in formulations for use on a variety of surfaces. However, there are very stringent requirements for ink cleanliness otherwise the fine jets would become blocked, and this may happen too if the ink is allowed to dry in them. Also, the only full–width ink jet printer currently available is very expensive.

Electrographic printers are also very fast, are less susceptible to contamination by dust and dirt, and are mechanically simple. An examination of several types of electrographic print engine confirmed that the ionographic printer was the simplest and most robust at a reasonable price. It is also potentially more accurate than a laser printer, and was the first choice for printing stitchmarking, though there were reservations about the small printing width and the lack of any variety of toner types available.

The second choice was the laser printer, which is not as robust and is inherently less accurate if using a scanned laser.

CHAPTER 4

COMBINING THE RECOGNITION AND PRINTING PROCESSES

4.1 Introduction

Chapter 2 concluded with a list of recognition system parameters that would be the most suitable for classifying shoe upper components. This largely coincided with those that had evolved at City University and BUSM, so there was no pressing need to modify the recognition system. Chapter 3 reviewed the printing technologies that could be used for stitchmarking, and decided that an ionographic print mechanism would be the most suitable. This chapter deals with the integration of the recognition system and the printer, and the design of a mechanism for transporting the shoe components accurately from one to the other. Chapter 5 presents a detailed description of the stitchmarking system which evolved from this experimentation.

The experimental stitchmarking machine passed through two distinct phases. The first saw it used for a demonstration of the feasibility of stitchmarking using electrographic printing. The second phase used the experience of the first to produce a stitchmarking system that would meet the required specification, as listed in section 1.5.

4.2 The shape recognition system

The basis for the stitchmarking machine was to be an experimental recognition system that had been built at BUSM based on that devised at

City University [3]. Photographs of it on arrival at Durham University are shown in plates 4.1 & 4.2, and a schematic diagram in fig. 4.1. As explained in chapter 1, as much as possible of the recognition system and mechanical components of this machine were to be retained unless there were significant advantages in changing anything for stitchmarking.

The machine had a reciprocating glass table to carry each shoe component to be recognised, and overall control and feature matching was performed by a Perkin–Elmer 1620 (PE–1620) 16–bit mini computer. Data from the camera was fed to edge detection and coding hardware, and a Z8000 16–bit microprocessor system, configured as a slave, calculated the features of the scanned shape. Another Z8000 microprocessor system, configured as the master, controlled the drive to the table and was used for communicating between the PE–1620 and the slave.

For imaging the 1728 pixel CCD linear imaging device used at City University had been replaced by a 2048 pixel device increasing the resolution by reducing the digitisation cell size from 0.35 mm to 0.203 mm. The reciprocating glass table to carry the components was driven by a stepper motor through industry standard toothed belts to give a step size of 0.203 mm. To match this, the camera was positioned at such a height that the digitisation cells at the shoe component were square rather than rectangular. Since the sensor had a pixel spacing of 13 μm, and the digitisation cell size at the component was 0.203 mm, the magnification required was $13 \times 10^{-6}/0.203 \times 10^{-3} = 0.064$. Having 2048 elements it could image a width of 2048×0.203 mm $= 415.7$ mm.

The shoe components being scanned were illuminated from below by a fluorescent tube driven by a high frequency inverter. The high frequency combined with the persistence of the fluorescent coating minimises the variation of the light output during the AC cycle so eliminating stroboscopic effects. Since backlighting provides the highest contrast images, which give the greatest differentiation between the shoe component and the background, it was decided that it ought to be retained, if at all possible, when the machine was converted to stitchmarking. However, there was concern about the effects of dust, dirt, and scratches on any transparent medium supporting

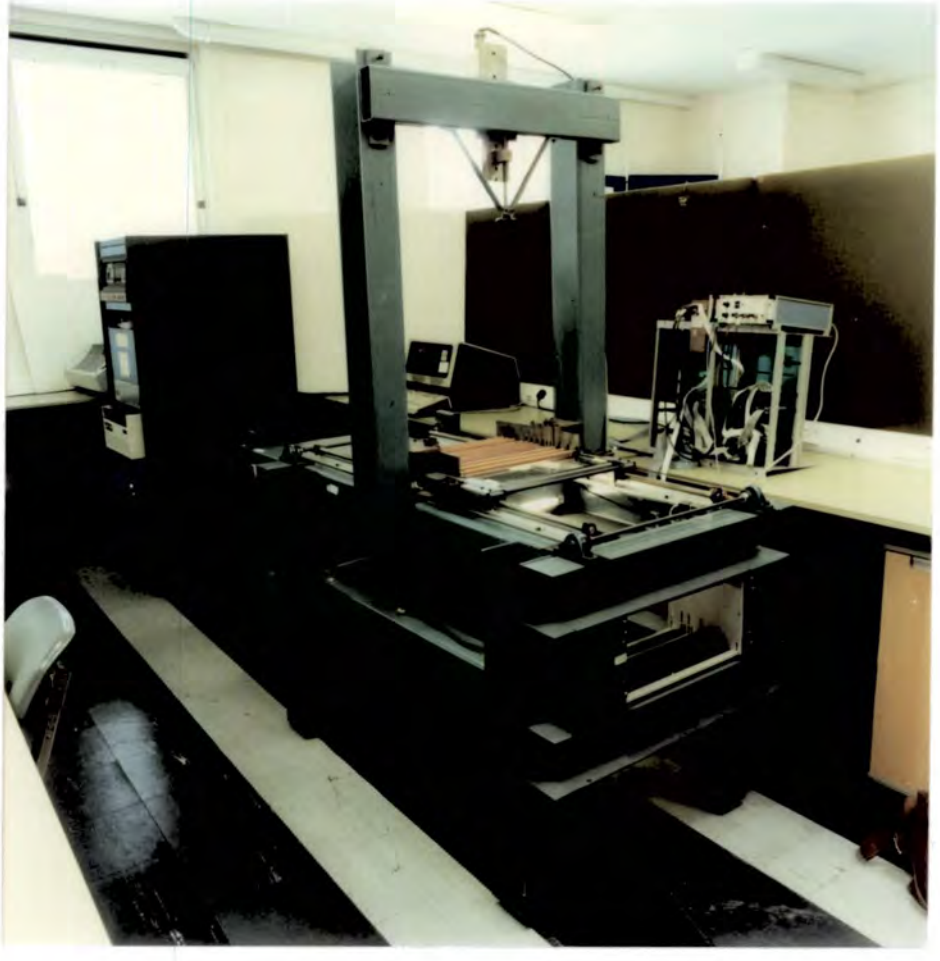


Plate 4.1 *The recognition system on arrival from BUSM (i).*

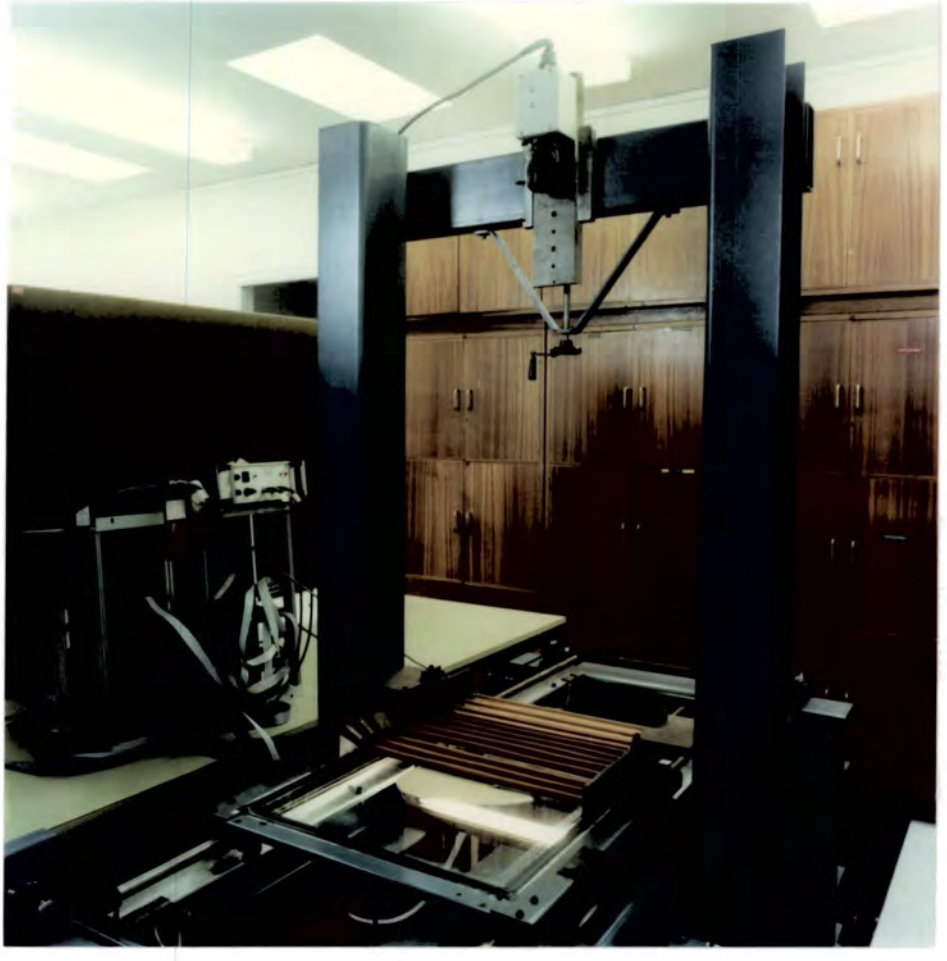


Plate 4.2 *The recognition system on arrival from BUSM (ii).*

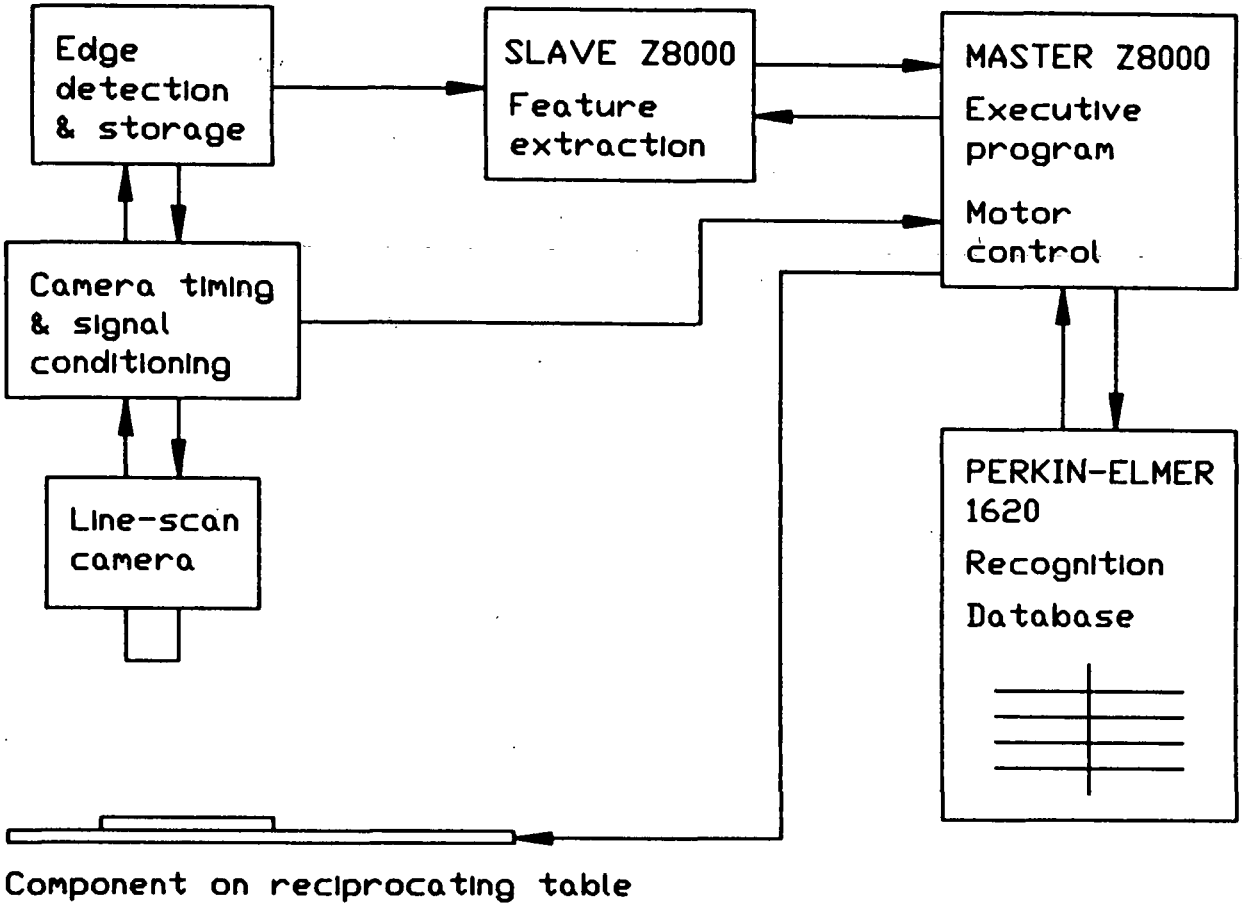


Figure 4.1 *Schematic diagram of the recognition system on arrival from BUSM.*

the components.

Although the feature extraction and matching hardware and software were based on those first developed at City University, they incorporated improvements that had been made at BUSM. A commercial camera incorporating the linescan imaging device was connected to a matching camera processing unit. This provided an interface between the camera and the computer system, and gave a thresholded video output. The digitised video output was directed into a pair of prototyping boards where pixels that were edge points of a shape were detected, encoded, and stored in memory, based on the method described by Koulopoulos [3]. This process was very fast since it was performed by hardware rather than software, so these boards were retained in the stitchmarking machine. The edge coordinates could be read by the slave microprocessor which extracted the features of the shape ready for matching. This too could all have been performed quicker by hardware, but this would not have had the versatility required at this experimental stage, where modifications would have to be made to the feature extraction.

The features extracted from the shapes of the shoe components were

i) the square root of the area,

ii) the fourth roots of the principal second moments of area, and

iii) 18 equispaced radii from the centroid to the outer edge, starting from the direction of the principal axis.

Some changes had been made to the recognition system at BUSM from that described by Koulopoulos [3] to improve the performance of the recognition system for use with shoe components. Firstly, it had been found that a shoe components's orientation could not be specified reliably enough using the longest radius as a datum, see section 2.3.6., and two other methods were used instead. For most shapes the principal axis was used as the orientation datum, though for some a polar coding technique had to be used, which is described in detail in section 5.8. Secondly, to speed up the matching process, the coding theory technique was not performed each time on all the classes within the knowledge base. Instead, a binary decision technique, see section 2.3.5, was used on the area and principal second moments of area,

with a method based on the coding theory technique used only on the radii of classes which passed those tests.

4.3 The printing mechanism

The recognition system was basically complete, but a printing mechanism was required to be connected to it. Chapter 3 concluded that an ionographic printer would be the most suitable for continuous use for high–speed stitchmarking. However, the versions of this type of printer commercially available used pressure to fuse and transfer the toner to the substrate. This would probably be unacceptable for leather, and would require work on electrostatic transfer.

A crude check on the suitability of the electrographic printing process in general, and electrostatic transfer of toner onto leather in particular, was performed by coaxing some A4 sized pieces of thin leather through an office copier. The print quality was excellent, but the leather was overheated by the copier fusing rollers. This was apparent since the leather emerged slightly harder and had a glossy surface. Although this would be unacceptable for shoe components, total fusing of the toner would not be required for stitchmarking, allowing the possibility of the heat being reduced to an acceptable level.

With the feasibility of electrostatic printing onto leather now demonstrated, the process of obtaining an ionographic printer was started.

4.4 Component transport

At this stage the machine still had the reciprocating glass table for moving the shoe components past the camera. A mechanism was now required which would do this and also accurately move the components onto and through an electrographic printer. It was decided that for high throughput this should be a continuously moving device rather than a reciprocating one, and the requirements of the printer would also have to be borne in mind when designing it. The use of robots as part of the system was rejected as being too slow to meet the projected speed of future machines developed

from the experimental one. They would also be expensive and would have problems in handling floppy shoe components with a variety of shapes and sizes.

Various arrangements of drums and belts, as shown in figs. 4.2 & 4.3, were examined and discarded. Although the use of a drum for component transport might have provided a compact machine, due to the uncertainties in the time required for the computations after scanning, this would have caused too many constraints and not have the versatility of conveyor belts. Also, there might be problems with the need to use a corona device through the transport drum to transfer the toner from the printing drum to the shoe component.

The desire to backlight the components and the concerns about transparent belts becoming scratched and dirty led to a design which used two conveyor belts with a gap between them through which the camera scanned. Experiments had to be performed with this arrangement to make sure that components did not fall through the gap, or droop into it so far as to cause errors in recognition.

4.5 Feasibility of electrographic stitchmarking

To recapitulate, it had been decided to retain initially the recognition system of the machine in the form that it was when received from BUSM, to transport the components on two conveyor belts, scan them using backlighting through the gap between the conveyors, and print the stitchmarking using an ionographic printer.

Unfortunately, initial attempts to obtain an ionographic print engine were prolonged, so a low cost laser printer was purchased in order to prove the principles. A Hewlett Packard Laserjet+ printer was chosen since it appeared to have a construction that could be adapted for use on the machine, was available for immediate delivery, and had one of the largest graphics memories. It also printed at the same resolution as the ionographic printer and the full technical manuals were available from Hewlett Packard and also Canon, the print engine manufacturer. This printer was a low speed, low cost

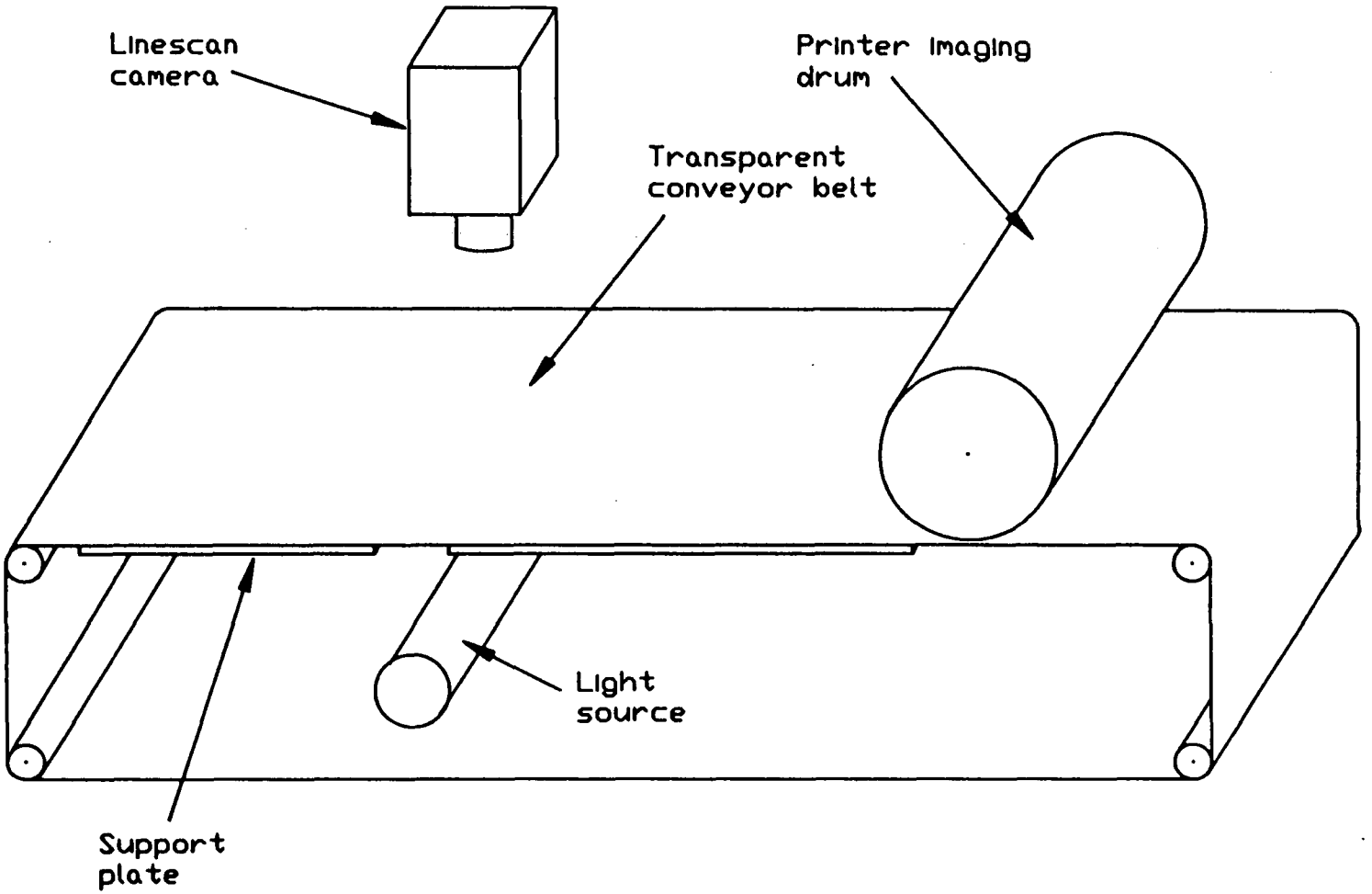


Figure 4.2 *Machine arrangement that was rejected (i).*

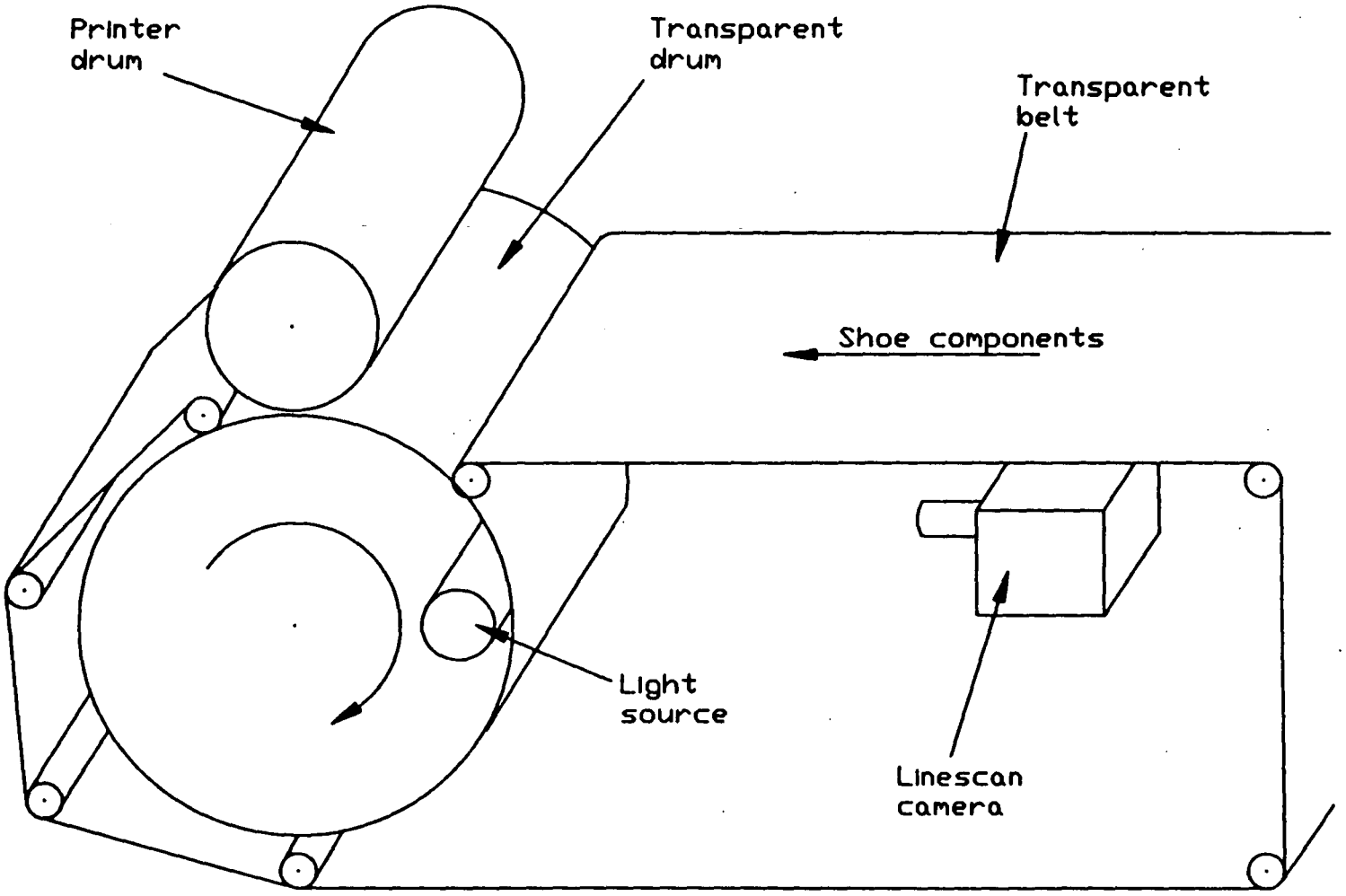


Figure 4.3 *Machine arrangement that was rejected (ii).*

printer designed for use in an office environment and could print a width of 210 mm at only 40 mm/sec.

Initially, it was decided to prove that the printer would print as required onto shoe components without being too concerned about the time taken. The construction of the printer allowed the top half to hinge upwards away from the bottom half for maintenance purposes. The top half contained the print engine and the control circuits, and the bottom half contained the drive motor, the paper delivery mechanism, and the fusing stage. By splitting the printer at its hinges it was possible to mount the top half so that it was able to print on surfaces passing below. The bottom half was left electrically connected since the print engine expected it to respond appropriately when printing. The only additional requirement was for the transfer corotron to be mounted under the substrate being printed to transfer the toner from the drum. This arrangement, with the engine mounted over a conveyor belt on the machine, is shown in the photographs of plates 4.3 & 4.4.

Consideration was given to methods of holding the shoe components flat during scanning and printing, and ways of making sure that they were moved accurately from one to the other. As can be seen in previous plate 4.2, the machine on arrival from BUSM had a set of rubber covered idler rollers under which the shoe components passed on the reciprocating glass table during scanning. Various methods of fixing the components to the proposed conveyor belts were considered, including the use of some form of vacuum belt [60], air jets pressing the components onto the belts, electrostatically charging the belts, use of adhesive layers, and the use of top belts or rollers to press the components against the conveyor belts. However, initially for simplicity it was decided to use plain conveyor belts and then assess what, if any, additional holding arrangement was required.

During this time, the best type of conveyor belt and drive were also considered. To give a positive drive to the belts and prevent them moving sideways it was decided to employ belts with sprocket holes along the edges. To test the feasibility of using these belts initial experiments were performed using strong teletypewriter paper, of which there was a plentiful supply. This had a width of 215 mm, which was compatible with the laser printer, was

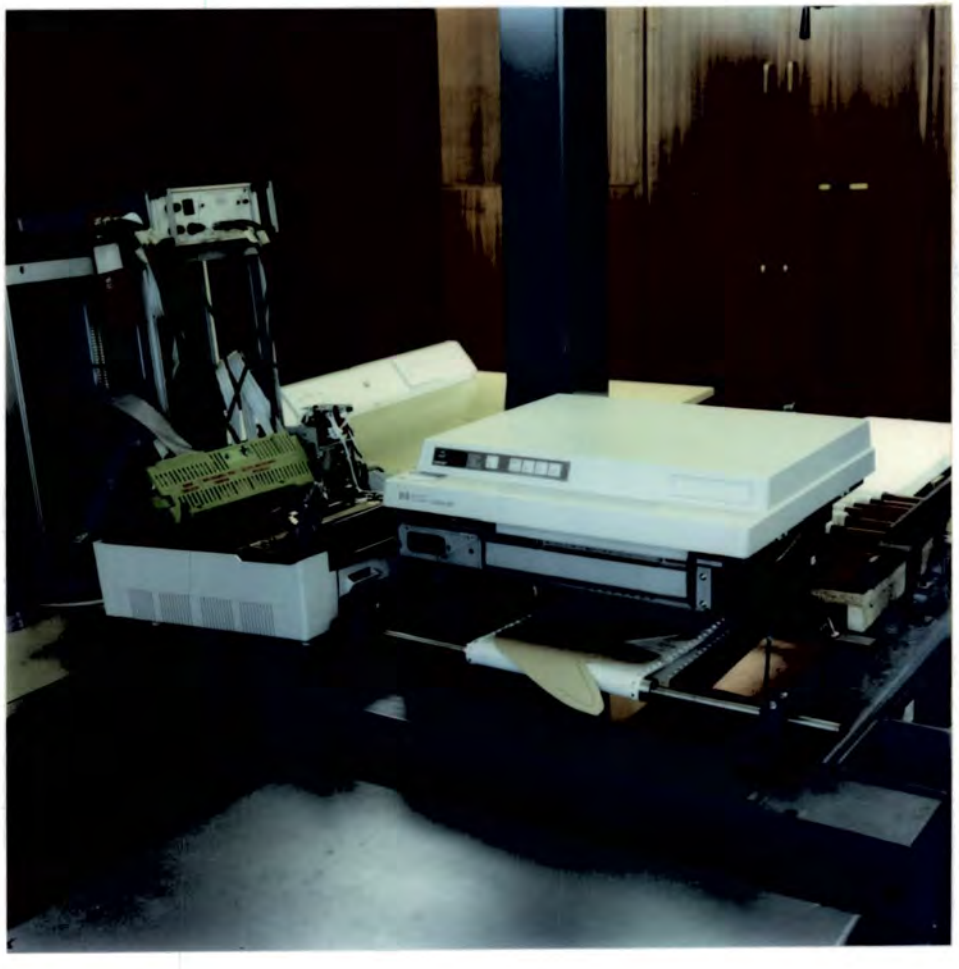


Plate 4.3 *The laser printer mounted for stitchmarking (i).*

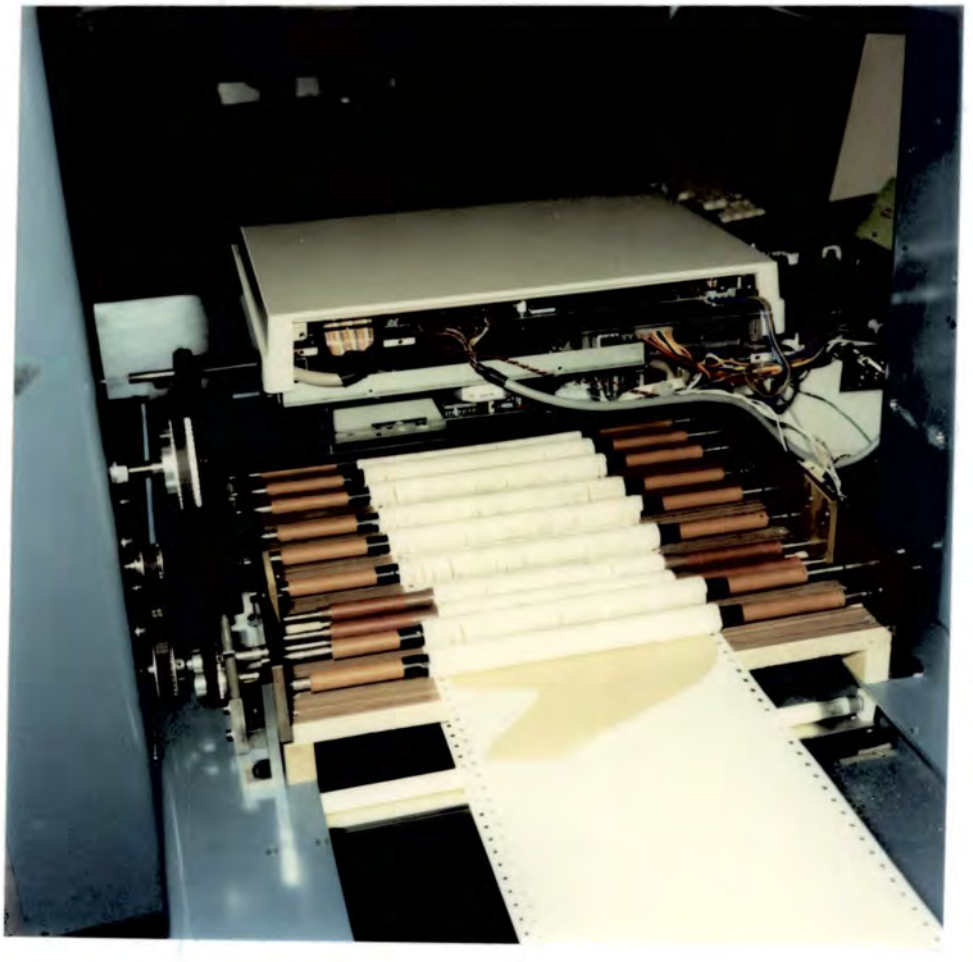


Plate 4.4 *The laser printer mounted for stitchmarking (ii).*

strong, and was thin so probably would not affect the electrostatic toner transfer. The paper had perforations of 4 mm diameter at 12.5 mm pitch down each edge, which would be suitable for accurately driving it.

A sprocketed drive roller system was made to drive the belts. This was driven by a stepper motor which was geared down using toothed belts to give a conveyor movement of 0.203 mm/step, the size of the digitisation cell. The sprocketed drive rollers at the camera scanning gap were made of as small a diameter as was practicable for driving the conveyors without them snagging on the sprocket pins. This was so that the separation between the feed belt and the takeup belt was as narrow as possible. This would minimise the potential for the components drooping as they passed over the gap.

The conveyor belts were made continuous by carefully joining together the ends of lengths of the teletypewriter paper with the perforations aligned. Horizontal flat sheets of plastic laminate–covered plywood supported most of the working surface of the belts, while an idler roller on the return side of the take up belt provided tension. The feed belt was only lightly tensioned, by adjusting its support roller separation.

The whole conveyor mechanism worked very well, though there was a tendency for the belts to ride up on the sprocket pins, probably caused by a slight mismatch between the pitches of the sprocket holes and the pins. The effect was reduced by polishing the shaft of the drive rollers with silicone wax which allowed easier slippage between the belt and the drive roller, so allowing compensation for this mismatch. However, over the long term the sprocket holes gradually became elongated. The machine operating programs were altered for use with the conveyor system and the details of approximately 150 components were entered into the system, using modified 'teach' programs. For teaching, a shoe component of a new class is put through the recognition system, the features of the shape are extracted and stored in the knowledge base together with the class name of the component, as described in detail in chapter 5. These 150 shapes could be recognised successfully.

The print engine of the laser printer was mounted over the takeup belt so that it depressed the belt by 2 mm. The toner transfer corotron was

mounted under the belt in the same position it had been within the printer, beneath the imaging drum, though a gap 1.5 mm larger had to be allowed for the leather to pass through. Fortunately, the corotron was able to obtain its high tension supply from the print engine via its normal contact.

The conveyor belt carrying the shoe components through the printer had to be driven at a reduced speed during printing so that it matched that of the print drum. The drum had a drive motor fitted with a tacho–generator, and was assumed to run at constant speed. Speed synchronisation between the belt and the drum was achieved by selecting a suitable time delay between pulses from the Master Z8000 system to the belt drive stepper motor. Signals also had to be output to the print engine to simulate the movement of a sheet of paper past the print engine's sensors, otherwise the print engine would indicate an error and stop printing.

It had been hoped that flat components could be scanned and would travel to the printer without moving about on the conveyor belt. However, at the slow speed required for printing the impulses given to the belts by the stepper motor drives caused the components to move appreciably on the belt. To overcome this a bank of foam rubber covered idler rollers was installed between the scanning gap and the print drum, to hold the components onto the belt by friction. This worked successfully, and no relative motion between the components and the belt could be detected by eye.

Two further rubber foam covered rollers were mounted over the feed belt just before the scanning gap. These, with the corresponding rollers on the other side of the gap, helped hold shoe components flat while they were scanned by the camera. Photographs of the machine with this arrangement are shown in the previous plates 4.3 & 4.4, and a diagram in fig. 4.4.

For stitchmarking with this arrangement a component was scanned, recognised, and its position and orientation were determined. Each component had a database file within the PE–1620 which gave details of its stitchmarking. The contents of this file consisted of the coordinates of all of the pixels in the image of the taught shoe component which needed to have stitchmarking printed on them. These details were entered either manually,

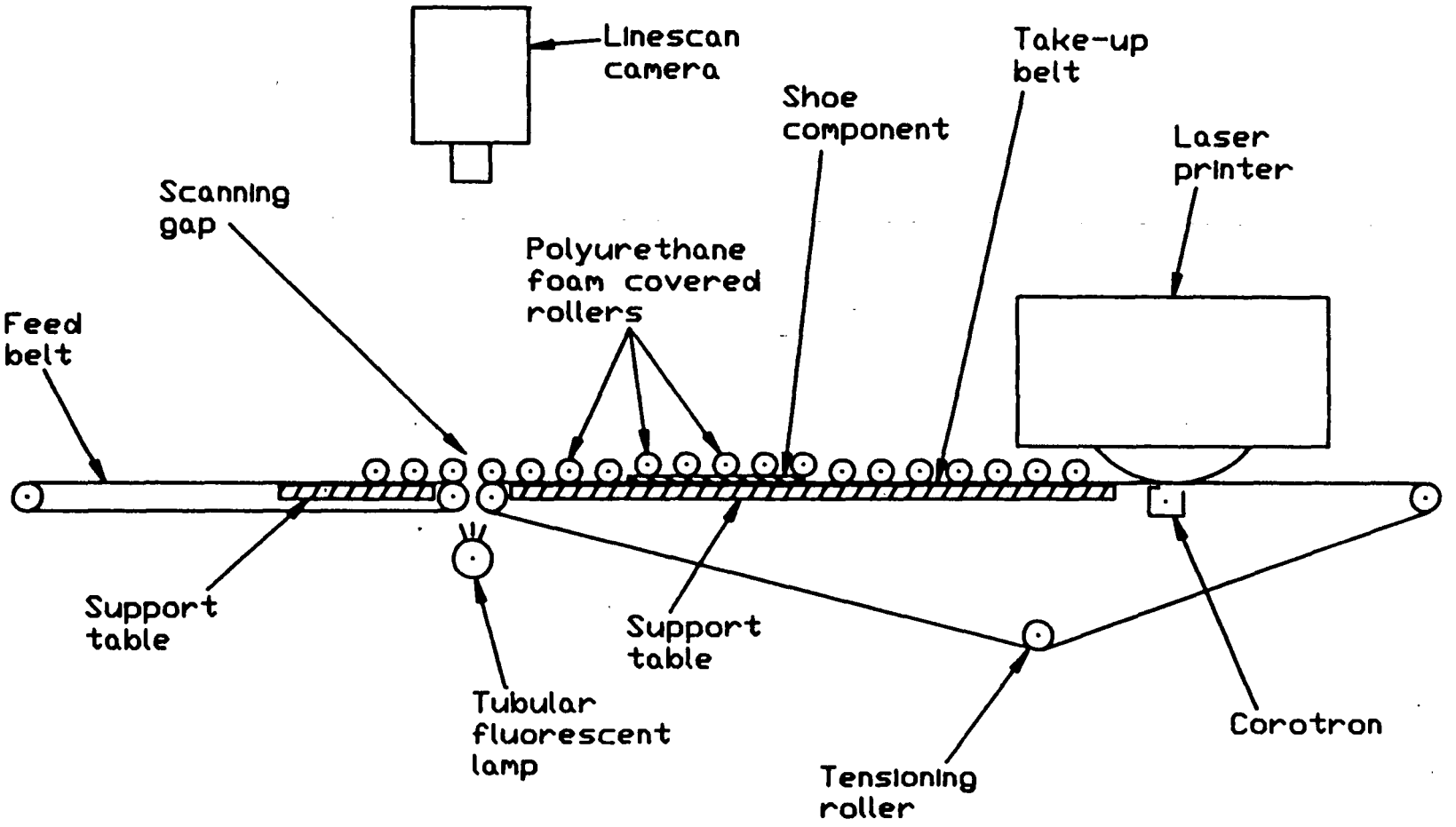


Figure 4.4 *Schematic diagram of the laser printer mounted with the vision system for stitchmarking.*

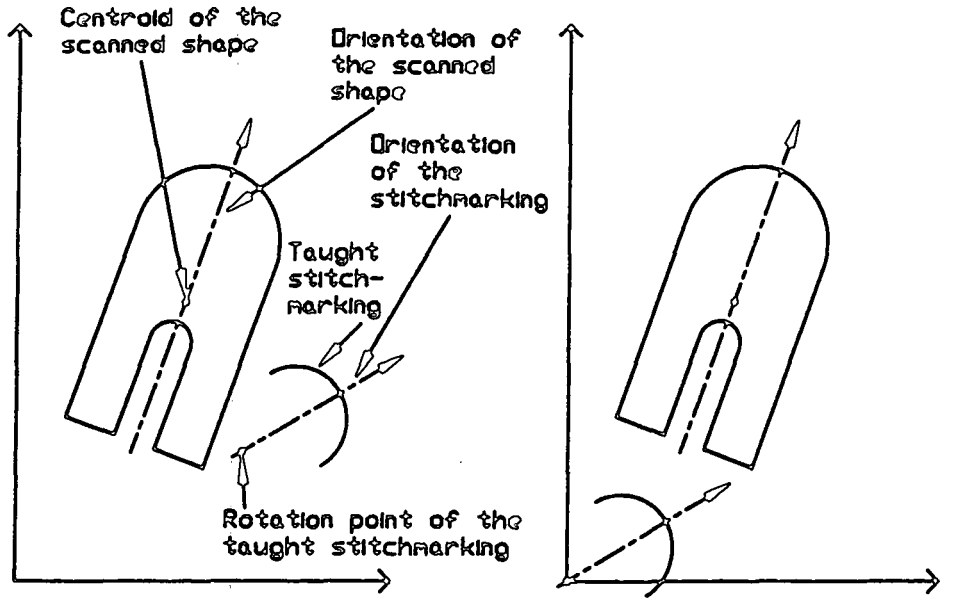
or could be derived by scanning with the camera a suitably cut shape and then editing the edge point file so produced. The stitchmarking coordinates had a rotation point which was the position of the centroid of the shoe component they would print on directly, and had an orientation which was the orientation of this component.

For printing, the pixel coordinates of the stitchmarking, which were in the coordinate system of the camera, had to be converted to the coordinate system of the printer. As well as applying an offset to the origin it was also necessary to perform some scaling. Since the laser printer could not store a full page of raster graphics data at 300 dots/inch but could store it at the lower resolution of 150 dots/inch it was this resolution which was used for printing. Since a data file which had been produced by scanning a shape and editing had a resolution of 125 pixels/inch, this was converted to the 150 dots/inch resolution of the laser printer by outputting twice every 5th. pixel on each line and every 5th. line.

After recognition, in order to print the stitchmarking in the correct position on the recognised shoe component, the stitchmarking coordinates had to be geometrically transformed [61] in three stages, as shown in fig. 4.5. Firstly, a linear translation was performed so that the rotation point of the stitchmarking coincided with the origin of the printer's coordinate system. Secondly the coordinates were rotated so that their orientation coincided with the orientation of the scanned shape. And thirdly, they were translated again, so that their rotation point coincided with the centroid of the scanned shape. They would then print in the correct position upon the scanned shape. This transformation was performed by Fortran subroutines on the PE–1620, which then had to sort the resulting coordinates into the order to be printed and encode them for sending to the laser printer in its raster graphics format. The Centronics parallel interface on the PE–1620 could not be used for sending the data since it translated some of the binary data given to it, so a slower RS232 serial interface had to be used instead.

The sequence of events for stitchmarking was

i) scan the component,



I) Scanned shape & taught stitchmarking

II) Stitchmarking translated so that its rotation point is at the origin

III) Stitchmarking rotated to the same orientation as the scanned shape

IV) Stitchmarking translated so that its rotation point coincides with the centroid of the scanned shape

Figure 4.5 *Transformation to align the stitchmarking data with the scanned shoe component.*

ii) stop the conveyor belts with the leading edge of the shoe component the distance from the printer that it would move in the time between the generation of the Vertical Synchronisation (VSYNC) signal by the print engine and the actual start of printing,

iii) recognise the component,

iv) read and transform the appropriate stitchmarking data,

v) send the data to the printer,

vi) send the 'Start Printing' command to the printer,

vii) send the pulses to the print engine to simulate the passage of a sheet of paper through the printer,

viii) wait for the VSYNC signal to be generated by the print engine then commence moving the shoe component through the printer,

ix) stop the belts after moving far enough for printing the component to be completed.

It had been reported [62] that the average length of stitchmarking applied per shoe component is about 20 cms. At the resolution of 4.9 pixels/cm (125 pixels/inch), at which the stitchmarking data was stored, this gave about 1000 coordinate pairs to transform. The execution times of the transformation programs were obtained using a representative stitchmarking pattern with 970 points. Reading from disc and transforming these 970 coordinate pairs took 17.5 sec, while ordering the coordinates and preparing the lines of scan for sending to the printer took 37 secs. The total time to perform the complete printing of a component after recognition was 3 min 19 sec. Some of the extra time was due to the printer going through the process of trying to deliver a sheet of paper from its feed tray, but the majority was the time for data transmission via the RS232 link.

That the total printing time was well outside the required specification was due to

i) the large number of coordinates–pairs needed to describe the stitchmarking of each component, all of which had to be operated on by

the transformation algorithms, which were written in Fortran,

ii) the list of transformed coordinates then had to be put into the order in which they were to be sent to the printer as a raster graphics image,

iii) the raster graphics image for the smallest rectangle that would enclose the stitchmarking had to be sent, and this included a lot of redundant information where nothing was to be printed, and

iv) all of the raster graphics information had to be sent to the printer via an RS232 serial link.

Printing was speeded up by making use of the 'rules' facility of the laser printer, rewriting the transformation program in assembler language to run on the Master Z8000 microprocessor, and using a Centronics parallel communications link instead of the RS232 serial link.

The 'rules' facility allowed the specification of the size of rule, where a rule is a solid black rectangle. Then, by giving an X–Y coordinate within the page to be printed and issuing the 'print rule' command, a black rectangle would appear with its top left hand corner at the position specified, when the page was printed.

The advantages of this were

i) the rule can be specified as a square whose size was the width of the required stitchmarking line, so the amount of data needed to specify the stitchmarking and which had to be transformed was reduced,

ii) less data had to be sent to the printer, only the rule coordinates and the 'print rule' command, not whole lines of raster graphics data, and

iii) it did not matter in which order the rule coordinates were sent to the printer, so no sorting was required.

At first, the original number of coordinate pairs were retained, but were treated as the coordinates specifying the positions of 4 × 4 dot rules. This reduced the total time for transforming and transmitting the data to the printer from 3 min 19 sec to just 48 seconds. Additionally the time taken to read the coordinate pairs from disc was reduced from 7 sec to 0.5 sec by

storing them unformatted at 64 per sector rather than formatted at 1 per sector.

Another strategy for reducing the computation time involved the fact that when a new design of stitchmarking data was generated it had a rotation point which coincided with the centroid of the taught shoe component on which it would print directly. Every time a similar shoe component was scanned for printing the stitchmarking coordinates had to be translated firstly so that their rotation point coincided with the origin of the printer's coordinate system, as in fig. 4.5 (ii). By performing this translation once and for all at teach time, before the coordinates were stored in the database, one translation of all of the points could be removed at printing time. The time for reading and transforming the coordinates was now 19.3 sec and the time for transmission using the RS232 protocol was 22.5 sec.

After recognition the PE–1620 sent the contents of the appropriate stitchmarking file to the Master Z8000 via a high speed parallel link. A peripherals board of the Master Z8000 system was modified so that an output port performed the function of a Centronics parallel interface port. This allowed a high speed print data link to the printer to be established. The transformation of 970 coordinate pairs by the Master Z8000 took 1.7 sec, compared with about 17.5 sec for the PE–1620 using Fortran. However, there was a penalty of about 4.5 sec while the coordinate pairs were sent from the PE–1620 to the Master Z8000. Using the Centronics interface to communicate with the laser printer was about twice as fast as using the RS232 interface. This increase in speed was disappointingly small, and was due to the laser printer giving the 'Busy' signal immediately after receiving each character while it dealt with it.

A program was written which reduced to 1/3rd. the number of stored stitchmarking coordinates produced by scanning a design with the camera. Used with a program which printed rules as 7 × 7 dot squares this gave stitchmarking with fewer coordinates but which was still adequately printed. The overall timing for recognising the small narrow component gg104 and printing a curved line, 195 mm long, with 365 coordinate pairs was now about 22.5 secs as detailed in table 4.1.

Operation	Time (secs.)
Scan & recognition	3.5
Reading stitchmarking & sending coordinates to Master	2.7
Transforming coordinates	0.5
Transmitting coordinates to printer	3.9
Waiting for printer to start printing	5.0
Printing	7
Giving a total time of about	22.6

Table 4.1 *Times for the processes of stitchmarking the small narrow component gg104 using 'Rules' and the laser printer.*

This was about as fast as was possible with that computer system, and without large scale modifications to the printer to bypass its format and control stages. A schematic diagram of this final version with the laser printer is shown in fig. 4.6.

A quick check of the accuracy could be made by running the same component through twice, the second time rotated through 180°. The two sets of stitchmarking produced on the component would usually line up to within less than a millimetre, though occasionally it was slightly more. The reason for this was not extensively investigated, but it was conjectured that it might still be due to the component moving relative to the belt.

The quality of the stitchmarking produced was very good, though was always in black, of course. However, there were no problems in printing onto the normal range of leathers or onto PVC or polyurethane synthetic shoe materials, as had been feared. No fusing was performed, which had the advantage of allowing the toner to be easily wiped off so the components could be reused. A photograph of typical stitchmarked samples and a lining marked sample is shown in plate 4.5.

Although a speed increase of about 3 times would be needed, good stitchmarking was produced on a range of components. Thus, the feasibility

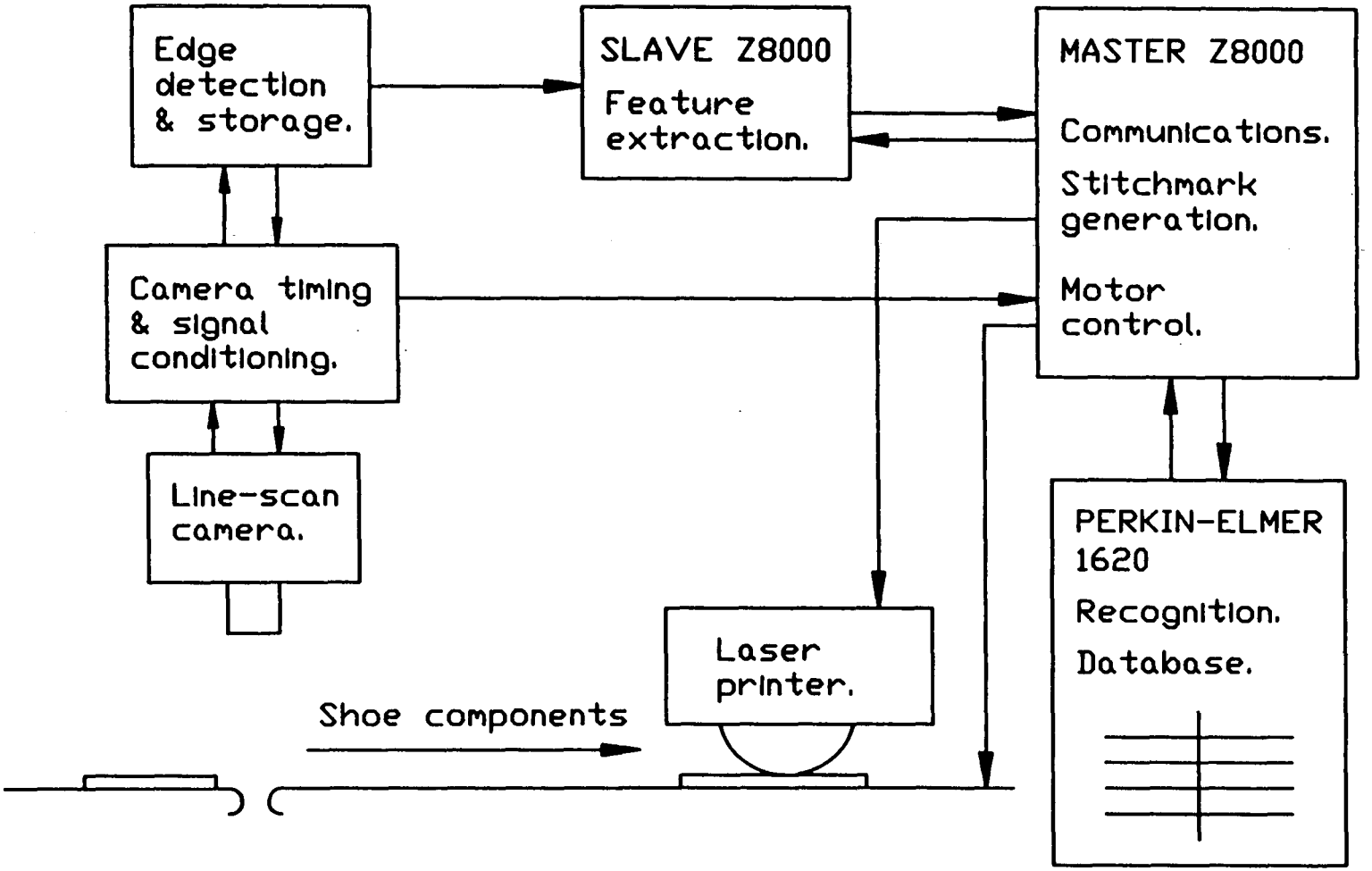


Figure 4.6 *Schematic diagram of the stitchmarking machine with the laser printer.*

Plate 4.5 *Shoe components stitchmarked and lining marked using the machine with the laser printer.*

of automatic stitchmarking with an electrographic printer had been proven, as required.

4.6 High speed stitchmarking

With the process having been satisfactorily demonstrated it was now time to convert the machine to high speed stitchmarking by using the ionographic printer, which had by then arrived. This coincided with a request from BUSM that the experimentation should be far enough advanced in a very short time so that the results could be used to produce a machine which would be capable of giving an impressive demonstration to potential purchasers at an exhibition. This gave added impetus and was to influence some of the decisions made during the rest of the project.

The Delphax S3000 ionographic printer that was obtained has been described in Section 3.3.2. It could print a 210 mm wide image at a speed of 18.5 cm/sec, which was 4.5 times faster than the laser printer, and used a single component, conductive, magnetic toner, which was fused to the paper by high pressure. The high pressure fusing caused calendering of the paper during printing which gave it a slight gloss, so it would have been surprising if the same did not happen to leather which was printed. This was confirmed by initial experiments running the complete printer in its 'test print' mode and feeding in A4 sized pieces of thin leather. The calendering of the leather during printing was unacceptable, so the fusing roll pressure was reduced gradually to try to find a point where the toner was fused sufficiently for stitchmarking but the leather was not damaged by the pressure. However, fusing stopped at a pressure which still damaged the leather.

To transfer the toner without damaging the leather it was decided to try using the standard electrographic technique of electrostatic transfer, as on the laser printer. To do this, the fusing roller was removed, and was replaced by a transfer corotron modelled on the one in the Hewlett Packard laser printer. This was connected to a variable voltage high tension supply, and experiments in electrostatically transferring the toner onto plain paper were performed. The densest image, judging by eye, was obtained with a corotron voltage of -4.5 Kv. The image density was somewhat less than had

been hoped, but the results were encouraging. However, experiments in heat fusing the toner on leather were unsuccessful since the leather was affected by the heat before there was appreciable fusing.

After consulting a company which manufactures toner for these machines a heat fusing toner, fusing at a temperature of about 80°C, was obtained which would also transfer more efficiently electrostatically, since it had a higher resistivity. Experiments showed that −4.5 Kv again gave the densest image, and this was much darker than the image produced by the standard toner.

Experiments were also performed in fusing this toner, printed onto a common type of brown leather, using a shoe factory cement heat–activator. This had six 500 W high intensity tungsten–halogen I.R. tubular lamps, each with an element length of 150 mm, arranged 50 mm apart in a parallel array. The duration for which the lamps were energised was controlled by a variable timer. With the printed leather 100 mm below the lamps the toner was not fused to the leather after 5 seconds, and was solidly fused after 10 seconds. However at times around 7 seconds the fusing was as required, sufficient to withstand handling but removable with vigorous rubbing. Fortunately, the leather was not adversely affected by this amount of heat. By reducing the separation between the leather and the lamps to 35 mm, it was possible to reduce the time to give the required degree of fusing to 2 seconds. The same conditions would be experienced by a shoe component passing under an array of such lamps of about 400 mm length suspended over the conveyor after the printer. Such an arrangement would be acceptable. One beneficial effect of fusing was that the toner attained a higher optical density.

Although the perforated paper belts for conveying the components had worked well, their sprocket holes gradually elongated so that eventually they had to be replaced. It was therefore decided that something more robust would be required for use with this faster system. Various options were considered, including making duplicate belts from rubber covered fabric. However, there were concerns about punching the perforations accurately in such a belt, and fears that elongation of the holes would still be a problem. Investigation showed that toothed rubber belts in the width required, and

of a suitable long length could be obtained to special order. These would be very robust and would give good accuracy in movement, though there were concerns with the possibility of their sideways movement and the suitability of the transfer corotron operating through such thick material.

To check the feasibility of the transfer corotron working through a toothed rubber conveyor belt experiments were performed using narrow drive belts of the same construction. Several widths of this belt were taped together alongside each other and the assembly was supported between the drum and corotron in the ionographic printer. Toner transfer was still adequate for stitchmarking, so two toothed conveyor belts were obtained. To support and drive these belts new toothed drive rollers were made, with small flanges to contain any sideways movement of the belts.

The print engine was stripped out of the printer, and was supported over the takeup belt, so that it depressed the belt by 2 mm. The speed of the print drum then had to be synchronised to that of the belt. Initially attempts were made to drive the print drum using the friction of the conveyor belt. This was successful until a shoe component passed between the drum and the belt, when the reduced friction caused the drum to stop. Eventually it was decided to drive the drum from another stepper motor whose stepping pulses were obtained from the same drive as the conveyor belts. By using the best choice of timing belt reduction ratios available, it was possible to nominally match the peripheral speed of the drum to the belt within 0.2%.

The disc drives on the PE–1620 computer had been very unreliable and the machine lacked the versatility and range of accessories of more modern computers. Therefore, the decision was taken to replace it with a Compaq 386/16 personal computer, which although a fraction of the size was much more powerful and had a much higher capacity disc storage. Another major advantage in using this machine was the large range of add–on boards which were available to plug into its standard IBM–PC bus. Thus, to communicate with the Z8000 Master microprocessor a 48 line I/O board was obtained. At this time there was an urgent requirement from BUSM for their machine for exhibition, which also had the same control system with a Compaq personal computer, to be made operational. So that this could be

done without holding up the research with the printer the low level inter–computer communications programs were adapted at BUSM for use with the Compaq computer and the 48 line I/O board in the language 'C'. The Fortran shape recognition programs were also converted there from running on the PE–1620 to running on the Compaq computer. Copies of these programs were then used on the research machine of this project.

So that the stitchmarking data could be transferred as quickly as possible and speed up the printing, it was decided to interface to the ionographic print engine at as low a level as practicable, bypassing the print control and character generation stages within its format controller. A schematic diagram of the original printer control system is shown in fig. 4.7. The printer allowed a choice of two low–level data interfaces, both of which were within its 'de–skew' board. One was serial and used vertical and horizontal synchronisation pulses with a format similar to that of a vdu, and the other was an 8–bit parallel interface with handshaking. It was apparent that software could not be used directly to send data to the print engine, since the data transmission rate required was up to 10 million bits/sec, which is beyond the capabilities of the Z8000 microprocessor's instruction timing. It appeared that a frame store memory, which would hold the binary image of the area to be printed, was necessary. The stitchmarking pattern would then be built up as a bit pattern in this frame store, and the whole could then be sent to the print engine through its low–level interface.

After considering this problem, it was decided that a special purpose graphics processor would be able to perform the task of transforming the stitchmarking and generating the image in a frame store, and would also be able to manage the transmission of the frame store contents to the print engine. One such processor, the Texas Instruments TMS34010, was available on a software development board (SDB) for plugging into the IBM–PC bus. Examination of the circuit of this board showed that it could be modified to drive the ionographic print engine instead of a vdu, though it would be short of video memory in its frame store to meet the full stitchmarking specification. A TMS34010 Software Development Board was obtained, and after familiarisation with its capabilities the best way of interfacing it to the print

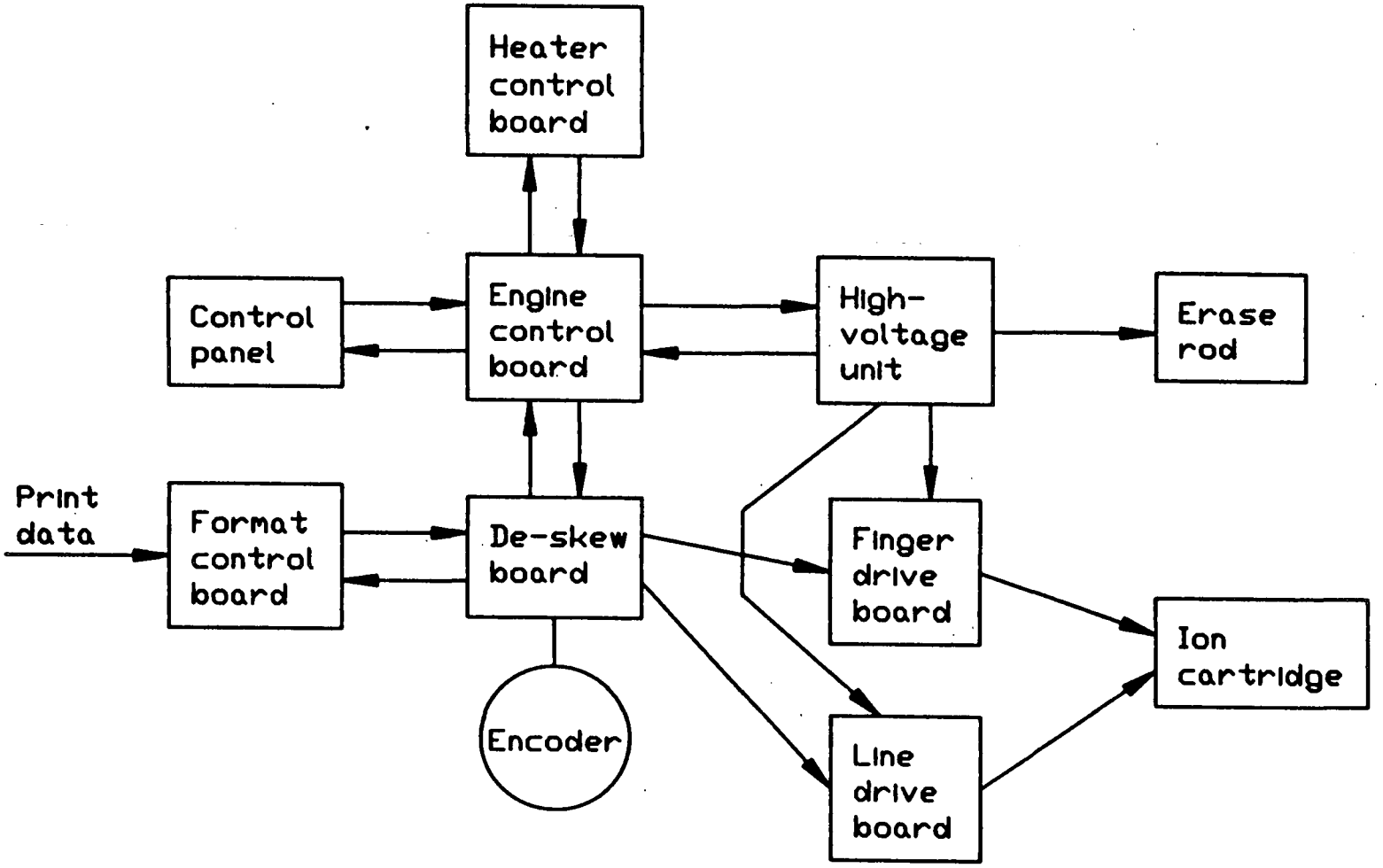


Figure 4.7 *Schematic diagram of the Delphax ionographic printer's printing control system.*

engine was investigated.

As has been mentioned, the SDB did not have sufficient video memory for printing stitchmarking along a long length of shoe component. Its Video RAM (VRAM) capacity was 256 Kbytes, which at 1 bit/pixel, could store 2,097,152 pixels. The Delphax printer printed at 11.8 dots/mm (the industry standard of 300 dots/inch) both along and across the direction of belt movement. Since the printing width was 203 mm this meant that each printed line was composed of $203 \times 11.8 = 2400$ dots. However to make full use of the capabilities of the TMS34010 instruction set, the line length specified for the image in its VRAM has to have a value of 2^n pixels, where n is an integer. The smallest value of 2^n greater than 2400 is 4096, so this was the required number of pixels/line in VRAM, though 1696 pixels in each line in memory would not be used. With this line length the VRAM could hold sufficient data to print $2,097,152/4096 = 512$ lines, which at the printing density of 11.8 dots/mm gave a distance of only 43.4 mm along the conveyor belt.

The first reaction of adding more VRAM to the board was thwarted due to the complicated addressing used to access the memory, compounded by the use of programmable logic arrays and a multi–layer circuit board, which made modification very difficult. However, the printing resolution of 11.8 dots per mm at which the VRAM would be storing data was much more than was required for stitchmarking, so ways of using the available VRAM at a lower resolution were investigated.

One approach which was examined was to try to output multiple times from each bit within the VRAM, so that one bit in VRAM would correspond to a matrix of, say, 2×2 or 3×3 pixels in the printed image. However, this would again have required a complicated modification to the SDB.

A more attractive solution was to make use of the large amount of general purpose RAM on the board. The graphics processor can be programmed to treat this as an area of memory for image formation, but its contents cannot be output directly to a video device. The following strategy was then developed to print components up to 300 mm long. The stitchmark

image was generated at a lower resolution in an area of memory in the general purpose RAM on the SDB which was called the 'image RAM'. The VRAM was then filled successively with slices of this low resolution image expanded to the printing resolution as the component passed through the printer. The transfer and expansion of slices of the stitchmark image could be performed very rapidly using the graphic processor's 'Pixel Block Transfer' instruction. Fig. 4.8 shows the organisation of the image RAM and the VRAM, and the whole process is described in more detail in section 5.9.

As supplied, the SDB only had an output to drive vdus, so this forced the decision to use the serial rather than the parallel interface of the Delphax print engine, since this had a communications protocol which resembled that of a vdu. However, there was still a problem in that the SDB was designed for use with colour vdus and gave a 4 bit/pixel parallel output, whereas the print engine required a 1 bit/pixel input. To overcome this discrepancy, an interface board was designed [Appendix A] which took each set of parallel 4 bits from the SDB and sent them serially to the print engine in synchronism with its dot clock (DOTCLK).

The best way of storing the stitchmarking details, and generating the stitchmarking pattern were investigated. It would have been possible to store the coordinates of every point of the stitchmarking, but this would have required a large database capacity, would have taken a long time to read and transmit, and a long time to transform all the coordinates. More efficient ways of storing and generating the stitchmarking were investigated. It was found that the best overall performance would be obtained by storing a few sparse data points and interpolating between them. Then, when a component had been recognised the sparse stitchmarking data could be read off the Winchester disc database and sent to the graphics processor. The sparse data would then be transformed and the stitchmarking between them could be interpolated and the relevant bits to form the image could be set in the image RAM. Cubic interpolation was the first choice since it appeared to have the speed and quality required; its usefulness can be appreciated by its wide use in computer graphics [61] [63] [64].

Programs were written at BUSM for using the vdu as a graphic input

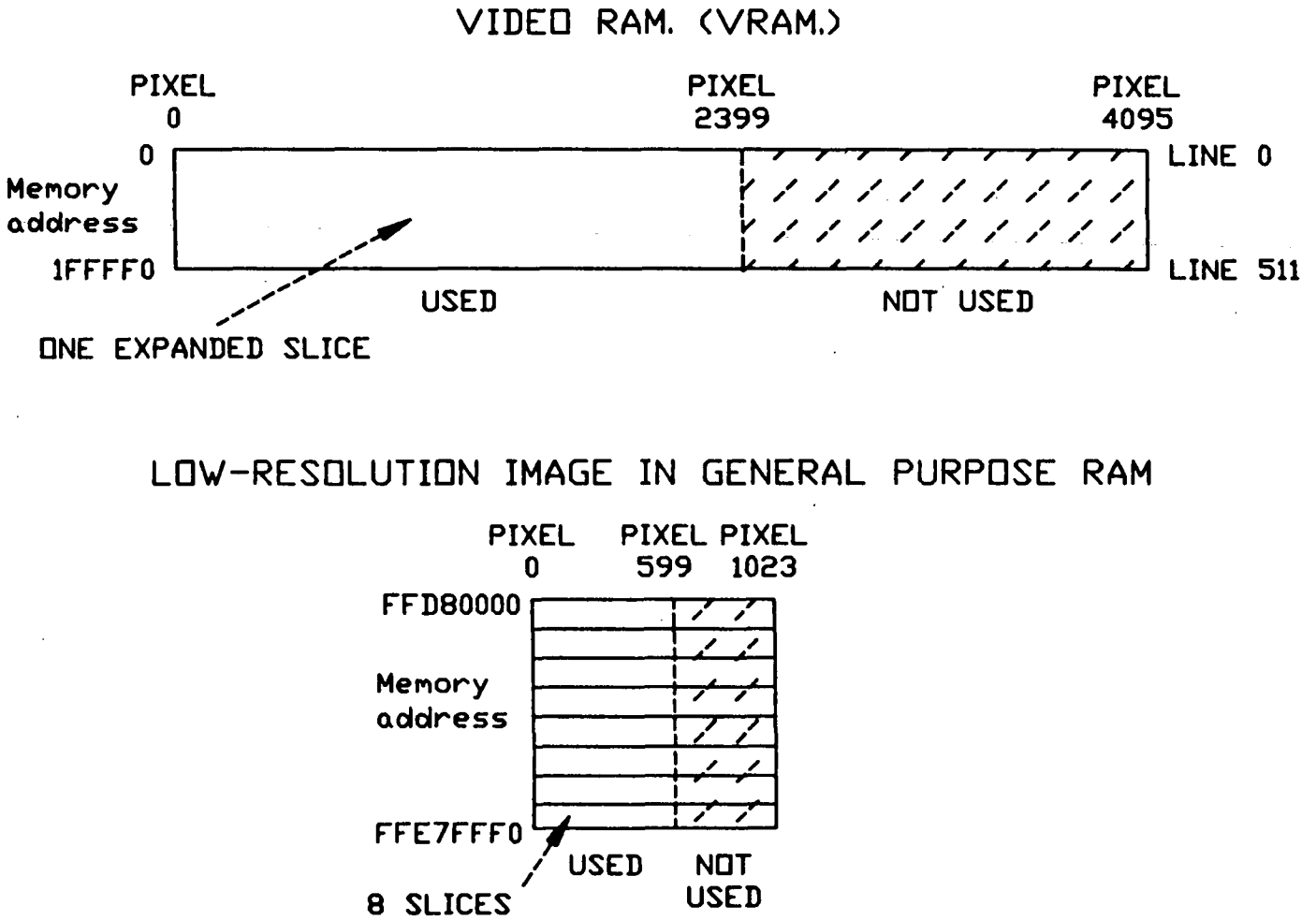


Figure 4.8 *Memory map of the VRAM and image RAM of the TMS34010 Software Development Board.*

device during the teaching phase for drawing the stitchmarking required on the shoe component. After storing the features of the component being taught, the edge coordinates of the shape were downloaded from the Z8000 Slave to the Compaq computer, which displayed the shape on its screen. A 'mouse' input device was then used to specify the points through which the stitchmarking was required, and cubic interpolation was used to draw the resulting stitchmarking on the screen. The points specified could be moved around on the image of the shape to finalise the stitchmarking. When satisfactory, the taught points were stored in the database under the code name of the component.

At Durham a version of the cubic interpolation was written to be run by the TMS34010 graphics processor. This program, when loaded with the stitchmarking points, reproduced in the SDB RAM, ready for printing, the stitchmarking design that had been displayed on the vdu at teach time.

The TMS34010 SDB bypassed the printer's format controller and was linked to the low level serial input via the interface board, as already described. The engine control board of the original printer was also discarded and its printing functions were performed by the Master Z8000. These included the output of various control signals to, and the monitoring of status signals from, the printer. A diagram of the modified control functions of the printer is shown in fig. 4.9.

4.7 Conclusions

The feasibility of stitchmarking on leather shoe components using an electrographic printer had been demonstrated with the laser printer. A Delphax ionographic printer had been successfully modified to print stitchmarking on shoe components using electrostatic toner transfer and heat fusing. This printer had been interfaced to the recognition system using a graphics processor, and a transport mechanism had been developed to convey the shoe components quickly and accurately from scanning to printing. Typical stitchmarked shoe components printed with the ionographic printer are illustrated in plate 4.6.

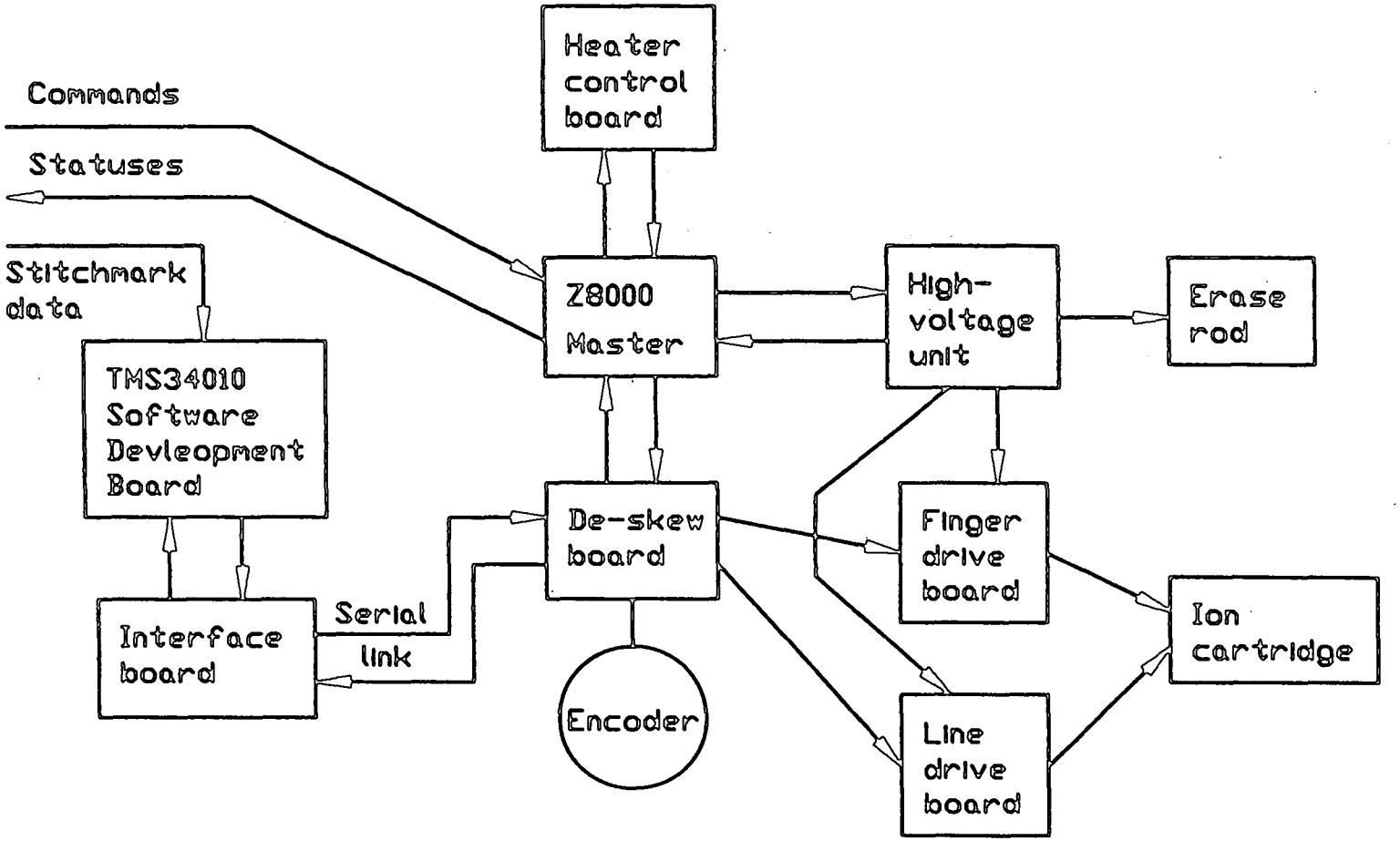


Figure 4.9 *Schematic diagram of the Delphax ionographic printer's printing control system after modifications for stitchmarking.*

Plate 4.6 *Shoe components stitchmarked with the ionographic printer.*

The machine was now in a state where it was ready for evaluation to see how well it met the specification. Its operation is described in detail in chapter 5, and the results of the evaluation in chapters 6 to 10.

CHAPTER 5

DESCRIPTION OF THE AUTOMATIC STITCHMARKING SYSTEM

5.1 Introduction

This chapter contains a description of the automatic stitchmarking system after fitting the ionographic printer. It will be helpful to refer to the photographs of plates 5.1 & 5.2, and the functional diagram fig. 5.1 as appropriate.

The machine consists of a substantial girder framework which supports the two conveyors, the camera, and the printer. As described in chapter 4 the camera, edge detection and encoding boards, and feature extraction and shape recognition programs are essentially unaltered since arrival from BUSM. The new subsystems produced by this project are the printing mechanism, its controlling hardware and software, the stitchmark manipulation programs, and the component transport mechanism.

5.2 Shoe component transport

The shoe components being processed have to be kept flat while they are transported accurately through the scanning station and on to the printing station.

The components are placed singly on the feed conveyor with a clear space between each, so that only one is scanned at a time, and with their

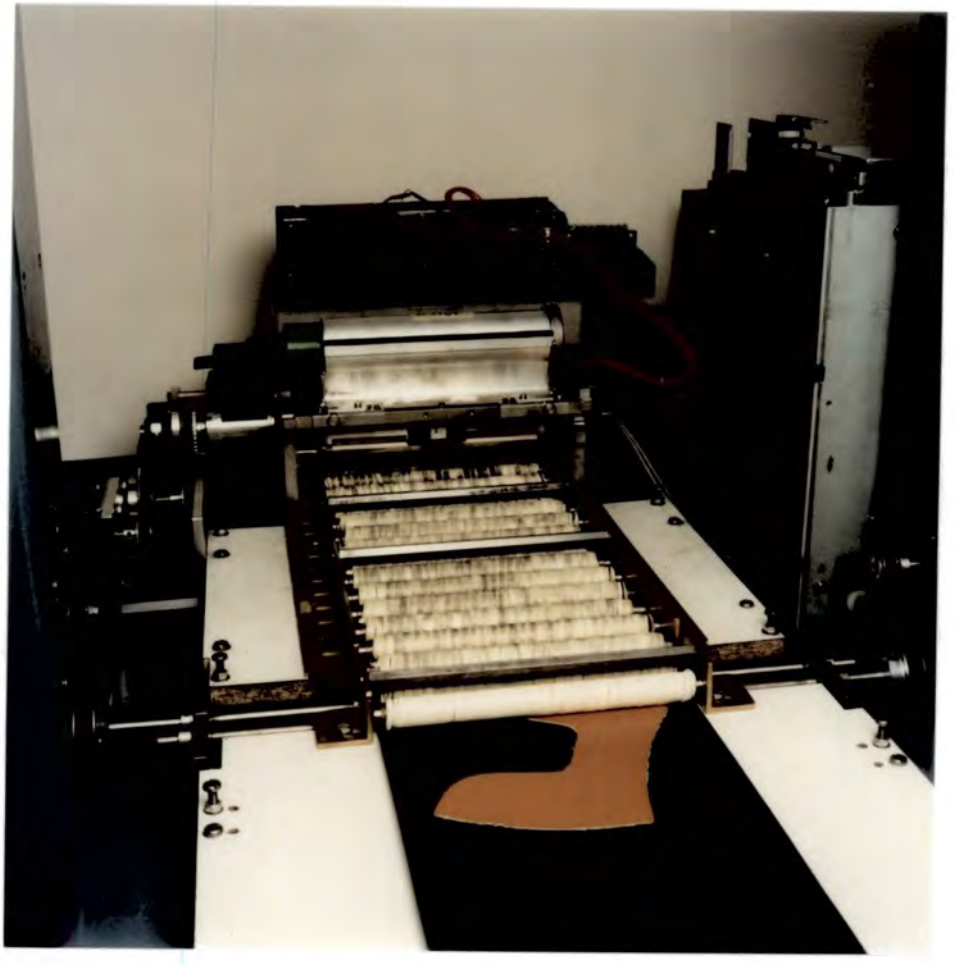


Plate 5.1 *The stitchmarking machine with the ionographic printer as a shoe component is being scanned.*

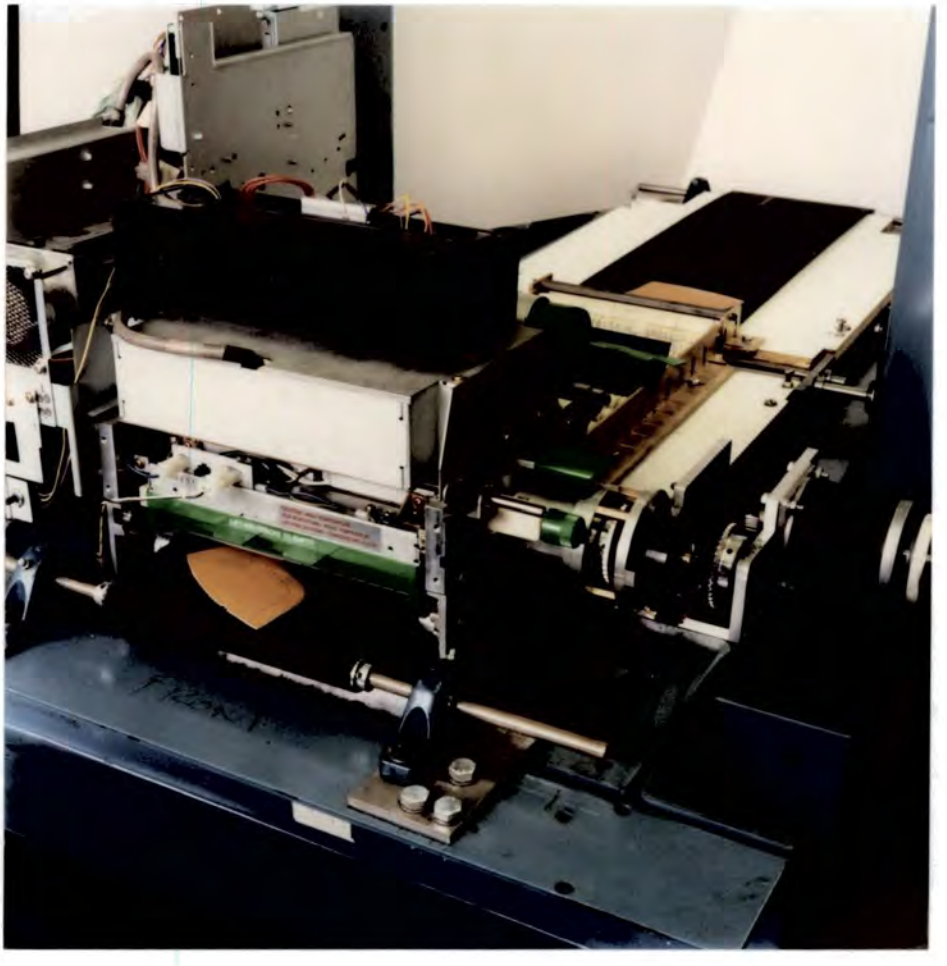


Plate 5.2 *The ionographic printer on the stitchmarking machine with a printed shoe component emerging.*

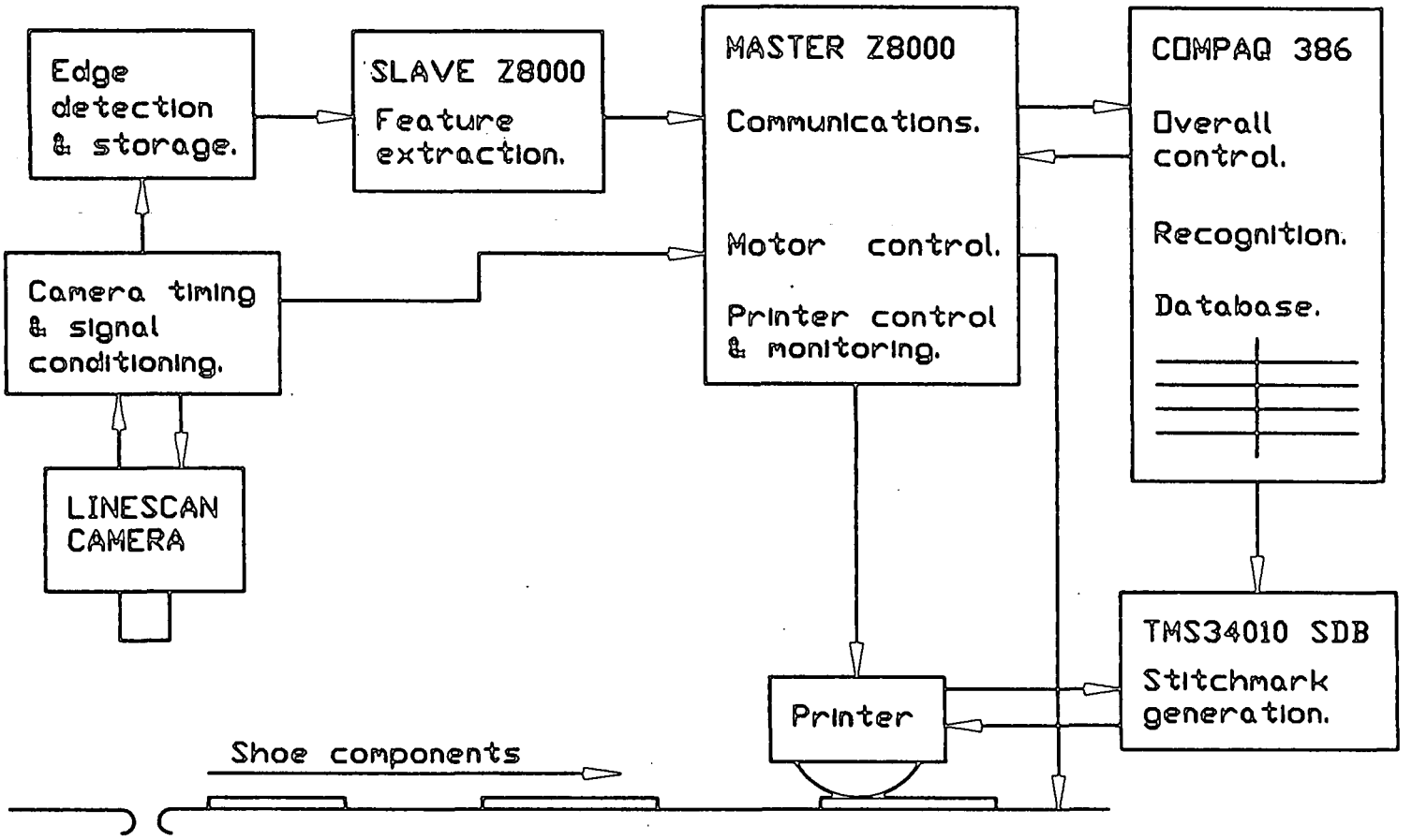


Figure 5.1 *Schematic diagram of the stitchmarking system*

surfaces to be printed uppermost. At the end of the feed conveyor they pass on to a takeup conveyor, over which the printing mechanism is mounted. There is a 1.8 mm gap between the two conveyors through which the component is illuminated from below in silhouette by a fluorescent strip lamp, driven from a high frequency power supply, and imaged from above by the linescan camera. Each conveyor is a 220 mm wide rubber covered fabric reinforced belt, toothed on the underside at a pitch of 5.1 mm (0.2 inch). The belts are individually driven by 200 step/rev stepper motors through toothed belt speed reduction stages to grooved drive rollers. The grooves on the rollers engage with the teeth on the belts, and end flanges prevent the belts wandering sideways. The two–stack stepper motors are powered from a 60 V supply through bipolar chopper drives. There is no feedback to sense the motion of the conveyor belts, since the motors are operated well within their capabilities and no steps are lost. The belts are accelerated up to a constant speed of 185 mm/sec for scanning, and are brought to a stop while the component is recognised before accelerating up to speed again for printing.

After being scanned by the camera, the shoe components are conveyed through the ionographic print engine, where the stitchmarking is applied. The separation between scanning and printing is limited by the maximum length of the wide, toothed belt that was available. This gives a scanning to printing distance of only 550 mm, though the effective distance is reduced by 157 mm since the stitchmark image is formed on the top of the print drum and has to travel round to meet the component at the bottom. The resulting short distance is of no consequence if the machine is used for recognition only, but when stitchmarking it is necessary for the belt to come to rest for the period of the recognition process. The maximum length of component that can be recognised and printed is limited to $(550 - 157) = 393$ mm, less the distance for decelerating and accelerating the conveyors of 30 mm each, giving 333 mm.

The shoe components are kept flat during scanning and printing, and are pressed against the conveyors to prevent them moving about, by idler–rollers covered in soft, resilient rubber.

5.3 Linescan camera

The linescan camera is mounted 95 cm above the gap between the conveyors. Its height and orientation are precisely set by screw adjusters to obtain the most accurate image of the shoe components. Within the camera, an image is produced on a 2048 pixel linescan imaging device by a 58 mm focal length camera lens. This focal length is quite a lot longer than that which would produce an image across the 27 mm long sensor, but only the central part of the image is used, where the quality is at its best. Other lenses have been tested at BUSM, but this one was used since it had given the best performance.

The lens and sensor are adjusted to give a pixel spacing at the shoe component of 0.203 mm (0.008 inch). The 'start/end of scan' pulses generated by the camera are directed to an interrupt input of the Master Z8000. The interrupt service routine initiated by the pulses outputs one step to the conveyor stepper motors, causing the belts to advance by 0.203 mm too. This results in the imaging of successive strips of the component at 0.203 mm centres, and the whole component is converted to an image on a 0.203 mm square grid. Each scan of the camera is completed in 1.1 msecs, and since the conveyors then move 0.203 mm the velocity of the shoe components through the machine is $0.203/(1.1 \times 10^{-3}) = 185$ mm/sec.

The camera is supplied with a 4 MHz clock, to which its video output is synchronised. The analogue video output from the sensor is thresholded to give a binary digital signal. A level '1' is output where light illuminates a pixel, and a level '0' where the shoe component blocks off light to a pixel.

5.4 Edge detection and coding

Since it is the outline of the silhouette which conveys the information about the shape, it is worthwhile extracting the edge points from the image data so that there is less for the feature extractor to handle. The digital output from the camera is examined in groups of three successive scans using hardware implemented logic. The digital value of a pixel is compared with the values of those pixels immediately before and after it on the same scan, and

the corresponding pixels on the previous and following scans, as illustrated diagrammatically in fig. 5.2.

A convolution is performed on each 3×3 matrix of pixels to determine whether the pixel under examination represents an edge point of the silhouette image. The coordinates of edge points so determined, measured from the start of the line of scan, are stored as x–coordinates in a buffer RAM mounted beside the edge detection circuits. This buffer has a capacity of 8192 edge points, which is sufficient for the majority of shoe components, but will have to be bigger to cope with such large items as boot legs.

Each edge point also has stored a code for its edge type (e.g. left, right, top, bottom) and a flag to indicate if it is the first edge point of the scan line. Note that only the x–coordinate (across the conveyor belt) of an edge point is recorded, since the y–coordinate (distance along the belt from the start of the shape) can be determined by counting the number of coordinates with the 'first edge point of scan' flag set from the beginning of the shape.

The edge–coordinate buffer has a capacity of 8192 edge point coordinates and buffers the detection and coding of the edge points against the rate at which they can be accepted by the feature extraction microprocessor. This is especially valuable if a long straight edge is imaged along one scan, when many pixels are detected as edge points in a very short time. The store has an output port, through which the coordinates are read by the Slave Z8000 microprocessor, as shown in the previous fig. 5.1, together with various status signals, such as 'store full' and 'no shape being scanned'.

5.5 Feature extraction

From the edge coordinates of a scanned shape stored in the buffer the features are extracted by assembler language programs running on the Slave Z8000 microprocessor. The unit of measurement for the calculations is the pixel spacing of the image, and the features calculated are

i) square–root of the area within the shape's outline (*RAREA*),

ii) fourth–roots of the Principal 2nd moments of area (*MMAX* & *MMIN*), and

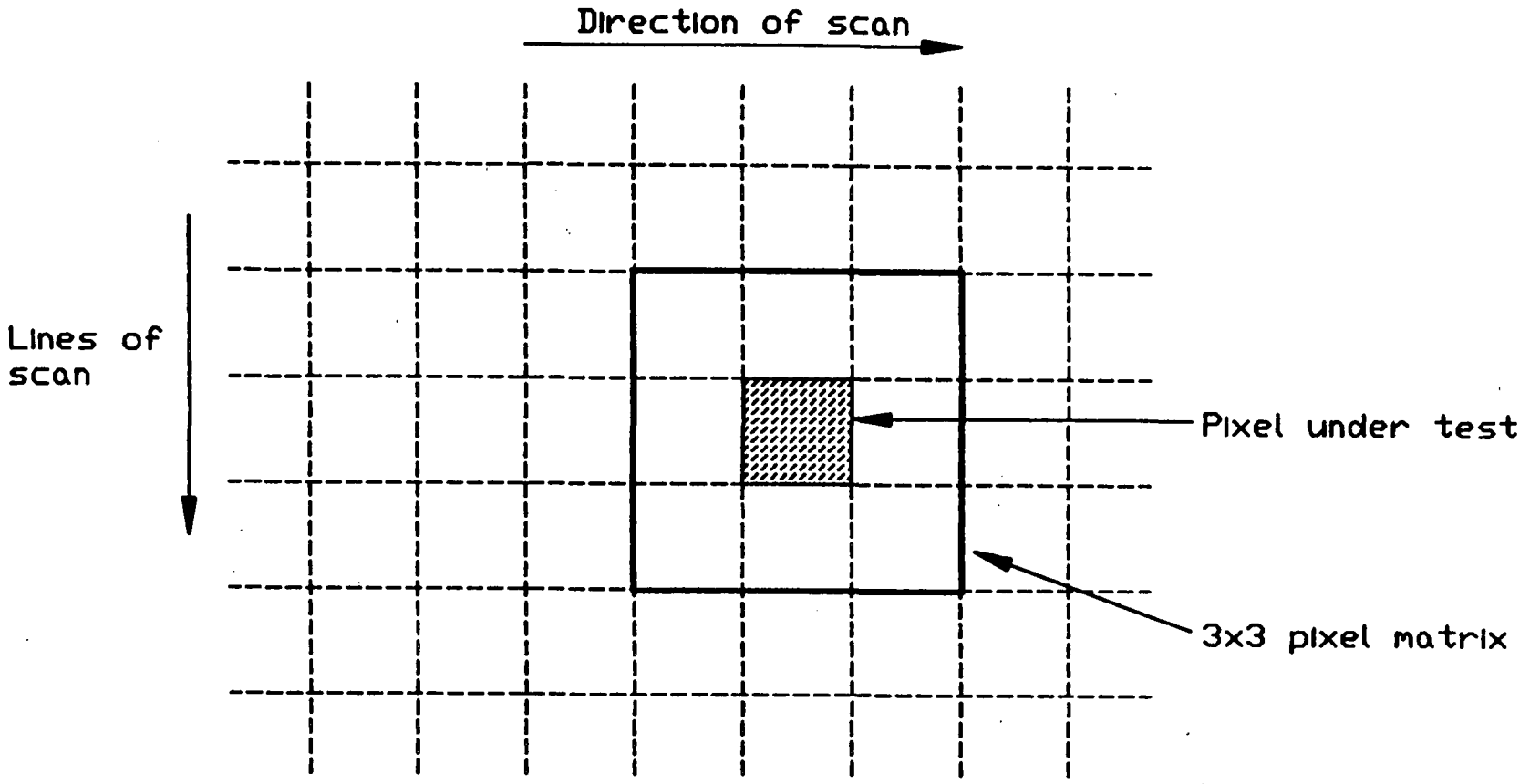


Figure 5.2 *Detection of edge points.*

iii) a set of 18 equispaced radii measured from the centroid to the edge, starting from the direction of the major principal axis (*RADII(1)* to *RADII(18)*).

Note that a tolerance is placed on the direction in which a radius is to be found, and it is the longest radius within the tolerance limits which is taken to be the feature radius. Multiple edge intesections are ignored, as shown in fig. 5.3. The only exception to this is when the shape is re–entrant so the centroid falls outside the boundary, in which case both the shortest and the longest radii are used, as shown in fig. 5.4.

As well as these features for recognition, the printing of stitchmarking also requires the calculation of the coordinates of the shape's centroid and the shape's orientation. The orientation given by the major principal axis is only reliably accurate enough for stitchmarking if the two principal second moments of area are sufficiently dissimilar, i.e. the shape is long and narrow. A parameter of the shape called the 'goodness coefficient' (*GCOEFF*) has been introduced at BUSM to decide whether the orientation is reliably determined with sufficient accuracy by the principal axis, where

$$GCOEFF = \frac{MMAX - MMIN}{MMAX + MMIN}.$$

If the 'goodness coefficient' of the shape is greater than a threshold value, which is determined by simulation and experiment [65], then the orientation of the component is specified with sufficient confidence by the calculated orientation of the major axis, and the shape is called a 'well–conditioned' or 'good' shape. If the 'goodness coefficient' is below the threshold then the two principal 2nd. moments are too similar to determine the orientation reliably and such a shape is called an 'ill–conditioned' or 'bad' shape. A polar coding and correlation technique has to be used instead, as described in more detail in section 5.8.

5.6 The teaching procedure

Before the machine can be used to recognise a new component a sample of that component has be fed through the machine while it is in the

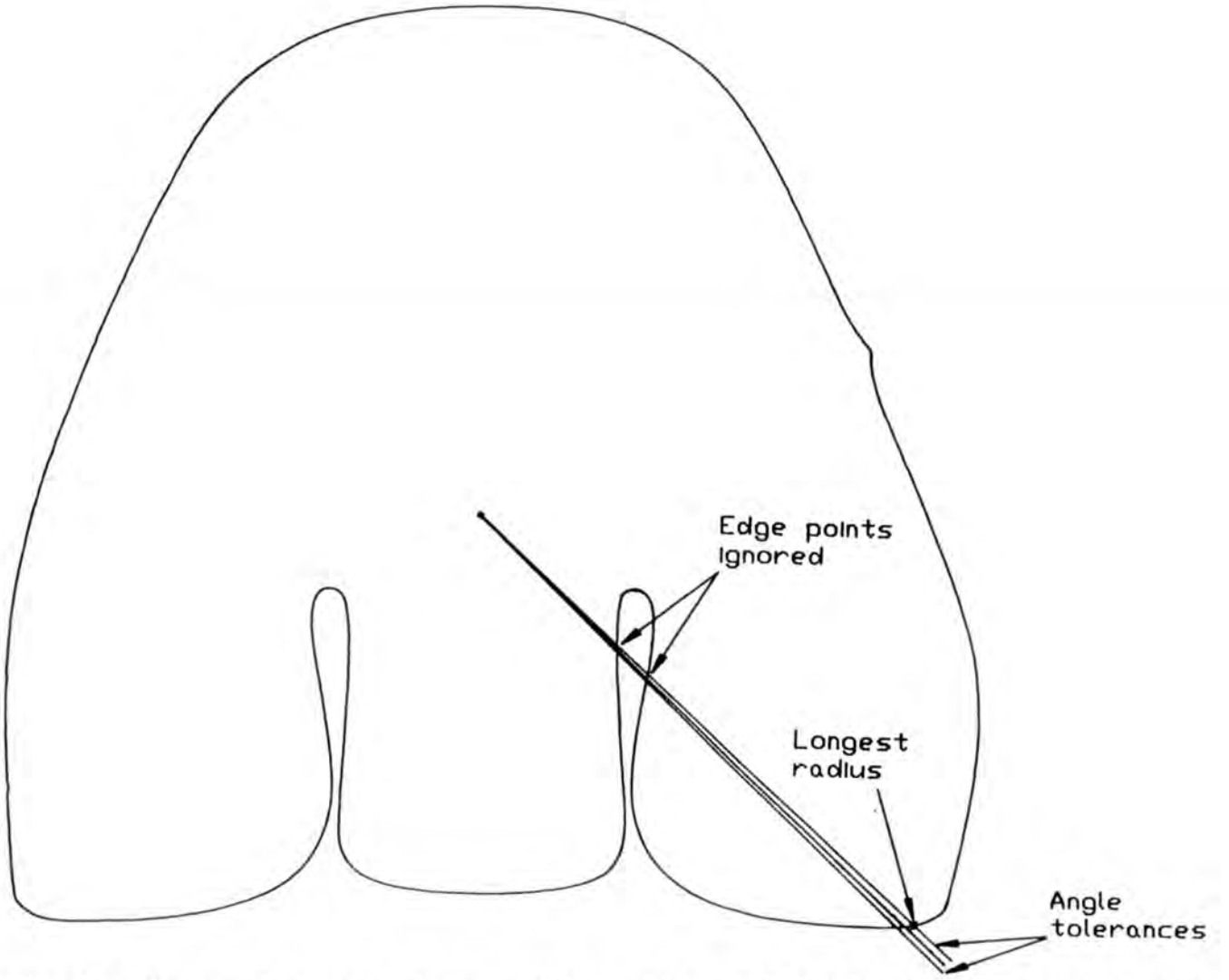


Figure 5.3 *Diagram showing the selection of a positive feature radius, and how multiple edge intersections are ignored.*

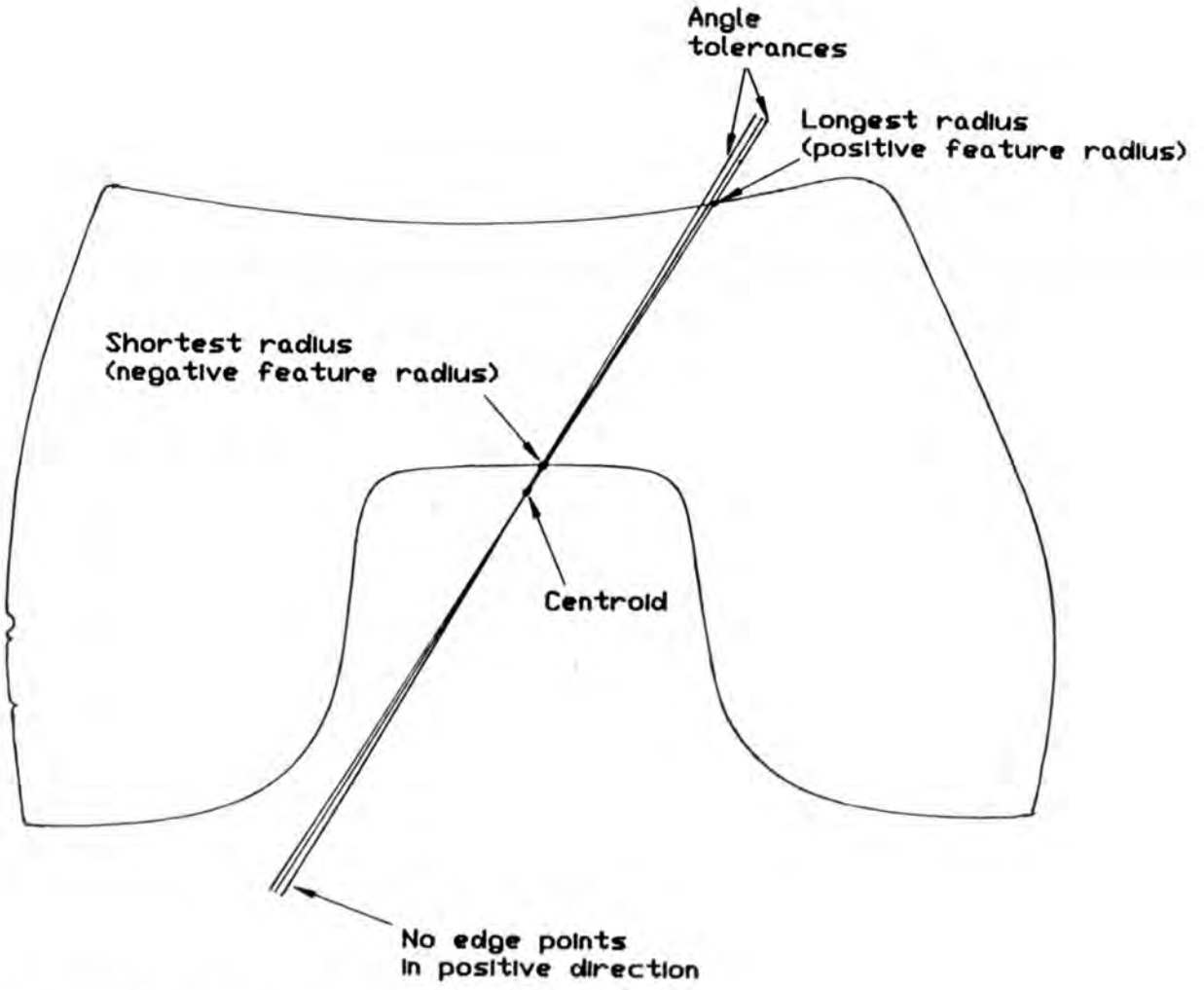


Figure 5.4 *Diagram showing the selection of a negative feature radius.*

'Teach' mode. This involves entering the class name of the component into the computer, which then causes the component to be scanned and its features extracted. The features of the shape are then stored against the class name in a temporary file, along with the same details of any other shape being taught. When all the new components have been taught another program is run which takes the sets of features from the temporary files and inserts them in the relevant places in the permanent database files, which are arranged in shape area order.

If the shape is a 'bad' shape then besides the standard features a further 19 sets of the 18 radii, starting at 1° intervals, making a total of 360 radii, are calculated and stored. These are used by the polar coding technique for determining the relative orientations of 'bad' scanned shapes, as will be described in section 5.8.

5.7 Database

The file *'GOOD.FTS'* has the standard set of features of all the shapes, 'good' and 'bad', while the 20 sets of radii of all the 'bad' shapes are stored in the file *'BAD.FTS'*. These files are stored on a Winchester disc as follows :–

i) *GOOD.FTS* – Holds the features of all the taught components, including the principal set of each 'bad' shape, stored in order of increasing root area. For each component the features are stored as shown in table 5.1.

ii) *BAD.FTS* – Holds the features of the taught 'bad' components. Each 'bad' component has 20 sets of the features shown in table 5.1 stored, with the starting direction of the feature radii in successive sets being increased by 1°.

The storage requirement for each 'bad' shape is thus 20 × 58 bytes = 1160 bytes, although some of the information in each set is duplicated and is thus redundant.

Feature	Storage type
Root–area	16–bit integer
4th. root of the maximum 2nd. moment of area	16–bit integer
4th. root of the minimum 2nd. moment of area	16–bit integer
18 feature radii	18 × 16–bit integers
Class name of the component	12–character string
Orientation of the taught component	32–bit fl–pt.
Total storage requirement for each 'good' component	58 bytes

Table 5.1 *Storage requirements for 'good' shapes.*

5.8 Feature matching

Feature matching is performed by a program running on the Compaq computer which receives the extracted features of the scanned shape from the slave Z8000. When an unknown component is scanned by the machine, recognition is attempted by comparing its calculated features with those of the previously taught components in the machine's database. A match between corresponding features is registered if they lie within preset tolerances of each other. The root–area and the principal 2nd. moments of area are used in a binary decision tree, as described in section 2.3.5, to filter out the majority of shapes and only allow through a small number for radii matching. This recognition technique is accelerated by storing the shapes in the database in order of increasing area. Then, only that part containing shapes whose root–areas match, within tolerance, that of the scanned shape is searched. While the root–area is within tolerance the principal 2nd. moments are checked to see if they too match within tolerance. If there are no shapes in the database with root areas and second moments which match within tolerance those of the scanned component, then that component is rejected as unrecognisable.

If the principal 2nd. moments match, then the radii of the scanned and each candidate shape which has passed through the binary decision tree are compared to see how closely they too match. However, the 'good' and 'bad' shapes have to be treated differently, as detailed below, since the orientation

will not be specified sufficiently reliably for 'bad' shapes by the principal axis.

For a scanned 'good' shape (i.e. GCOEFF > the threshold) the number of radii which match, within tolerance, those of each candidate database shape is determined. For each candidate database shape, if the number of matching radii is greater than in the previous best match, then the details of that database shape are recorded.

On the other hand, if the scanned shape is 'bad' (i.e. GCOEFF < the threshold) then, as explained in section 5.5, the orientation as given by the principal axis is not sufficiently reliable for printing stitchmarking. A polar coding and correlation technique is then used in which radii are matched and the orientation is determined simultaneously. All 20 sets of radii of each candidate database 'bad' shape are treated as a 360 radii superset. All possible sets of 18 equispaced radii within this superset are compared with the 18 radii of the scanned shape, and details of the best match and its orientation are recorded.

When the features of all the shapes in the database within the area tolerance have been compared then the number of radii within tolerance of the best matching candidate shape is examined. A small number of non–matching radii is allowed for a scanned shape to be recognised as a candidate shape. If the number of radii which do not match is less than the limit allowed then an exact fit is signalled. If the number of radii which do not match is greater than the limit allowed then only a 'best fit ' is signalled, which is taken to be a rejection.

If the component is recognised, the rotation of that component compared with its orientation when taught, is calculated. The rotation of a scanned 'good' component is determined from the directions of the major principal axis of the scanned and taught components. However, the rotation of a scanned 'bad' component is calculated using parabolic interpolation between the best fitting set of radii of the taught shape and the set of radii on each side.

5.9 Stitchmarking data

During the teaching procedure, after the features have been passed from the Z8000 systems to the Compaq, the edge coordinates of the shape are downloaded too and the shape is scaled and displayed on the vdu screen. Crosses through which stitchmarking lines are to be drawn using cubic interpolation are placed within the outline of the image of the shape on the screen by using a 'mouse' input device. The crosses can be dragged around using the mouse, as necessary, to produce a trace through the required path. As many stitchmarking lines as needed can be drawn on the shape.

When the stitchmarking lines have been drawn as required, the coordinates of the taught points are saved in the database in a separate file whose identifier includes the class name of that component. For example, if *XXXX* is the class name of the component being taught then its stitchmark data file is *XXXX.STM*. To reduce computation during stitchmarking the coordinates of these points are stored relative to the centroid and orientation of the taught shape as scanned and are converted to the coordinate system of the printer. Also stored are the coordinates of the dummy points, which are required for the interpolation, before the first point and after the last point of each stitchmarking line. This also saves a lot of recalculation every time that class of component has to be printed.

Each stitchmarking file is arranged in the format shown in table 5.2. This has been designed to work with a variety of interpolation methods, or no interpolation at all, as flagged by the interpolation code number. However, only cubic interpolation is available, and is designated by the code number 1. For cubic interpolation the coordinate pairs x[0] & y[0] and x[n+1] & y[n+1] are the dummy points for the interpolation. The 'Null' interpolation code indicates the end of the set of stitchmarking data.

When the machine is running in stitchmarking mode the class name of the component that has been scanned and recognised is passed to a routine which generates the name of the file in the database in which its stitchmarking data is stored. As shown in the flowchart of fig. 5.5, the routine then reads the data from the database and loads it into the memory of the Texas In-

Code for the type of interpolation to be used for the next set of points
Number of coordinate pairs in this curve (n)
X–coordinate[0]
X–coordinate[1]
...
X–coordinate[n+1]
Y–coordinate[0]
Y–coordinate[1]
...
Y–coordinate[n+1]
Code for the type of interpolation to be used for the next set of points
Number of coordinate pairs in this curve (n)
X–coordinate[0]
X–coordinate[1]
...
X–coordinate[n+1]
Y–coordinate[0]
Y–coordinate[1]
...
Y–coordinate[n+1]
...
etc.
Null

Table 5.2 *Format of the stitchmark files.*

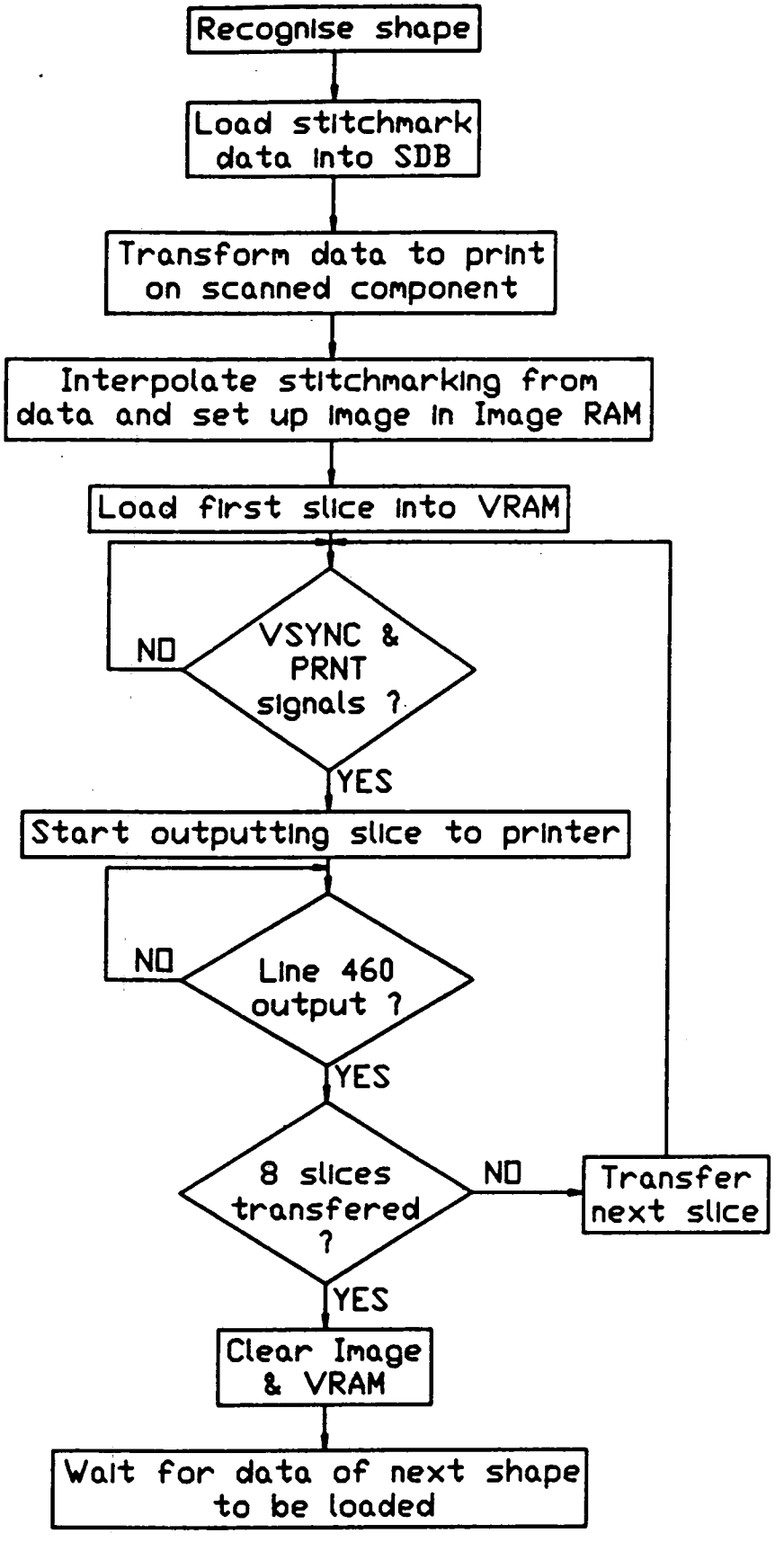


Figure 5.5 *Flowchart of stitchmark generation and printing.*

struments TMS34010 Software Development Board (SDB). The coordinates of the centroid and the length of the scanned component are converted from the camera coordinate system to that of the printer, and are also sent to the SDB.

On the SDB the stitchmark coordinates have to be transformed so that they are aligned for printing in the correct place on the component, as depicted in the previous diagram fig. 4.5. Since they are stored already translated relative to the taught shape's centroid, only a rotation and a translation are required. The rotation bringing them to the orientation of the scanned shape and the translation moving them to print in the correct position. The coordinates after rotation through a clockwise angle θ are given by [61]

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

and after translation by distances T_x in the x direction and T_y in the y direction are given by

$$x' = x + T_x$$

$$y' = y + T_y$$

The stitchmark image is then generated by performing a cubic interpolation through the stored coordinates, and setting up the result as a pattern of 1's and 0's in memory on the SDB. As explained in chapter 4, due to limitations in the amount of Video–RAM (VRAM) on this board, the stitchmark image is first generated at low resolution in general purpose RAM, and is split up into 8 slices.

From the length of the shape along the conveyor belt, the number of steps to the start of printing is calculated and is passed to the Master Z8000 microprocessor which controls the conveyor. As the scanned shoe component passes through the printer each slice of the low resolution stitchmarking image is then expanded and loaded in turn into the VRAM for printing under the control of synchronisation pulses from the Master Z8000.

The programs for generating the stitchmarking are listed in appendix B, and the synchronisation process is described more fully in section 5.11.

5.10 Conveyor control

The conveyor belts have to be driven at a speed which matches the scanning rate of the camera, otherwise the image of the component would be distorted. To synchronise the movement of the belts with the scanning, the 'Start/End of scan' pulses originating from the camera causes interrupts to the conveyor–control microprocessor. The interrupt service routine of the microprocessor outputs a step to the conveyor belt and print–drum drive motors, so that for each scan there is a step.

The microprocessor also synchronises the start of the printing process with the arrival of the component at the printer. Firstly, the number of steps of the belt to move the component to the start of printing from where it stopped after scanning is determined by subtracting the length of the component from the distance between scanning and printing. Then, these steps are counted as they are issued, and a 'PRINT' signal and 'VSYNC' (Vertical Synchronisation) signals are sent to the printer at the correct times. Statuses giving information about the printer are also monitored, and the stitchmarking system is shut down if any fault conditions occur.

5.11 Printing

The adaptation of the Delphax print engine to print on the shoe components passing through the machine is described fully in chapter 4. The resulting relationship between the coordinate systems of the camera and the printer is represented in fig. 5.6. Fig. 5.7 shows a diagram of the print mechanism with the main mechanical change which is the replacement of the pressure transfer of the toner by electrostatic transfer.

The printer operates by the generation of an electrostatic image on the print drum by the ion cartridge. The image is made visible by the development station where the coloured toner adheres by electrostatic force. The toned image is transferred to the shoe component using an electrostatic transfer stage, after which any toner remaining on the drum is scraped off. Residual charge on the drum is neutralised by the erase section. A detailed description of the operation of an ionographic printer is given by Rumsey and

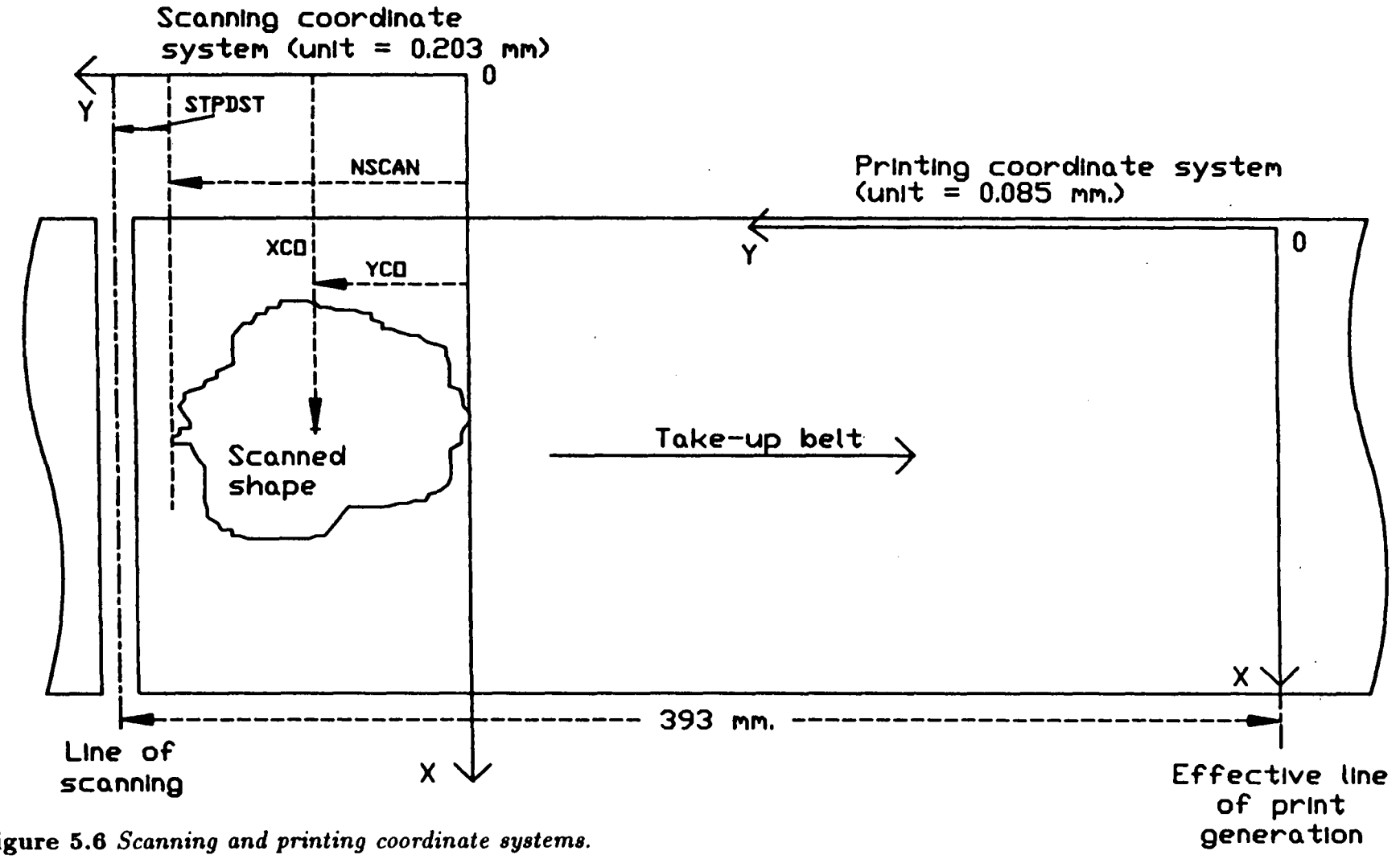


Figure 5.6 *Scanning and printing coordinate systems.*

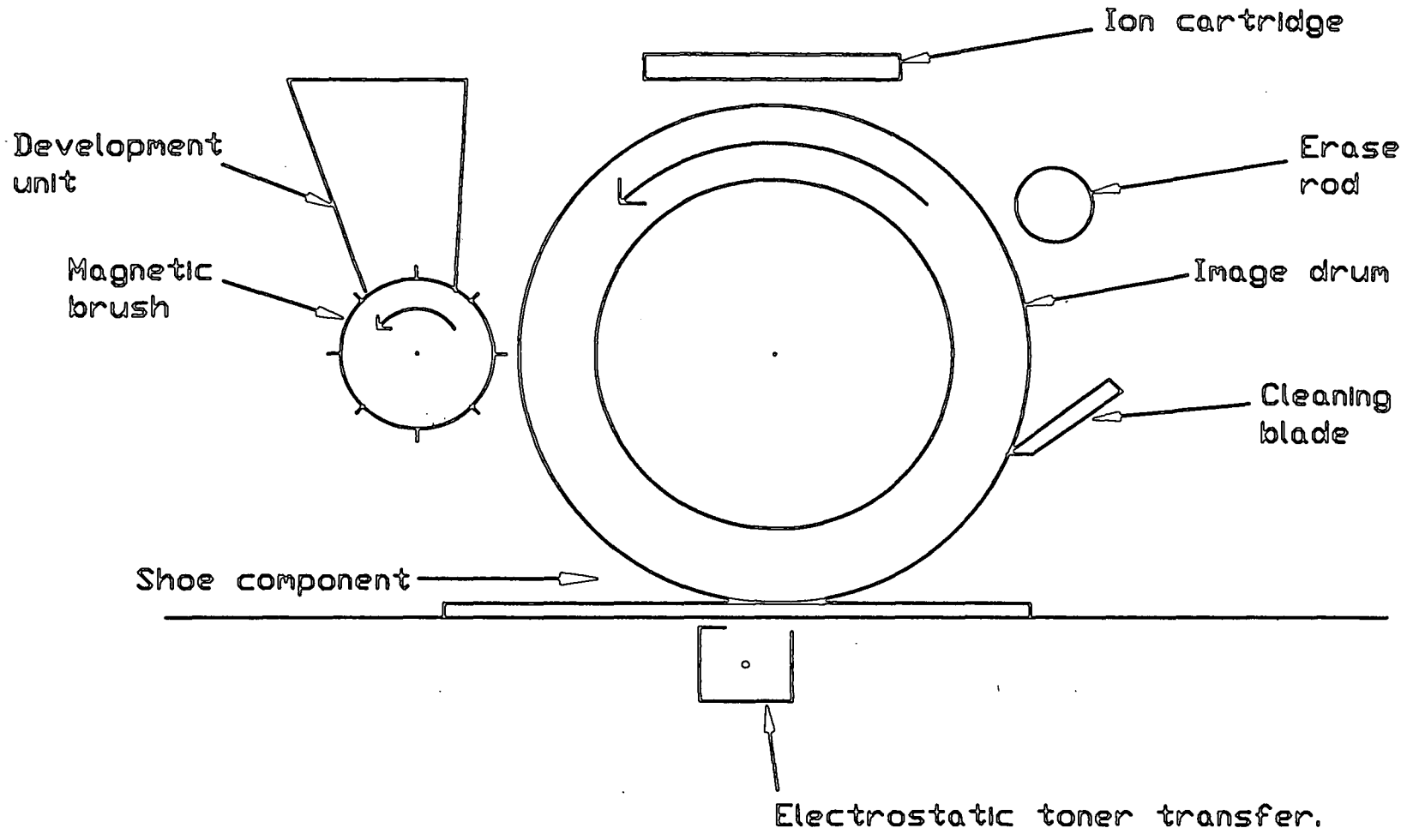


Figure 5.7 *Diagram of the modified ionographic printer mechanism.*

Bennewitz [56], though a summary and details of the electrostatic transfer stage are given here.

The ions to form the charge image are produced by the ion cartridge. This consists of a multi-layer structure of three sets of electrodes, radio frequency (rf), control, and screen, separated by insulating layers. The 20 sets of rf electrodes and 124 control electrodes form a multiplexed matrix with 2400 intersections producing ion generation cells across a width of about 200 mm. A diagram of a section of the ion cartridge is shown in fig. 5.8 where the arrangement of the 20 rf. electrodes in lines of cells down the cartridge and some of the 124 control electrodes in fingers across the cartridge can be seen. The ion generation cells occur where the control and finger electrodes cross forming groups on diagonal lines. The screen electrode is not shown since it covers the whole of the area shown and is connected permanently to all of the cells. Image formation with a resolution of 11.8 dots/mm (300 dots/inch) is performed by imposing electrical signals on the appropriate combination of rf. and control electrodes to release charge from the correct cells as the image drum rotates underneath.

The principle of operation is the generation of ions with both positive and negative charges by a corona discharge produced by a high frequency, high voltage signal across a capacitor with a hole in one electrode. This is the control electrode of figs. 5.9 & 5.10. When required, the negative ions are gated by a superimposed DC voltage between the screen and control electrodes, are focused by the screen electrode, and are accelerated towards the dielectric layer by the high potential gradient.

In operation the rf electrode lines are successively excited by a high voltage 1 MHz burst controlled from the rotation of the image drum by a shaft encoder. This rf energy is applied between the drive electrode and the control electrode of each cell, and produced a pool of positive and negative ions by glow discharge. In the ungated state, shown in fig. 5.9 with the electronic control switch in position A, the negative ions are constrained within the cell by the negative potential of the screen electrode with respect to the control electrode. The positive ions are constrained by the high positive potential of the image drum with respect to the cell.

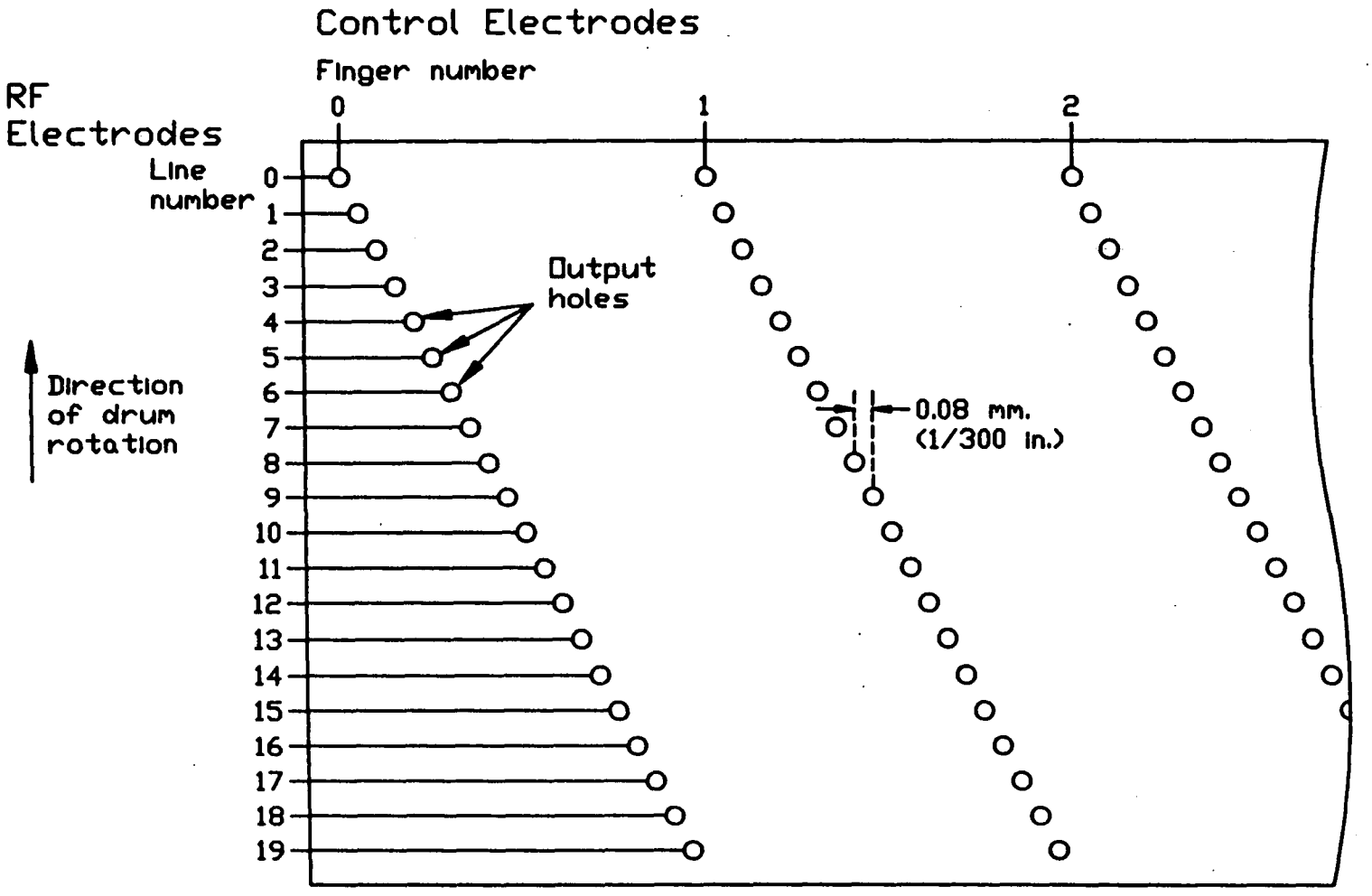


Figure 5.8 *The electrode and cell arrangement of the ion cartridge.*

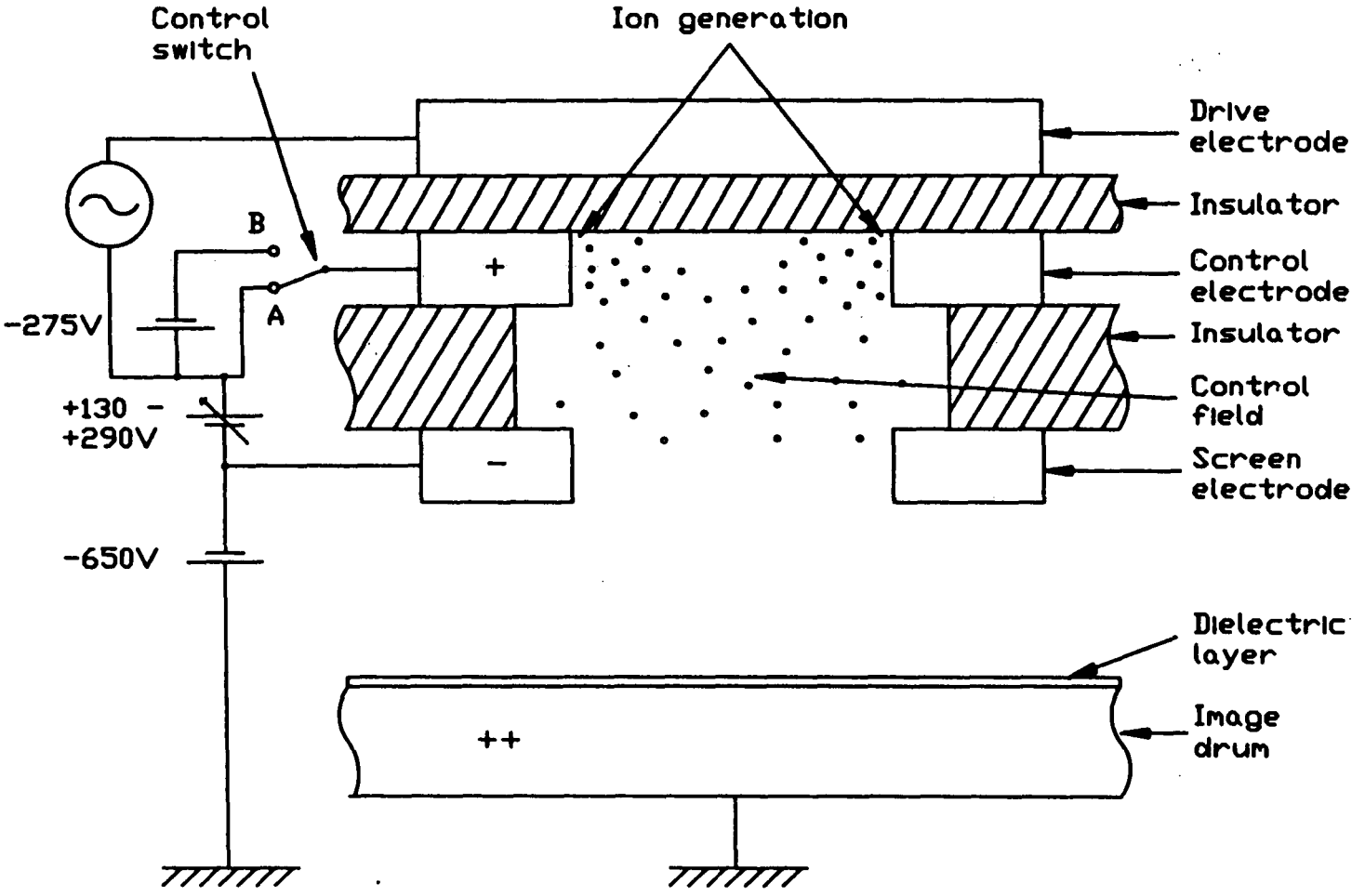


Figure 5.9 *Ion generation cell in the ungated state.*

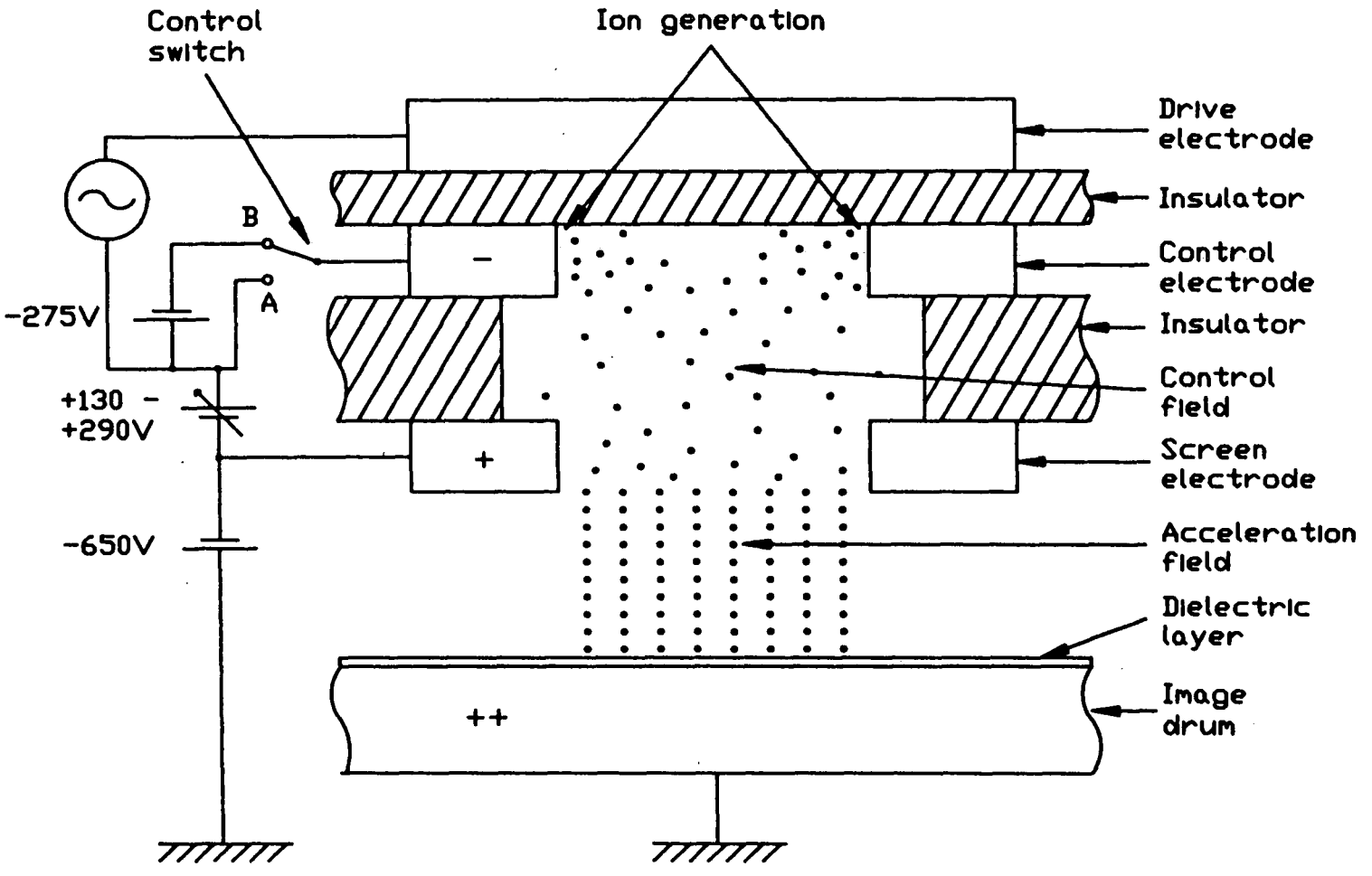


Figure 5.10 *Ion generation cell in the gated state projecting ions.*

When it is necessary to release the charge to the image drum the electronic switch introduces the additional DC source between the control and screen electrodes, as represented by position B in fig. 5.10. This brings the potential of the screen electrode close to that of the control electrode or even makes it more positive. Although some of the negative ions are captured by the screen electrode, a large proportion escapes from the cell and are accelerated to the image drum by its high relative positive potential.

The image drum is made of aluminium and has a hard, dielectric coating about 40 μm thick produced by an anodising process. This layer has an electrical time constant sufficiently long for the charge to be retained, with a surface potential of about -180 V, for toning and transferring the image. The drum is heated to about 62°C to prevent the condensation of by-products of the corona discharge which would increase the conductivity of the dielectric layer.

The charge image is made visible by dusting it with fine coloured toner powder at the development unit. This makes use of the magnetic brush principle by which the single component magnetic toner forms brushlike bristles on the surface of a sleeve surrounding a cylindrical assembly of magnets, as shown in fig. 5.11. The magnet assembly remains stationary while the outer sleeve rotates, carrying the toner which is brushed against the charge image on the drum. The toner is conductive so that individual grains close to a charged area on the dielectric layer became charged by induction, as depicted in fig. 5.12, and are retained on the drum by the resultant electrostatic attractive force.

The image drum revolves carrying the toned image to the transfer station where it meets the shoe component which is to receive the image. Here a negative potential with respect to the image drum of about -6.5 KV is applied to the corona wire of the corotron, which consists of a corona wire surrounded on three sides by an earthed screen with an opening through which charged particles flow. The corona current produces a negative charge on the underside of the conveyor belt which induces a charge separation in the toner particles of the image in the opposite sense to that holding them to the drum, as depicted in fig. 5.13. The toner is thus attracted from the drum

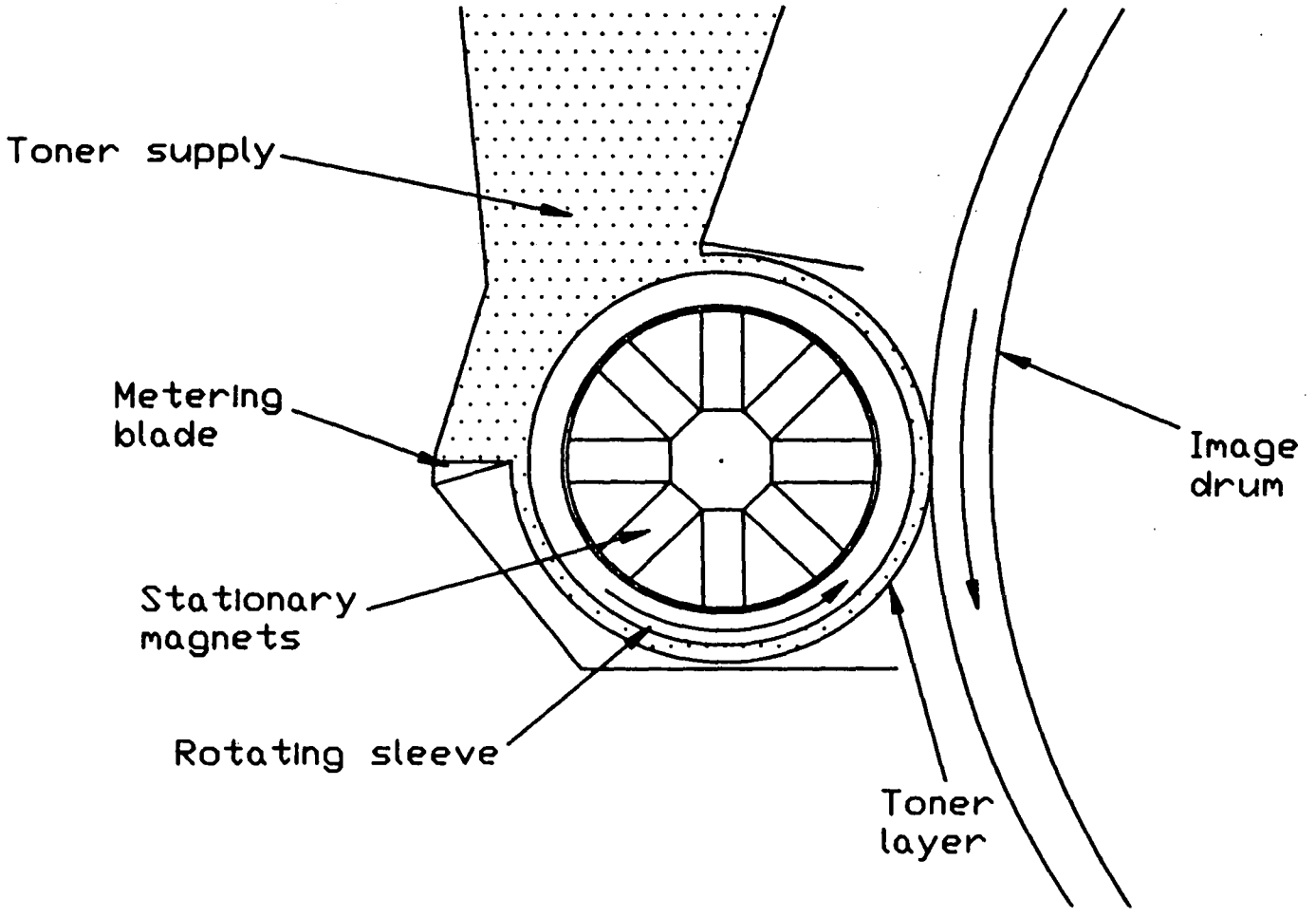


Figure 5.11 *Magnetic brush development unit.*

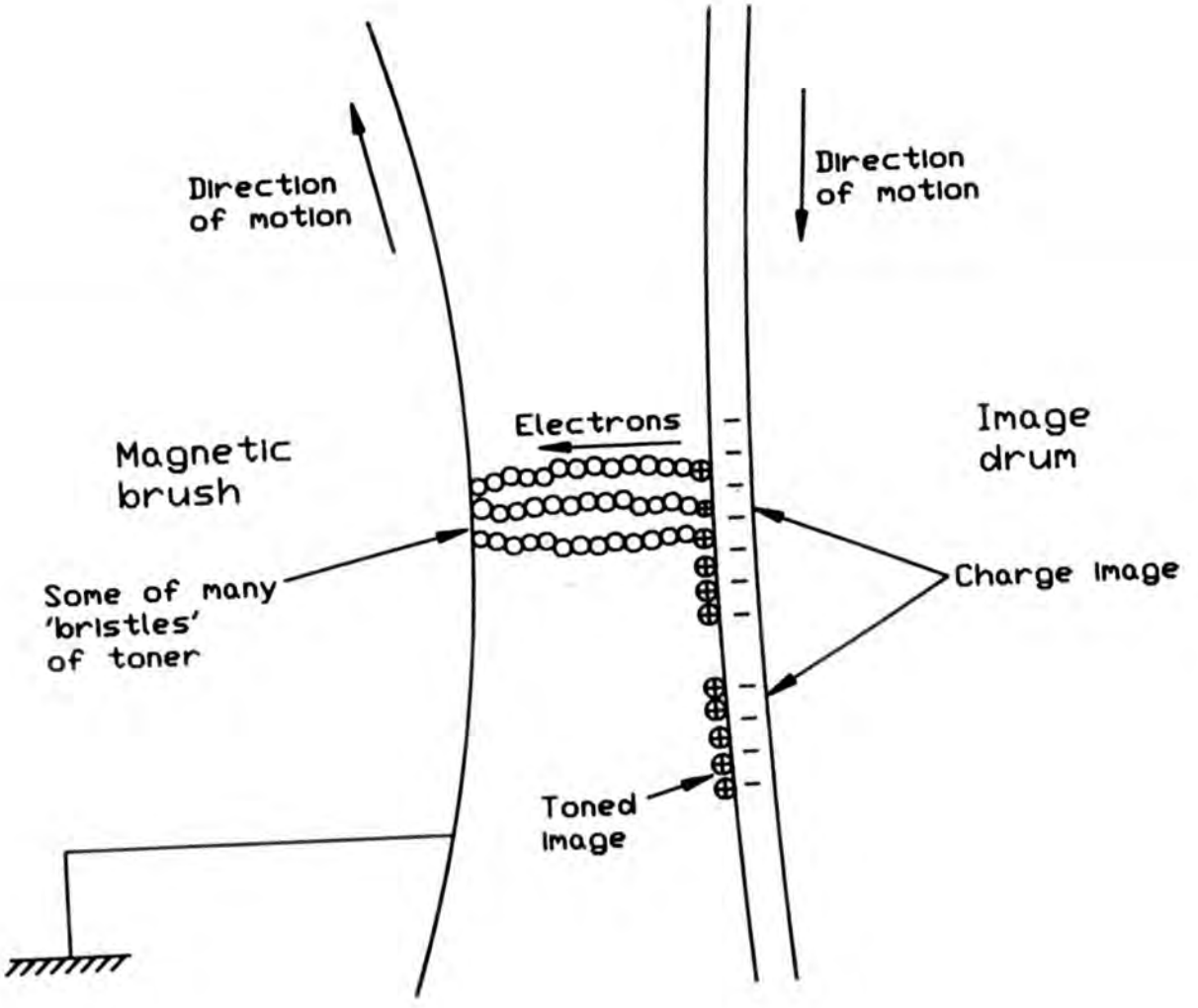


Figure 5.12 *Toner development mechanism.*

towards the corotron but is intercepted and held by the intervening surface of the shoe component being printed. Toner is in fact deposited at the bottom of the rough grain pattern of leather as well as on the peaks where the leather touches the toner on the image drum. The charge on the underside of the conveyor is discharged by the earthed metal drive rollers. The construction of the corotron is based on that of the Hewlett Packard laser printer, which was found to perform well. It replaced the printer's original pressure transfer stage, as explained in chapter 4.

The toner is carried away from the drum on the shoe component to which it can then be partially fused by radiant heat. However, no permanent toner fusing stage is used, since it is more convenient to be able to wipe the toner straight off the limited number of shoe components available so that they can be reused.

Any toner remaining on the drum is mechanically removed by a steel blade and is caught in a tray. Charge remaining on the drum is neutralised by the erase rod, the operation of which is described in detail by Landheer & Devitt [66]. This consists of a glass rod along which are mounted four equispaced glass coated wires. A wire helix is wound around the assembly to form many corona intersections with the glass coated wires. By applying a signal of about 2300 V peak-to peak at 1.15 MHz between any of the glass covered wires and the helical winding a copious supply of ions of both polarities is produced. With the glass coated wire spaced from the image drum by 100 μm, the residual charge on the drum is reduced to less than plus or minus 15 volts. The drum is then ready to receive a fresh image.

As shown previously in figs. 4.7 & 4.9, the printer's Format Controller board, which provided a Centronics interface and produced a raster graphics output from the ASCII format input, was discarded and replaced by the TMS34010 software development board (SDB). This outputs directly to the de–skew board, which is described below. The functions of the Engine Control board, which provided overall printer control and monitoring, are taken over by the Z8000 Master microprocessor.

In the original printer the pressure roller normally used for transferring

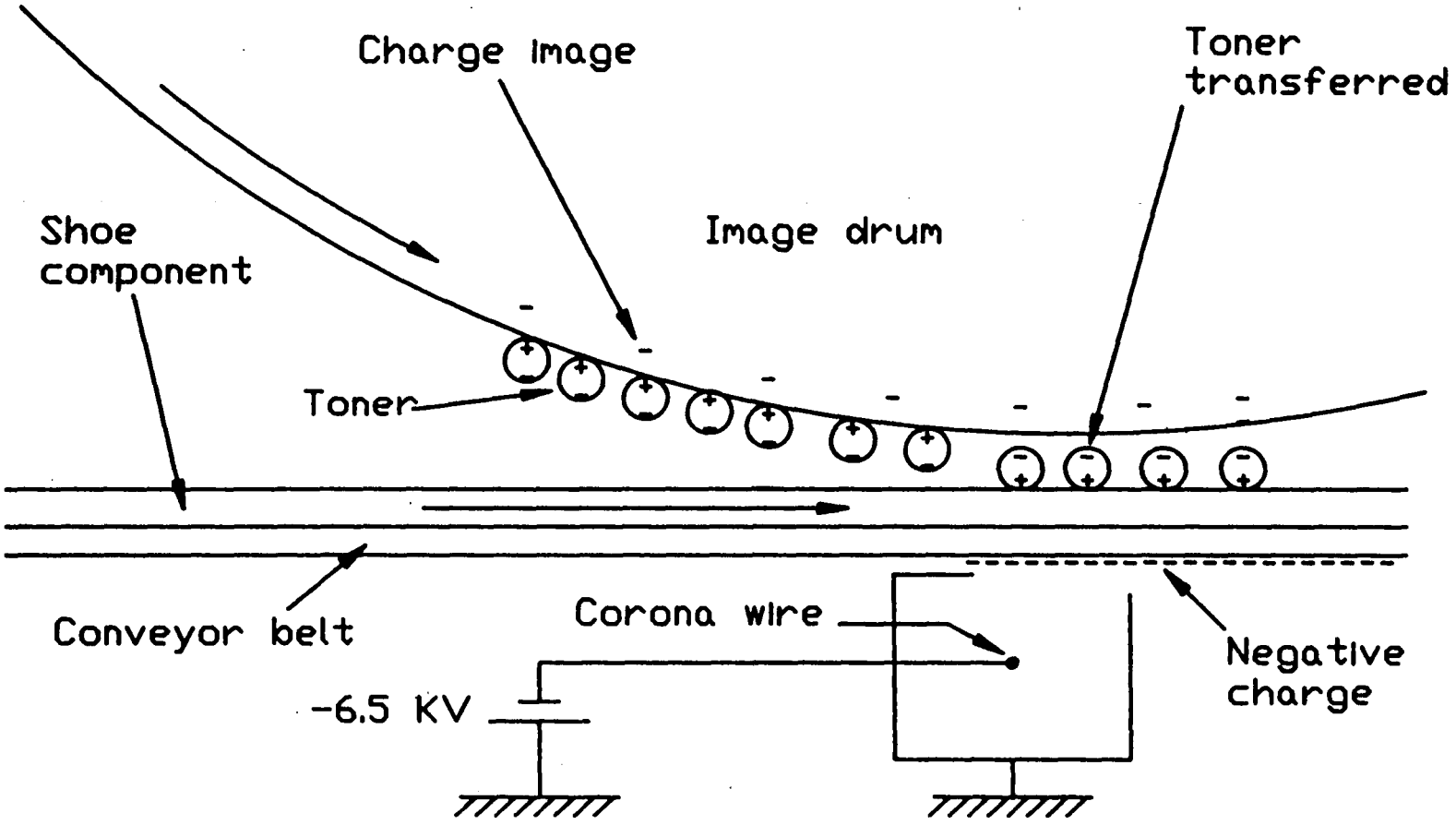


Figure 5.13 *Toner transfer mechanism.*

and fusing the toner was mounted at a small skew angle to the image drum to maintain even pressure along its length. This caused skewing of the resulting printing on the paper. The de–skew board is so called because one of its functions is to apply a correction to the printing so that it is not skewed when transferred to the paper. This process is not now required, and so it is bypassed. However this board also received the raster data to be printed and converted it to the form required by the Line Drive and Finger Drive boards of the print engine shown in fig. 5.14. The de–skew board will accept either serial or 8–bit parallel data, but in this case serial data is used since it is more compatible with the output of the SDB.

Transmission of the serial image data is controlled by vertical synchronisation (VSYNC) and horizontal synchronisation (HSYNC) pulses in a similar way to those in the operation of a vdu. In this case a VSYNC pulse signifies the start of a new page, and an HSYNC pulse signifies the start of a new line. This is described in more detail in appendix A. During printing the Z8000 Master microprocessor generates a VSYNC pulse every 213 steps of the conveyor belt. This distance corresponds to the printing of the 512 lines of print data held in the VRAM. The VSYNC pulses are passed through the de–skew board, where they control its operation, and on to the SDB. The TMS34010 on the SDB is programmed to work in external synchronisation mode, so that its video output is controlled by the arrival of VSYNC and HSYNC pulses. Each VSYNC pulse then resets the output from the VRAM to line number 0, and each HSYNC pulse resets the output to pixel number 0 on the next line.

The HSYNC pulses are generated by the de–skew board in response to the VSYNC pulses from the Z8000 Master and the rate of printing, as determined by the input from the shaft encoder on the image drum. Between HSYNC pulses enough data for one line of 2400 pixels is transmitted from the SDB to the de–skew board, but the actual time between the pulses varies with the speed of the image drum. The de–skew board generates a dot clock (DOTCLK) at 10 MHz, which is sent to the SDB to synchronise the transmission of the image data. The data returned to the de–skew board is latched by the low–to–high transition of the return clock (RCLK), which

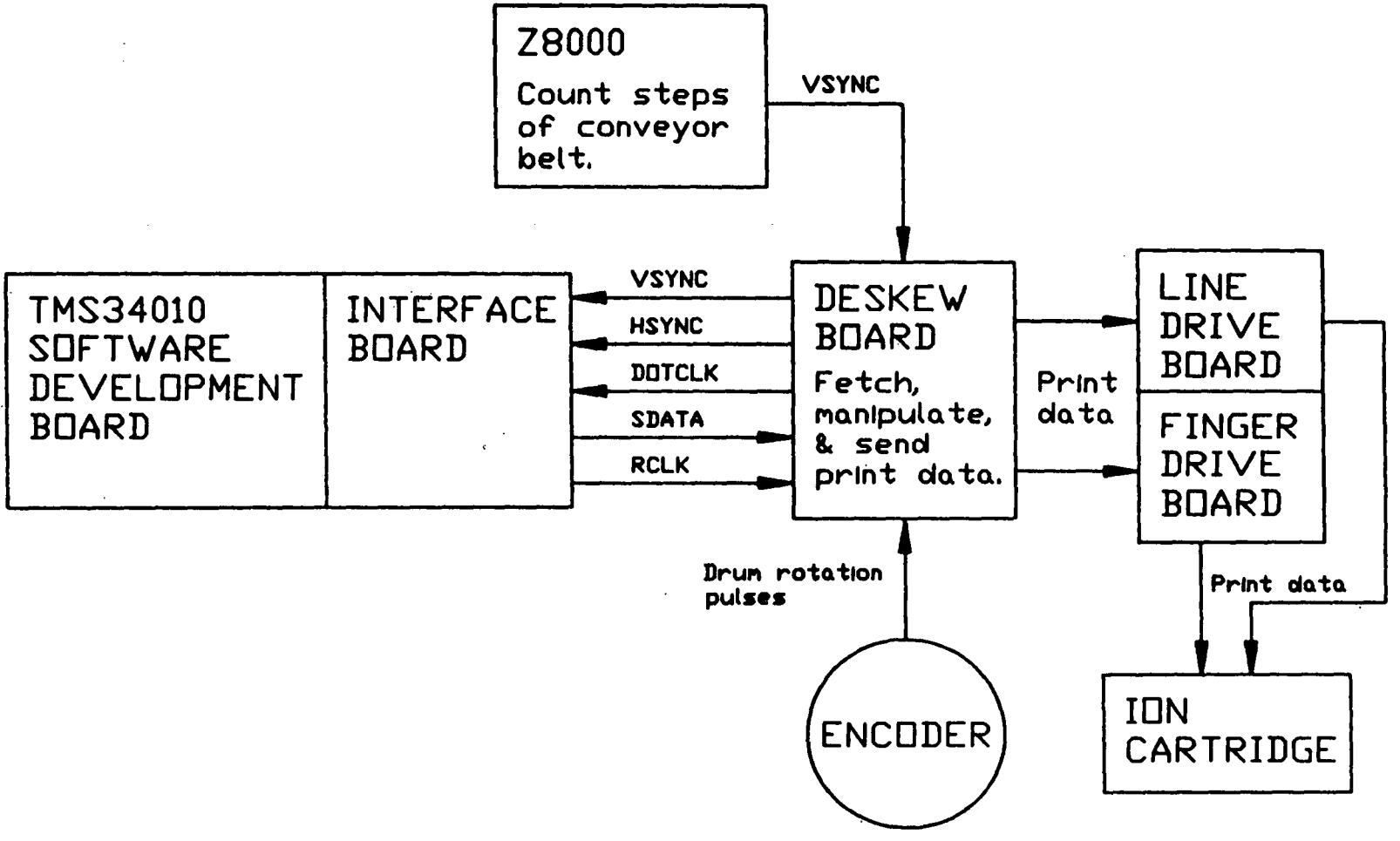


Figure 5.14 *Printing synchronisation paths.*

is derived from DOTCLK by a specially designed interface board connected between the SDB and the de–skew board. The main purpose of this interface board is, however, to convert the 4–bit parallel output of the SDB to the serial input required by the de–skew board. This interface board is further described in appendix A.

As described in chapter 4, printing continues, with slices of the low resolution image being transferred to the VRAM, until the Z8000 Master has calculated that the whole image has been printed.

5.12 Overlapping the functions

An overall diagram of the processes which are performed during the recognition and printing of a shoe component is shown in fig. 5.15.

All of these individual processes have to be completed on each component being stitchmarked before the next one can be dealt with. The processes have been designed so that they could be overlapped with the minimum of alteration to increase the throughput of the machine by having several components in process simultaneously. However, this is not a facile alteration, as will be seen from the discussion of section 8.4, and is best left until the individual processes are optimised.

5.13 The exhibition machine

At this point it is valuable to describe the differences between the research machine as just described and the machine developed from it at BUSM for exhibition purposes, which was mentioned in section 4.6.

The electronic hardware and the control software of the exhibition machine are essentially identical to that of the research machine. The major difference is in the reversion to conveyor belts perforated along the edges. These are used since it is required to have sufficient spacing between scanning and printing for the shoe components to be recognised without stopping the belts. Toothed belts of the necessary length and width are not readily obtainable. The belts are made of polyurethane coated polyester fabric, and

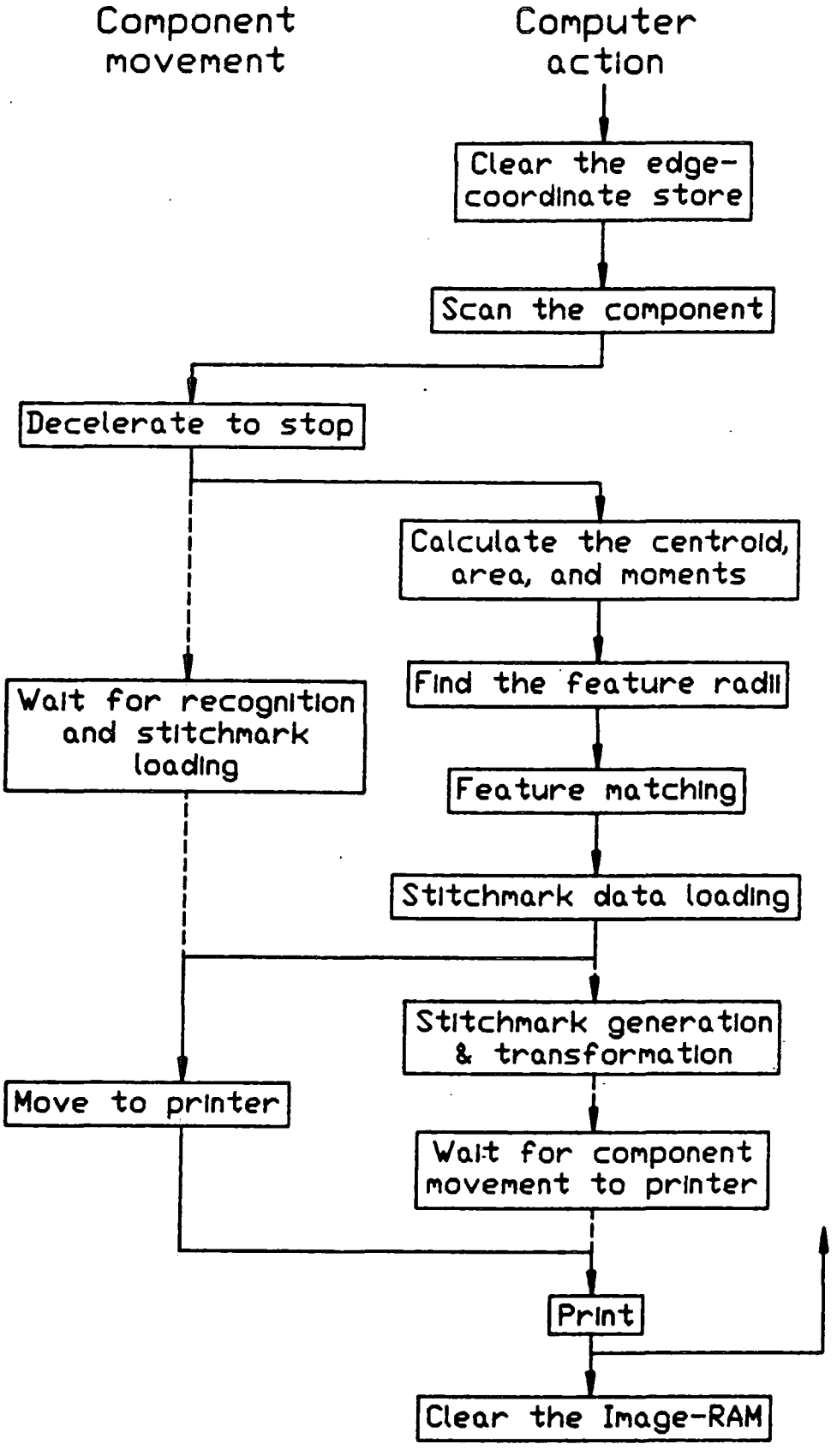


Figure 5.15 *Flow chart of the stitchmarking process.*

are of the full width of 400 mm. They perform very well, and only developed elongation of the perforations after many hours of use.

The shoe components travel through a toner fusing stage after printing. By the use of efficient reflectors, this requires only two 500 mm long high intensity tungsten halogen lamps across the belt at a few millimetres spacing to produce the required degree of fusing. After the fusing stage there is a 10 station collator so that similar components can be collected together. The fusing stage and the collator require alterations to the software so that they operate correctly as the components reached them.

CHAPTER 6

ACCURACY EVALUATION

6.1 Introduction

This chapter deals with a theoretical and practical evaluation of the accuracy of the stitchmarking system. Each function of the system will be considered in turn.

6.2 Accuracy of the recognition system

The precision of aspects of digital vision systems have been considered theoretically by Ho [8]. Sources of error within the digital vision system include

i) relative motion between object and camera,

ii) parallax,

iii) distorted image, and

iv) digitising error due to discrete sensors on the image plane.

Another possible source of errors is the finite precision of the mathematics used for calculating the features, though the most appropriate methods were chosen when the recognition programs were written.

6.2.1 Distortion

Relative movement between the shoe component and the imaging device will cause distortion of the resulting image. The shoe component is actually moving while the electrical analogue of the image is being generated within the imaging device. The result will be a loss of resolution in the direction of conveyor movement and slight distortion in that direction. Any vibration of the camera will cause it to be pointing at the wrong part of the component being imaged. The effect is minimised by reducing the distance between the object and the camera, and by making the camera mounting as rigid as practical.

A shoe component away from the optical axis of the lens will present more of its sides to the sensor than a similar component on the optical axis, causing distortion of the image. This parallax error is reduced by increasing the distance from the object to the camera.

A distorted image can be caused both by imperfections of the lens, see section 2.2.3, and also by incorrect placement of the camera and lens. The shoe component moves past the camera at the rate of 0.203 mm per scan. In order for the component to be quantised on a square grid the camera has to be adjusted so that

i) the line array sensor is perpendicular to the direction of movement of the conveyor belts,

ii) the plane of the sensor is parallel to the plane of the conveyor belts, and

iii) the magnification is such that the digitisation interval across the components is also 0.203 mm.

In order to minimise distortion the conveyor mechanism and the camera and its mounting were finely set up using engineering instruments and a calibration program. All critical horizontal and vertical parts were adjusted using a spirit level, and all critical perpendicular angles were set using a large engineer's try square. This ensured that the camera was perpendicular to the conveyor, and that the drive shafts and idler rollers of the conveyor system

were perpendicular to the direction of motion. By examining the analogue output of the camera for any intrusion into the light beam, the scanning gap was made as narrow as possible and the line sensor array of the camera was aligned with the gap. The calibration program was then run and the height of the camera was finely adjusted to give the least distorted image of the calibration disc, as described in Browne & Norton–Wayne [4]. This occurred when the three conditions listed above were complied with.

Thermal expansion of the machine causes distortion of the image by altering the object and image distances and hence the magnification. An estimate of the effects of temperature changes on the magnification of the image was made by considering the lens equations

$$\frac{1}{V} + \frac{1}{U} = \frac{1}{F}$$

$$\frac{V}{U} = M$$

where

U is the separation between the object and the front principal plane of the lens.

V is the separation between the rear principal plane of the lens and the imaging device.

F is the focal length of the lens.

M is the magnification.

The linear array has a pixel spacing of 13μm, and since the digitisation grid at the component was designed to be 0.203 mm, the lens magnification required is $13 \times 10^{-6}/0.203 \times 10^{-3} = 0.064$. Since the lens has a nominal focal length of 58 mm, the above equations give $U = 964.25$ mm, and $V = 61.712$ mm. The actual figures for these parameters will differ slightly since the actual focal length of the lens will be different from its nominal focal length.

The camera system is represented diagrammatically in fig. 6.1, where use has been made of the fact that the camera is clamped to the support near the plane at which the sensor is held. Assuming that the camera box and lens

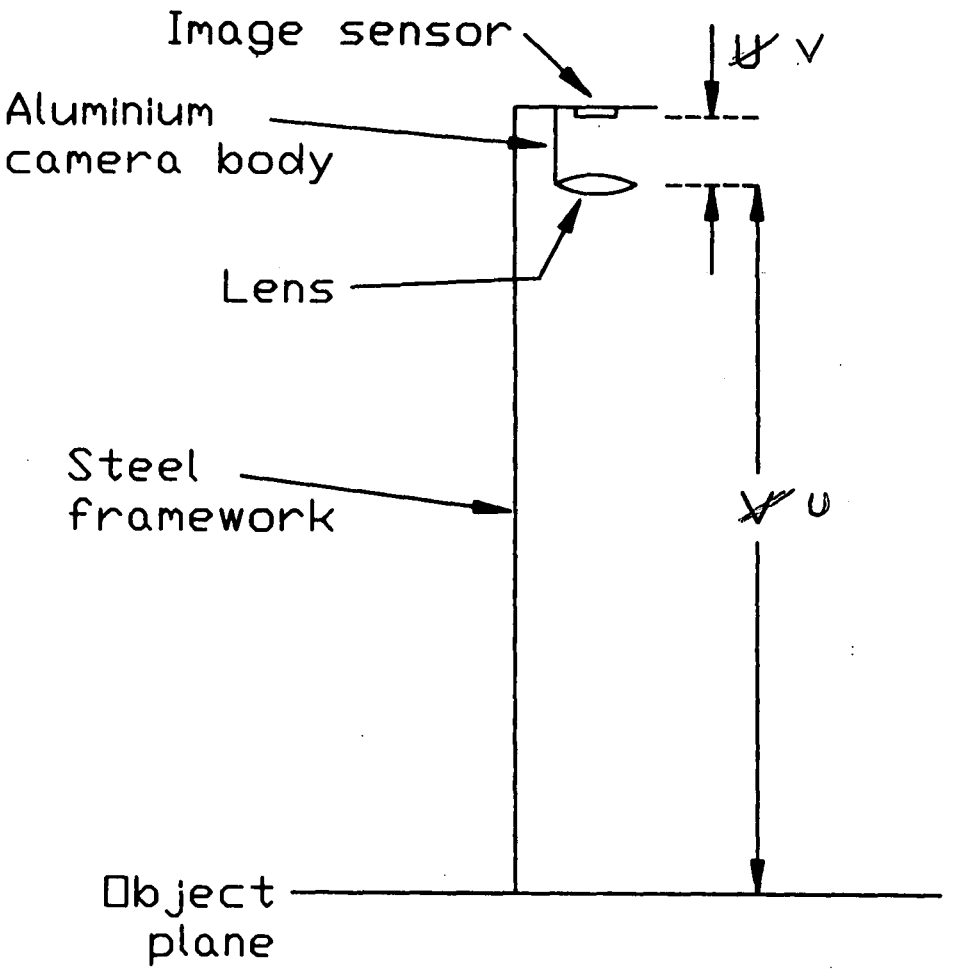


Figure 6.1 *Simplified diagram of the camera and its support.*

mount are made of aluminium with a linear expansivity of $23 \times 10^{-6}\ K^{-1}$, then V will increase by $61.712 \times 23 \times 10^{-6} = 0.0014$ mm per K temperature rise. U will increase due to the expansion of the mild steel support, and decrease due to expansion of the lens mount downwards. Since mild steel has a linear expansivity of $15 \times 10^{-6}\ K^{-1}$, this gives a change in U of $((964.25 + 61.712) \times 15 \times 10^{-6}) - 0.0014 = 0.014$ mm per K. Hence, a 10 K temperature rise will cause a change of magnification to $(61.712 + 0.014)/(964.25 + 0.14) = 0.064$, i.e. a negligible change.

6.2.2 Component position during scanning

Ideally, the shoe components would be rigid 2–dimensional objects. Such components would have no thickness and would pass over the scanning gap without drooping down or bowing up, which would introduce parallax and alter the object to lens distance.

The effect of thickness of the component being scanned was discussed by Koulopoulos [3], and a diagram based on his is shown in fig. 6.2. The ratio of apparent object width w' to actual object width w for an object on the axis of the lens is given by

$$w'/w = U/(U - t)$$

where U is the distance of the object from the lens and t is the thickness of the object.

For the stitchmarking system the object is 964 mm from the lens and the range of component thicknesses that have to be handled is 0.5 to 3.0 mm. The apparent width of objects of these thicknesses is hence increased over the actual width by 0.05 % to 0.3 % respectively. These differences are smaller than the likely minimum differences between adjacent sized shapes, but will contribute to the rejection and substitution rate of the recognition system.

If a thick shoe component is taught to the system, and is then scanned by the system in the same position across the conveyor and at the same orientation, then the two images will be identical. If the orientation of the component is altered, then the shape will alter since it is only the width that

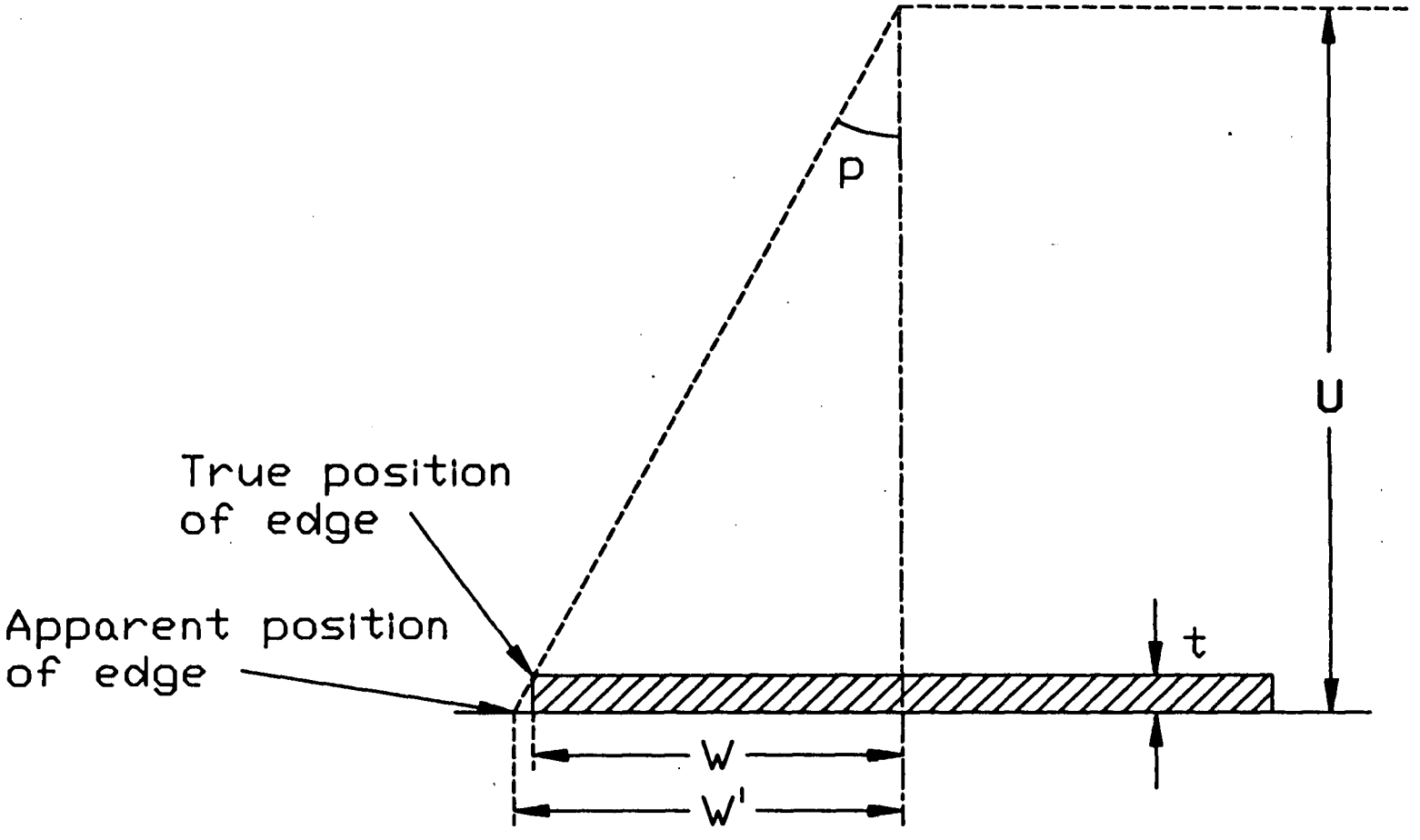


Figure 6.2 *Effect of finite thickness of the shoe component.*

has the apparent size change. If the position across the conveyor is altered then the apparent width of the component will also alter, due to parallax.

Thus the error caused by different thicknesses of shoe components and the variation between the thickness of taught and subsequently scanned components has quite a complicated effect. It could be reduced by calibrating the camera using a disc whose thickness is the median thickness of the leather being scanned, i.e. about 1.5mm.

Since the shoe components have to move across the scanning gap there is the possibility of them drooping down as they pass across it. This will increase the object to lens distance and cause the components to be imaged too small. To reduce the possibility of drooping as far as possible the size of the gap was adjusted to the minimum of 1.8 mm, consistent with the conveyor belts not intruding into the view of the camera. This intrusion starts to occur at the two edges of each belt where they bow up slightly as they pass round the drive rollers.

The factors affecting the degree of droop were determined by considering the shoe component to be a beam rigidly clamped between the top roller and the belt drive roller at the start of the gap. The deflection is at a maximum when the component has travelled across the gap and has just reached the take up conveyor belt, as shown in fig. 6.3.

The deflection (D) of a beam firmly clamped at one end is related to the properties of the material by [67]

$$D \propto \frac{W}{E \times I}$$

where

W is the load per unit length.

E is the Young's modulus.

I is the vertical second moment of area.

Thus the deflection will be greater for leather with greater density, lower stiffness, and smaller thickness.

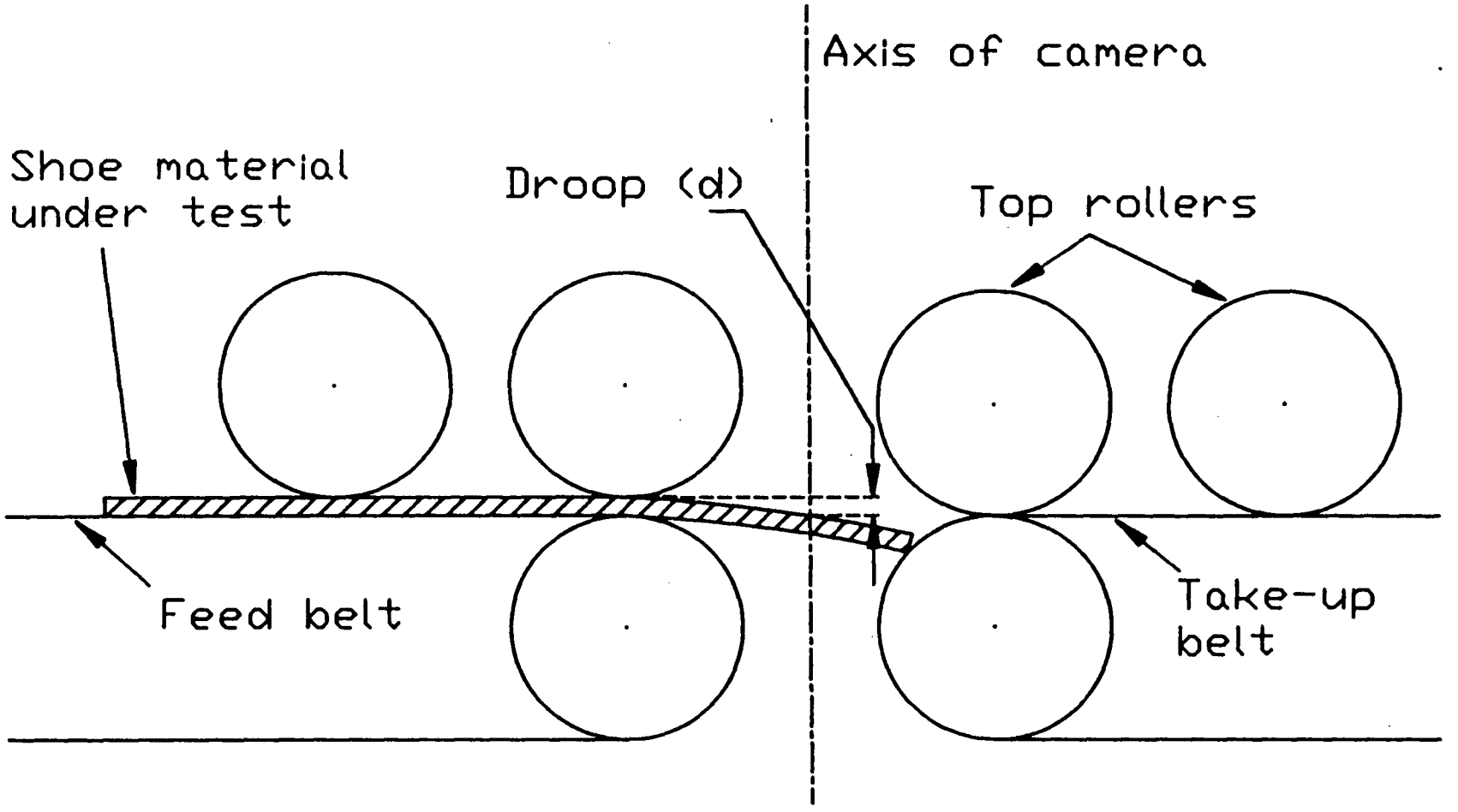


Figure 6.3 *Measurement of the maximum droop of shoe upper material as it passes across the scanning gap.*

Of the twenty different types of leather that had been obtained to represent the range of those used in shoe components the one with the most droop was selected for experiments. This was done by clamping one end of similar samples of each horizontally and selecting the one which drooped to the most vertical angle. The thicker, stiffer leathers showed negligible droop, but some of the thin supple leathers drooped considerably.

A 50 mm wide piece of the selected leather, which was 0.70 mm thick, was placed along the centre of the feed belt. The belt was moved forwards by turning its drive motor by hand until the leading edge of the piece of leather just touched the takeup belt. This made the droop of the leather at the centre of the scanning gap the maximum that it would reach. The droop was determined by measuring the difference in height of the component from where it passed over the feed belt drive roller to the centre of the scanning gap using a travelling microscope, as shown in fig. 6.3. The droop of this very floppy leather was measured to be 1.4 mm at the line of scanning.

A measurement of the droop when the leather was suspended between the feed and takeup belts was made by driving the belts and stopping them when the component was over the gap. The droop in this case was measured to be just 0.1 mm.

These measurements were repeated using a small shoe component cut from thin stiff leather, but which was bowed so that its centre was higher than its leading and trailing edges. This gave a droop of 0.05 mm and -0.7 mm (i.e. raised above the belts) respectively.

The effect of any droop or bowing is much more complicated than the effect of component thickness, since it has different effects at the start and end of a component than in the centre.

An experiment was performed to assess the effect of the droop of the worst leather on the actual features of shapes measured by the machine. To do this a template of a triangular shape was cut from thick card. This was used to cut out two identical shapes for scanning; one from the leather sample, with the greatest droop, and one from thin stiff card. Each shape was passed through the machine 30 times using varying arbitrary positions

Feature	Stiff card		Floppy leather	
	Range	Standard deviation	Range	Standard deviation
Root–area	1	0.51	3	0.59
Maximum 2nd. moment	1	0.35	2	0.53
Minimum 2nd. moment	1	0.25	2	0.32
Radius 1	3	0.75	9	1.84
Radius 2	3	0.86	4	1.12
Radius 3	2	0.57	3	0.62
Radius 4	1	0.51	3	0.68
Radius 5	1	0.51	3	0.63
Radius 6	2	0.66	3	0.76
Radius 7	5	1.22	5	1.47
Radius 8	5	1.28	8	2.14
Radius 9	2	0.55	5	1.36
Radius 10	1	0.49	4	0.99
Radius 11	4	0.82	3	0.99
Radius 12	6	1.56	9	2.22
Radius 13	4	1.07	8	2.01
Radius 14	3	0.82	3	0.73
Radius 15	1	0.49	3	0.63
Radius 16	1	0.49	2	0.74
Radius 17	3	0.63	2	0.63
Radius 18	4	0.94	3	0.86

Table 6.1 *Range and standard deviation of the features of similar shapes of stiff card and floppy leather.*

and orientations, and the resulting calculated features were recorded. From these values the standard deviation of each feature for the two cases was then calculated. The results are summarised in table 6.1, where the unit of measurement is the pixel.

As was expected, in the majority of cases the standard deviations of the features of the floppy leather were greater than those of the stiff card. The range of values that the features covered was also usually greater, and sometimes was beyond the range of the tolerance for matching that feature. This is the case with radius 1 and radius 12 where the range of values of 9 is greater than the +/− 4 allowed by the tolerance. So these radii would not match a class prototype on some occasions.

Several ways of reducing the droop are possible. One is to insert a shaped transparent rod into the scanning gap to support the components as they pass, as shown in fig. 6.4. However the basic principle behind using the gap was that there is nothing to keep clean and scratch free. Also, this would not help stiff components with a tendency to bow upwards. Another possibility which does not have these problems is an arrangement of shaped bars within the scanning gap, as shown in fig. 6.5. This would guide the components within fixed limits from the camera as they pass over the gap. Dust and dirt which may collect on the edges of the small gap and intrude into the field of view of the camera could be removed by a sharp air blast upwards through the gap after the passage of each component.

One other point of concern was that the taught features of a component might be at an extreme of the measurement range due to the varying errors which have been discussed, and so cause problems during recognition. This could be overcome by passing components to be taught several times through the machine, and storing in the database the average value of each feature.

6.2.3 Image digitisation

During the digitisation of the shoe component the input image which is in a continuum space is transformed into a discrete space. The error generated by this transformation is called the digitising error. For shapes imaged at random orientations the digitising error is random. It also depends on the level of the threshold when the analogue signal from the camera is digitised, as was discussed in section 2.2.2. Its magnitude can be reduced by increasing the spatial resolution of the imaging array. However, this is

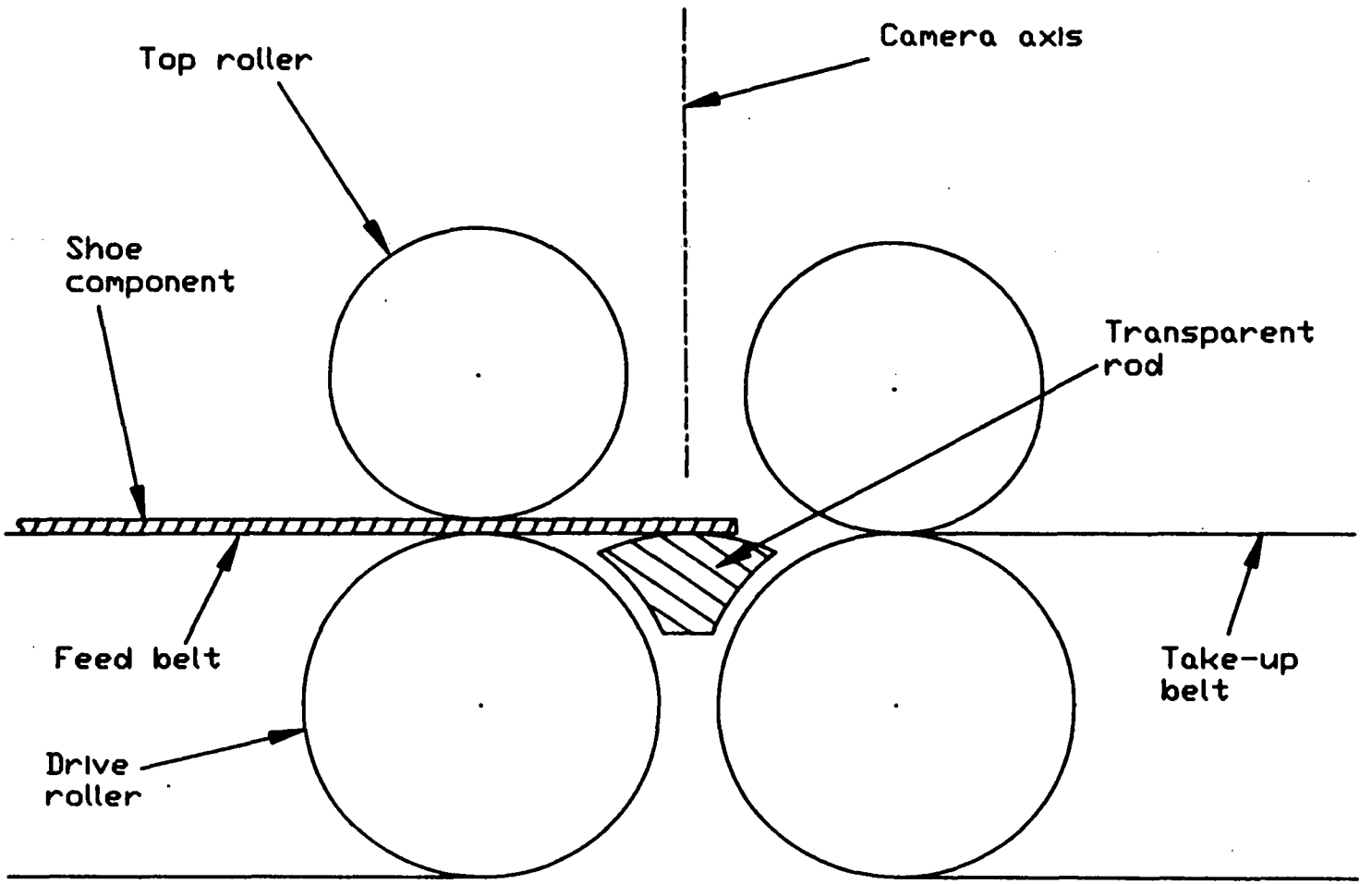


Figure 6.4 *Reducing component droop using a transparent support rod.*

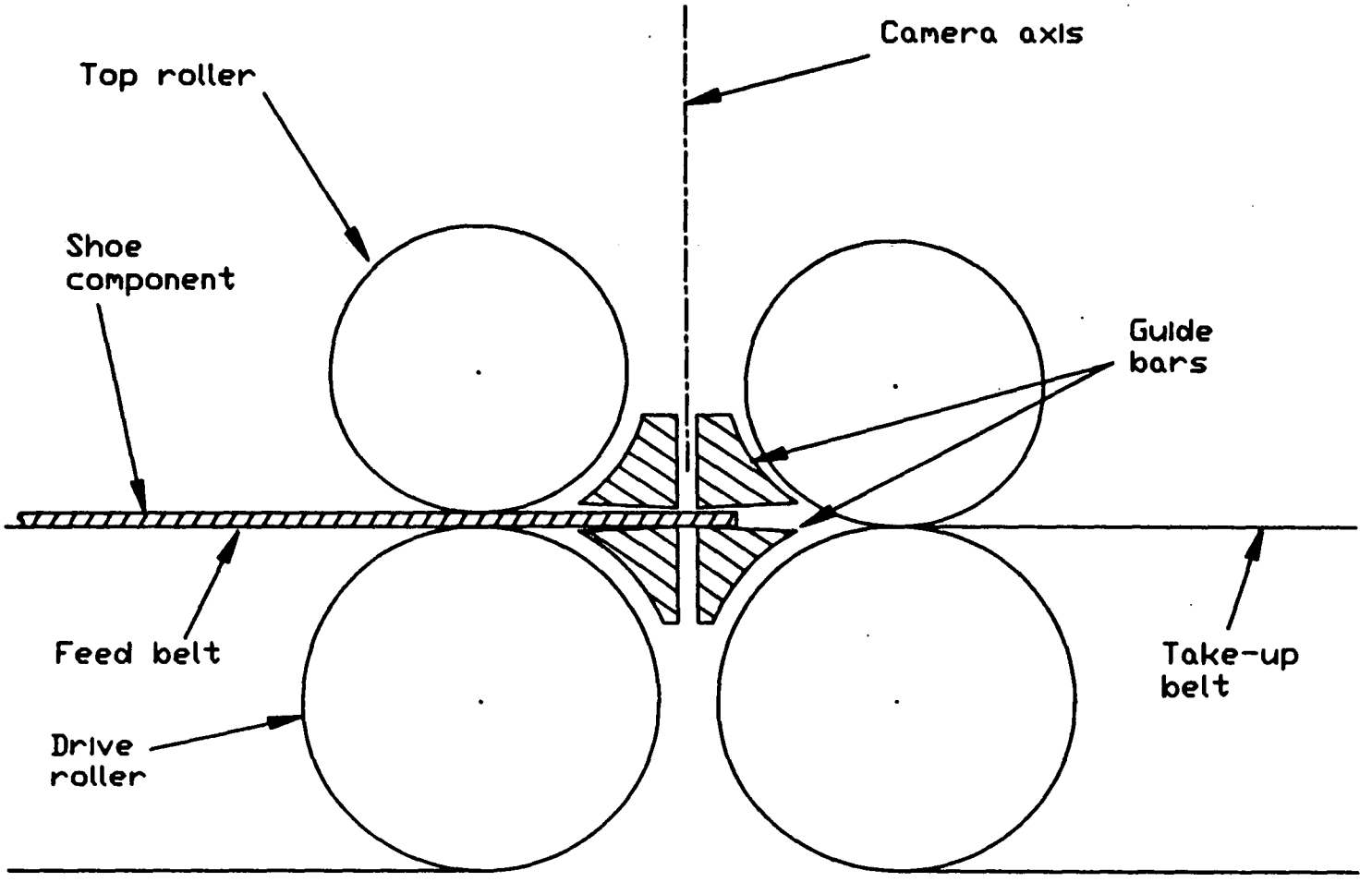


Figure 6.5 *Reducing component droop using guide bars.*

not always a practical solution since the resultant increase in the amount of data obtained from the image will increase the computation time, and the hardware cost too.

The effect of the size of the digitisation cell and the digitising error on the accuracy of the orientation of a shape as determined by the principal axis has been investigated at BUSM [65]. This investigation looked at the way in which the cell size affected the accuracy of orientation of shapes with different 'goodness coefficients' (GCOEFF). As has already been explained in section 5.8, as GCOEFF becomes smaller so the use of the principal axis of a shape for determining its orientation becomes less reliable. This is caused by noise in the outline of the shape of the components due to the digitising error affecting the values of the principal moments used to determine the orientation. By using a computer simulation, the edges of ellipses, rectangles, and triangles were defined accurately mathematically on a square grid. As the elongation of these shapes was varied graphs of the standard deviation of the orientation against GCOEFF were drawn for different pitches of the digitisation grid, the pixel size. These were used to find the lowest value of GCOEFF at which the chance of significant error in the orientation became too large. This was taken to be the value at which 1 in 10,000 shapes would have an orientation error large enough so that a point on the edge of the shape would be displaced by 0.125 mm.

Table 6.2 shows how the lower limit for GCOEFF, corresponding to a 1 in 10,000 chance of poor orientation, varied with the shape and the pixel size. Note that the triangle was worse, and that with a pixel size of 1.27 mm the orientation accuracy was unobtainable at any GCOEFF.

However, it was noted that in practice at a pixel size of 0.127 mm the threshold had to be increased from 0.025 to 0.2 due to i) whiskers on the edges of shapes ii) mechanical vibrations iii) optical distortion iv) imperfect thresholding. These made the actual system a lot less precise than a simulation. Since the time these experiments were performed the pixel size had been increased from 0.127 mm to the 0.203 mm, though the threshold value of GCOEFF had been kept at 0.2.

Shape	Pixel size 0.127 mm	0.254 mm	1.27 mm	127 mm represents
Ellipse	0.02	0.05	0.9	semi-major axis
Rectangle	0.015	0.035	0.5	semi-longest side
Triangle	0.025	0.15	–	2/3 height

Table 6.2 *Theoretical variation with pixel size of the lowest value of GCOEFF for which the principal axis specifies the orientation sufficiently accurately.* The GCOEFF is a measure of the elongation of each shape.

Note that the specification, section 1.5, calls for the sum of all errors to be within 0.25 mm. To print within +/− 0.25 mm at 200 mm from the centroid of a large component requires an upper limit to the angular accuracy given approximately by

$$\sin \alpha = 0.25/200$$

therefore

$$\alpha = 0.072\ ^\circ$$

This is a very high order of angular accuracy to meet.

6.2.4 Effect on recognition of errors in features

Various workers have attempted to determine the theoretical accuracy of the features of digitised 2–dimensional shapes. Kammenos [26] examined the error in the position of the centroid calculated from the edge points only, and applied weightings to the edge points to correct this. However, his shapes had very small areas for their edge length, and in the stitchmarking machine the interior points of the shapes are used in the calculations too.

A theoretical estimation of the variances of various features of digitised shapes, including area, position of boundary points, centroid, and orientation by second moments, was made by Ho [8]. The equations evolved were very specific to the shape digitised, and the number of edge points, and are difficult to apply to a large range of shapes of different sizes. There is also interdependency between features, for example the accuracy of the feature

radii will depend on the accuracy of the calculated centroid position and the calculated orientation of the principal axis. This topic of the effect of errors on the recognition process is discussed further in chapter 7.

6.3 Accuracy of component transport

One vital link in the chain of events which governs the accuracy of stitchmarking is the transportation of the components from scanning to printing. The component's position and orientation on arrival at the printer has to be known accurately. The accuracy of the component's position and orientation depends on the accuracy of movement of the take–up conveyor belt, and any movement of the component relative to the belt.

6.3.1 Transverse accuracy of movement of the conveyor

It was necessary that the part of the take–up conveyor belt which passes a given point of the scanning gap always passes the same point across the width of the printer. The take–up conveyor belt always travels with one edge locating against the flanges on its grooved drive and idle roller. The transverse accuracy of the belt movement was measured by rigidly mounting a fine silver ink pen vertically above the take–up belt next to the scanning gap. When the belt was run the pen drew an easily visible fine line upon the belt. Any sideways wandering of the belt would have been visible as a thickening of the line in certain areas. The line thickening measured was about 0.2 mm.

More important is any transverse movement of the belt as it travels from the scanning gap to the printer. This was measured by fixing a scale to the output side of the printer mechanism and noting how the position of the line which had been drawn on the belt at the scanning gap varied as it passed under the scale. A variation of the position of the centre of the line of +/− 0.2 mm was noted.

These two measurements reflect the accuracy with which the edge of the belt was made, and any movement of the belt across the drive rollers.

6.3.2 Movement of components relative to the conveyor

Experiments were performed to see if there was any movement of the components with respect to the conveyor belts when they passed under the top idler rollers. These rollers had been designed to press the components firmly against the conveyor belts, as was explained in section 4.5.

The outline of the component being tested was traced onto the centre of the feed conveyor using a finely pointed, silver ink, ball point pen. The bank of rollers normally on the takeup belt was then carefully lowered onto the component without disturbing it, with the component almost fully under the first roller. The belt was then accelerated up to constant velocity, as normal, and stopped so that the component was just protruding from under the last roller. The bank of rollers was then carefully lifted off the component and any movement of the component relative to the belt was measured.

The shapes of the components used in the experiments are shown in fig. 6.6. These included the very small components btlk & btln, the medium sized component gg155, and the large component gg23. The use of a range of sizes would show if the area of the shape resting on the conveyor affected the movement. The small shape btl was used in thick (1.90 mm, designated btlk) and thin (0.85 mm, designated btln) versions, to determine the affect of component thickness on the movement. Initially tests were made with the rollers covered in skinned polyurethane resilient foam. The bank of rollers had 11 leading ones of 30 mm diameter and 4 trailing ones of 20 mm diameter. With each component three runs were made, with the results listed in table 6.3.

The results indicate that the movement was greater for small, thick components, though the effect of the orientation was not certain. In some cases there was also a small amount of sideways movement, usually less than 0.1 mm and always to the left with the belt receding, and occasionally a hint of rotation which was not consistent. The same movements could be obtained by keeping the component and the belt stationary and moving the bank of rollers by hand over them. Some of these tests were repeated with the component only moving under half of the rollers, with only about half of

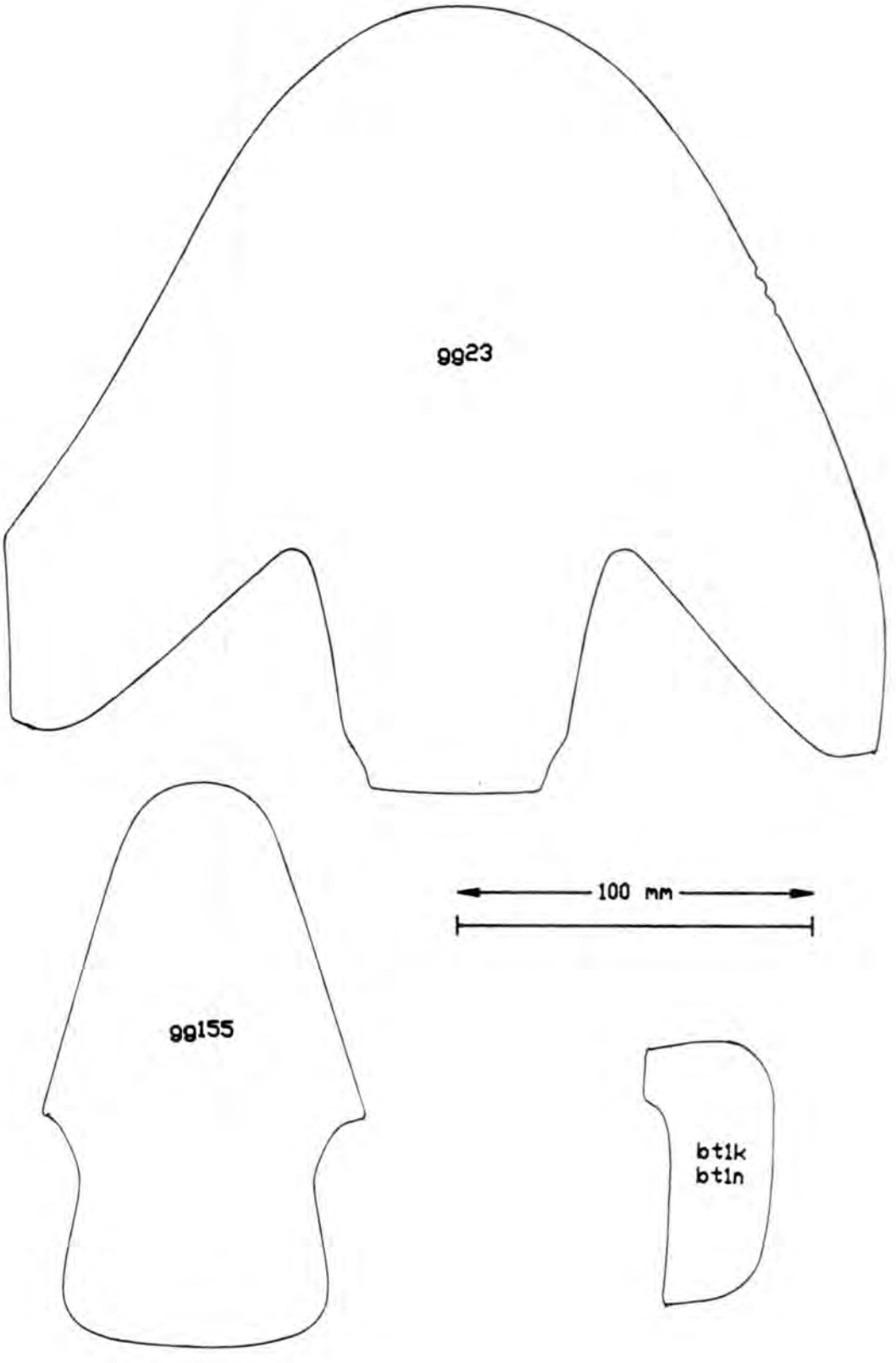


Figure 6.6 *Components used in measurements of movement relative to the conveyor belt.*

Component	Orientation along the belt	Movement along the belt (mm.) First time	Second time	third time
gg23	sideways	0.1	0.1	0.2
gg155	length ways	0.4	0.4	0.4
gg155	side ways	0.2	0.2	0.1
btlk	length ways	1.6	1.2	1.3
btlk	side ways	2.8	4.2	3.2
btln	side ways	3.2	2.8	2.5

Table 6.3 *Movement relative to the conveyor belt of shoe components passing under 30 mm & 20 mm diameter skinned polyurethane resilient foam covered rollers.*

the above movement resulting.

The movement measured appeared excessive, so the covering of skinned polyurethane on the rollers was replaced by the softer unskinned polyurethane covering used when the laser printer was mounted on the machine. This gave the rollers the slightly smaller diameters of 23 mm and 16 mm. The tests were repeated with the results shown in Table 6.4.

Component	Orientation along the belt	Movement along the belt (mm.) First time	Second time	Third time
gg23	side ways	0.2	0.2	0.2
gg155	length ways	0.4	0.4	0.4
gg155	side ways	0.0	0.0	0.0
btlk	length ways	1.6	1.2	1.3
btlk	side ways	2.8	1.8	1.8
btln	side ways	0.8	1.3	1.0

Table 6.4 *Movement relative to the conveyor belt of shoe components passing under 23 mm & 16 mm diameter unskinned polyurethane resilient foam covered rollers.*

As can be seen, there was no consistent improvement with the change to the softer covering, though with the smaller components the movement was reduced. However, even with this reduction, the movement was still greater for small, thick components. It was conjectured that the movement was caused by the backwards component of the force caused when the shoe component had to lift and drive each roller as it passed underneath. The movement was generally greater with the small components where the friction forces between the component and the belt are less.

This hypothesis was tested by repeating these experiments with the polyurethane covered rollers replaced by a set of aluminium ones, of 25 mm and 16 mm diameter. Each roller had a groove every 10 mm along its length. Rubber bands seated in the grooves linked adjacent rollers together, as shown in fig. 6.7. The idea behind this arrangement was that the conveyor belt would drive the rollers and each would drive each other, rather than the component having to drive the ones that it was lifting. Also the rubber bands would start to lift a roller as the component approached, rather than abruptly when the component hit the roller. With this arrangement the results in table 6.5 were obtained.

Component	Orientation along the belt	Movement along the belt (mm.) First time	Second time	Third time
gg23	sideways	1.0	0.8	0.8
gg155	length ways	1.0	1.0	1.0
gg155	side ways	0.2	0.1	0.1
bt1k	length ways	0.8	0.6	0.8
bt1k	side ways	1.0	1.0	1.0
bt1n	side ways	1.0	1.0	1.4

Table 6.5 *Movement relative to the conveyor belt of shoe components passing under 25 mm & 16 mm diameter aluminium rollers linked together by rubber bands.*

Surprisingly, apart from a little improvement in the movement of the

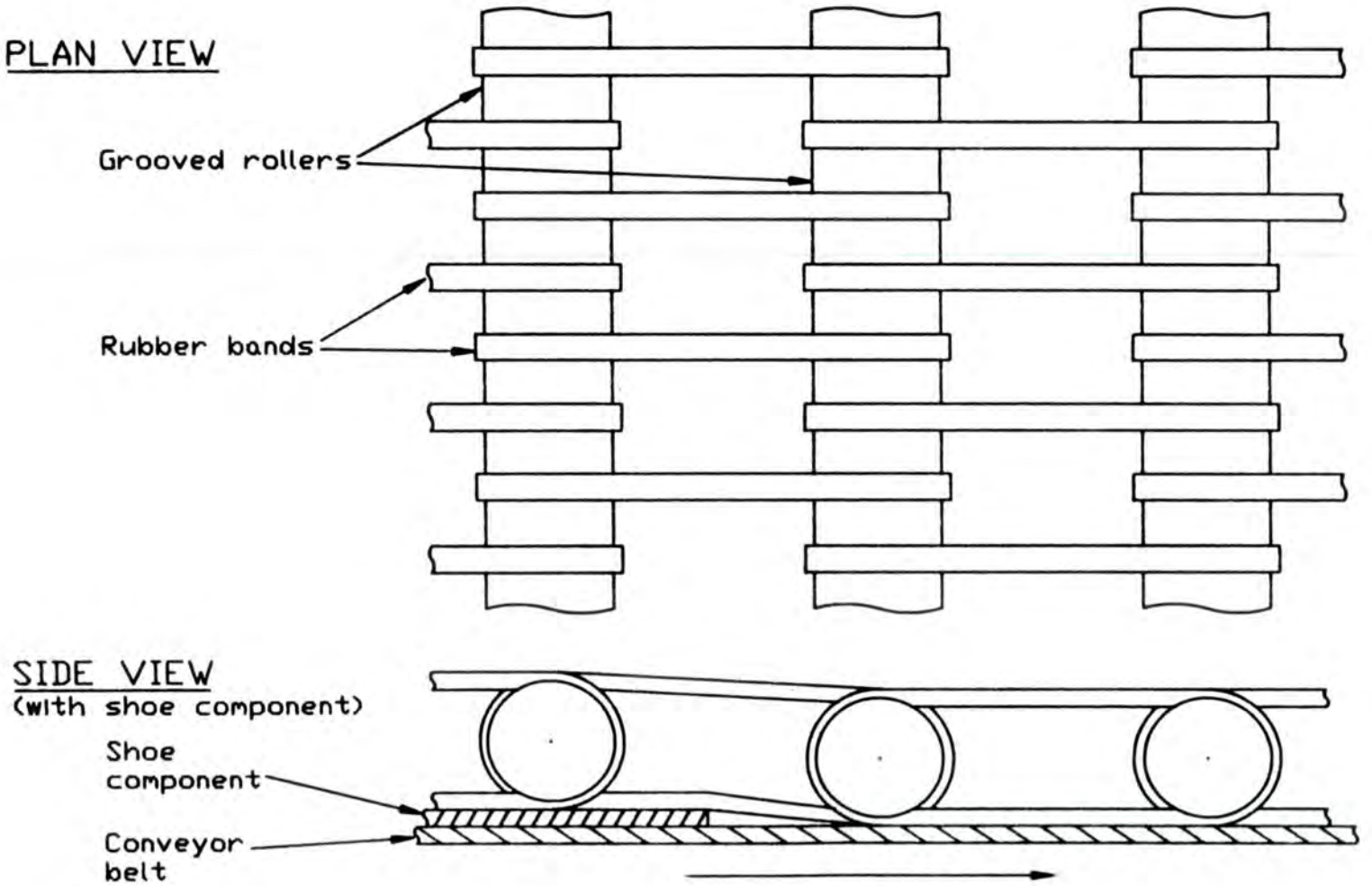


Figure 6.7 *Top rollers linked with rubber bands.*

small components bt1k and bt1n, the results were worse with this arrangement than with the polurethane foam covered rollers. The movement of the components relative to the conveyor belt could be seen as the belt was driven slowly by hand. This movement may be due to a peristaltic type action between the rollers and the component which is more pronounced with these heavier solid metal rollers.

Due to the lack of any significant improvement, these metal rollers were removed and replaced by the previously used unskinned soft polyurethane covered rollers. The machine continued to be operated with these rollers for the remainder of the experimental work.

Some time later, when measurements were being made of the accuracy of printing stitchmarking, a knowledge of the movement under the rollers of the test pieces at different orientations was needed. The results obtained from measurements were not very repeatable, and another mechanism which was involved in the movement of components relative to the belt was suspected. To gain more information, a squat cylinder of steel of 835 g mass was stuck to the top of a sample from the same piece of leather from which the printing test samples had been cut and the force needed to pull the two across the surface of the conveyor belt was measured using a spring balance.

This showed that on pulling the leather for the first time in a new direction the force required to start the leather moving was high, the leather and mass jerked a few times, then settled down to steady motion requiring a reduced force. If the motion was then stopped and restarted then the force necessary to restart the motion was less than it had been at first and there was none of the initial jerking motion. If the piece of leather was then turned to a different orientation and pulled then the whole process was repeated, with the initial force to move being very high again.

This seems to indicate an effect due to the distortion of the fibrous structure of the surface of the leather which contacts the conveyor belt. The shoe components thus have a memory of the way they have previously been handled which may have affected the degree of movement on the conveyor when the components passed under the rollers.

The need for any top rollers was questioned, since there is very little impulse from the stepping of the conveyor drive motors at the high velocity at which the machine runs with the ionographic printer. It is negligible when compared to that at the slow printing speed which was required by the laser printer, as described in section 4.5. With no top idler rollers, the components move slightly during the acceleration and deceleration of the conveyor belts, but when viewed by eye do not appear to move when the conveyor travels at constant velocity. This indicates that there may be no necessity to press the components onto the belts if the belt moves at constant velocity without stopping for recognition. A pair of top rollers would only be needed to hold the components flat while they move across the scanning gap. Unfortunately, with the short belts on the machine there is not sufficient time for shape recognition and stitchmark generation before the component reaches the printer, so this could not be tried. Clearly, more work is required on the transportation of the components and ways of preserving the knowledge of their position.

6.4 Accuracy of stitchmark generation

The accuracy of the printed stitchmarking will depend on the accuracy of the computer implementation of the maths used. For example, whether single or double precision floating point functions are used. The interpolation algorithms used when generating the stitchmarking is the same as those used when the stitchmarking is taught to the system, so the two should correspond faithfully.

All of the stitchmark generation programs were written bearing in mind the accuracy that different implementations would provide. As can be seen from the listings in appendix B all floating point functions are double precision.

6.5 Accuracy of the printing process

There were certain inaccuracies inherent in the printing process, and areas where errors could have crept in if sufficient care were not taken. When

the ion cartridge of the printer was ever replaced it was always inserted into the print engine and pushed hard against the stop, so that it always printed in the same position across the drum. It was held down firmly by spring pressure and did not move when the machine was running.

6.5.1 Thermal expansion of the framework

Since the distance between scanning and printing is large, thermal expansion and contraction of the machine will have an effect on the accuracy along the belt of printing stitchmarking. The framework of the machine is made of mild steel, which has an expansivity of $15 \times 10^{-6}\ K^{-1}$, and the separation between scanning and the printer mounting is 706 mm. So the rate of expansion will be $706 \times 15 \times 10^{-6} = 0.01$ mm K^{-1}, i.e. 0.1 mm for a temperature change of 10 K, which will probably be the maximum excursion from room temperature of 20°C that an operating stitchmarking machine would experience. This will use up about half of the allowable tolerance in stitchmark positioning as given in the specification of section 1.5. This error could be reduced by measuring electronically the temperature of the machine's framework, passing the signal through an analog to digital convertor to the Master Z8000 which could produce a correction to the distance of component travel at which printing commences.

6.5.2 Digitisation of the printing

The printing accuracy is inherently limited by the resolution of the printed image, which is governed by generating the low resolution image in the general purpose RAM of the TMS34010 Software Development Board, as explained in sections 4.6 & 5.11. The printer's resolution of 11.8 dots/mm (300 dots/inch) is reduced by 4 times to 2.95 dots/mm (75 dots/inch) by the need to generate the image at low resolution. As can be seen by considering fig. 6.8, this means that the centres of adjacent printed dots are 0.34 mm apart, and the centre of a printed dot can be up to $0.34 \times 1.41/2 = 0.24$ mm away from the point at which it is required.

The accuracy of printing in the direction along the belt will also be affected by the discrete stepping action of the drive to the printing drum,

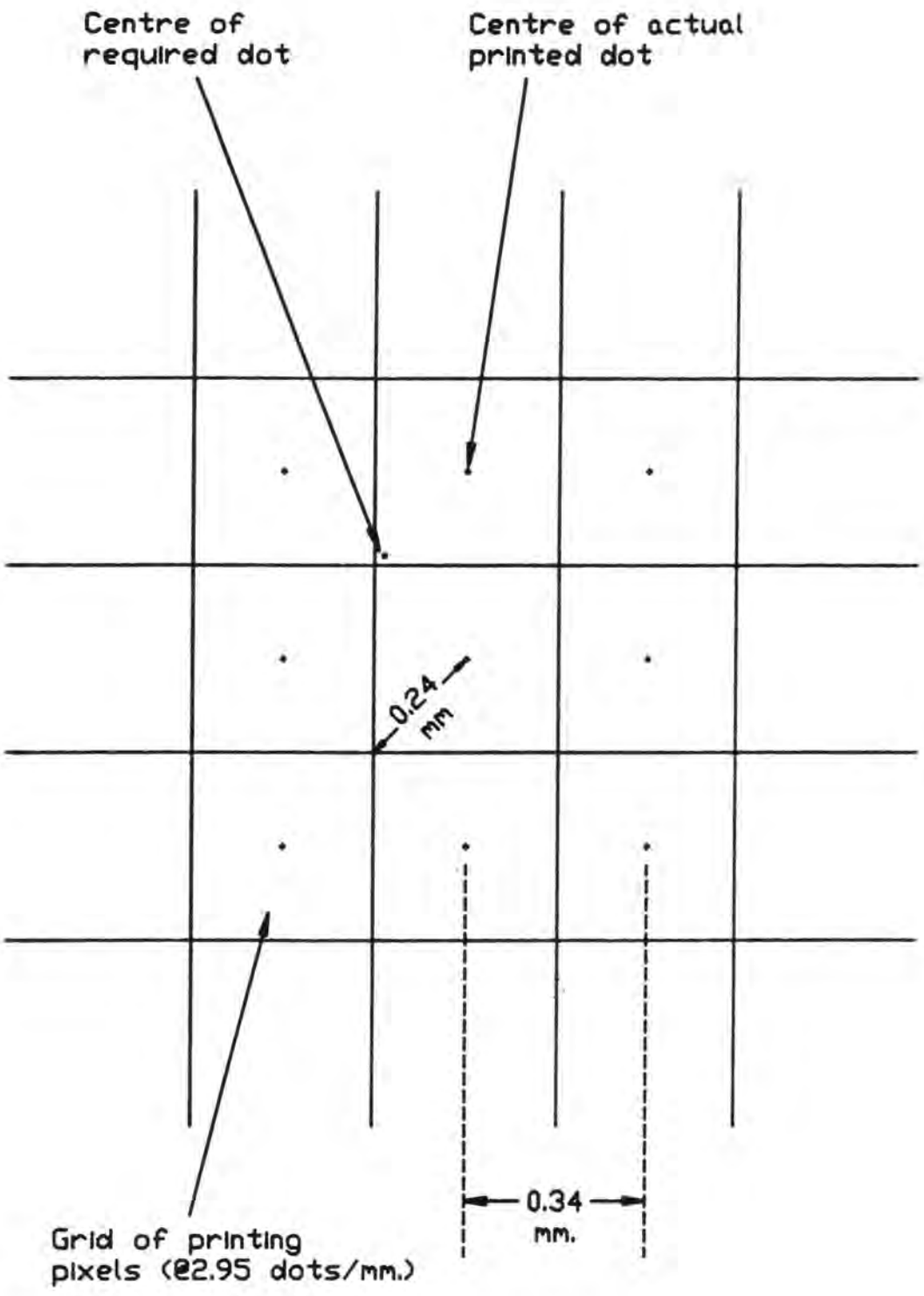


Figure 6.8 *Error due to digitisation of the printing.*

since the belt drive from the print engine's original induction motor had been replaced by the toothed belt drive from the stepper motor. The drum now moves in discrete steps of 0.203 mm at its periphery, with the production of the image on the drum being governed by a shaft encoder which inputs to the deskew board, as explained in section 5.11. There was concern that the printing would be influenced by the stepping of the drum, and interference effects would be visible.

An experiment was performed to see if the discrete stepping affected the image that was printed. A printing test program was written which generated alternate black and white parallel lines one pixel wide in the Image RAM for printing across the drum. This program worked in the normal way, as explained in sections 4.6 & 5.9, expanding and transferring slices from Image to Video RAM as the shoe component passed through the printer. A section of a copy of a printing of this test pattern is displayed in fig. 6.9.

It can be seen that there were some flaws in the printing, but there are no signs of interference patterns. Two possible reasons for this are that the printer electronics, via the encoder responds fast enough to the stepping of the drum to deposit the print image in the correct position. The other is that with the high stepping rate of over 900 steps/sec, driving through a double reduction toothed belt drive, the movement at the drum is smooth. Since there appeared to be no problem with this aspect of the printing it was not pursued any further.

The start of printing is determined by the Master Z8000 microprocessor which counts the pulses it issues to the stepper motors. It sends synchronisation signals to the printer when it has determined that the leading edge of the shoe component has arrived there. Thus, the start of printing can not be more accurate than to the nearest step, which gives a maximum error of 1/2 step. Since the step size is 0.203 mm the maximum error from this cause is 0.1 mm.

6.5.3 Accuracy of joining the slices of the printing image

As explained in sections 4.6 & 5.9, the stitchmarking is produced by

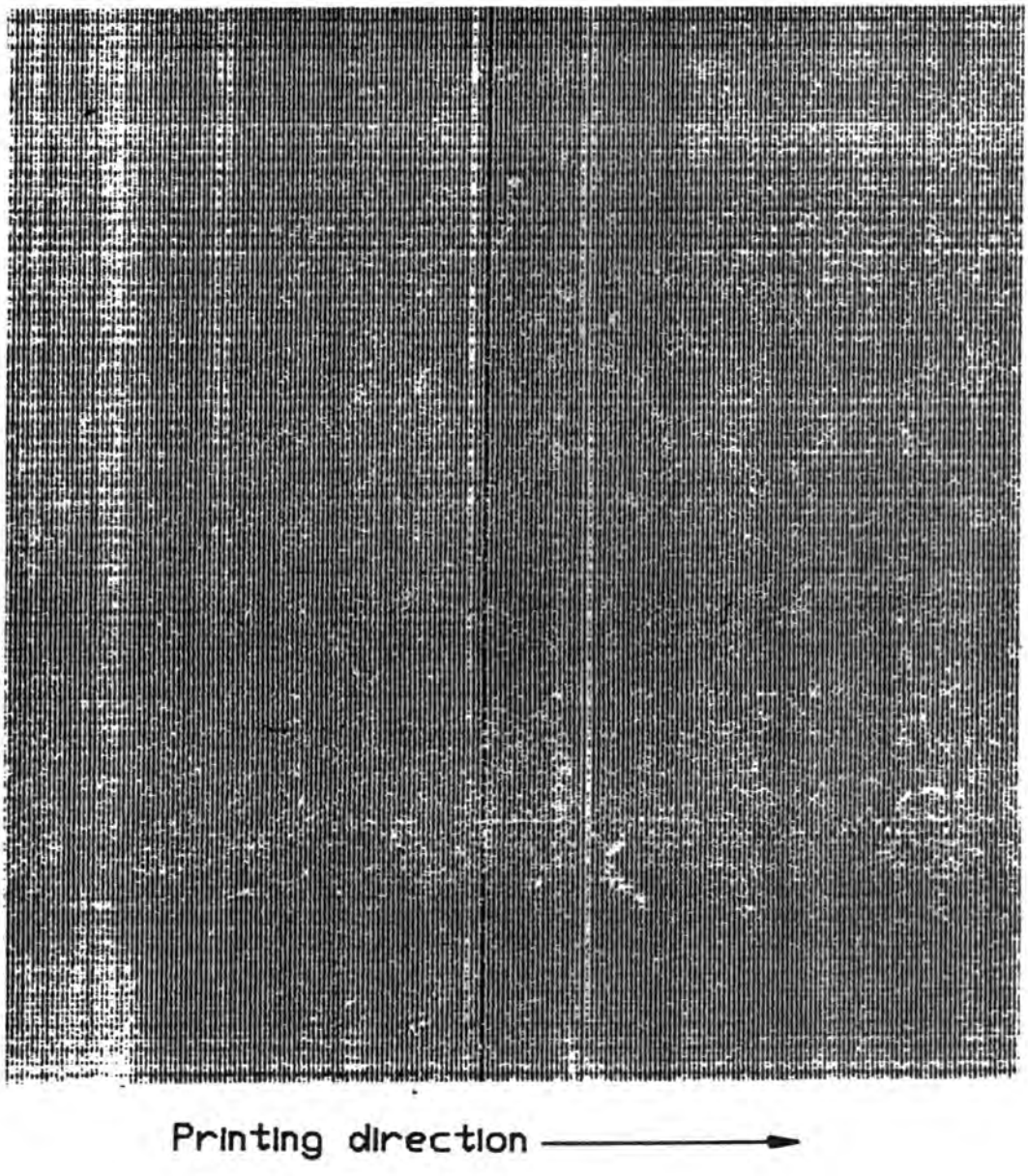


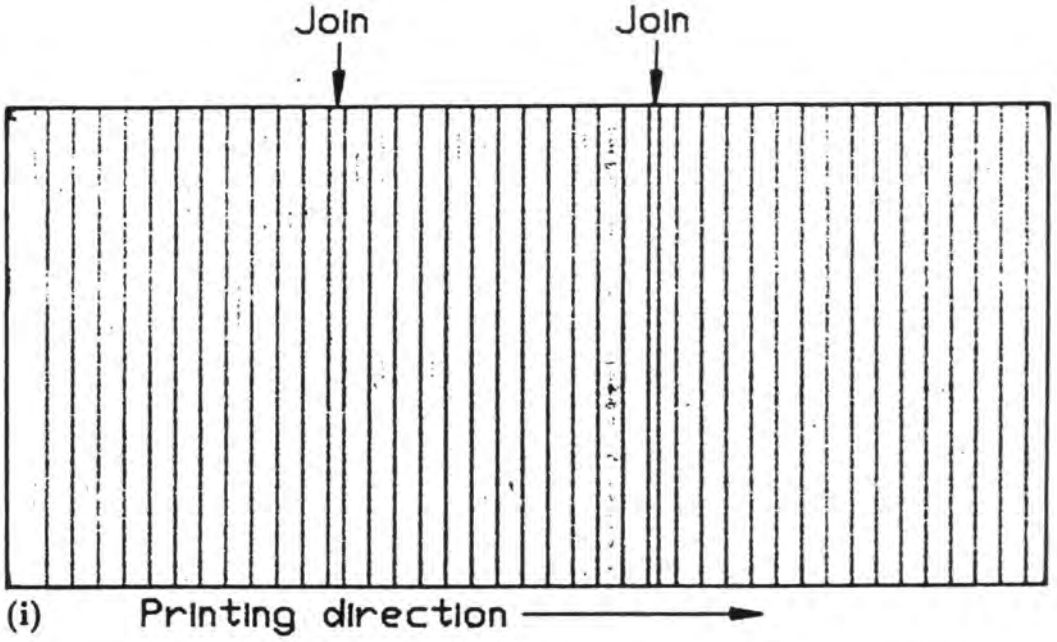
Figure 6.9 *Test printing to determine the presence of interference patterns caused by the discrete stepping of the print drum.*

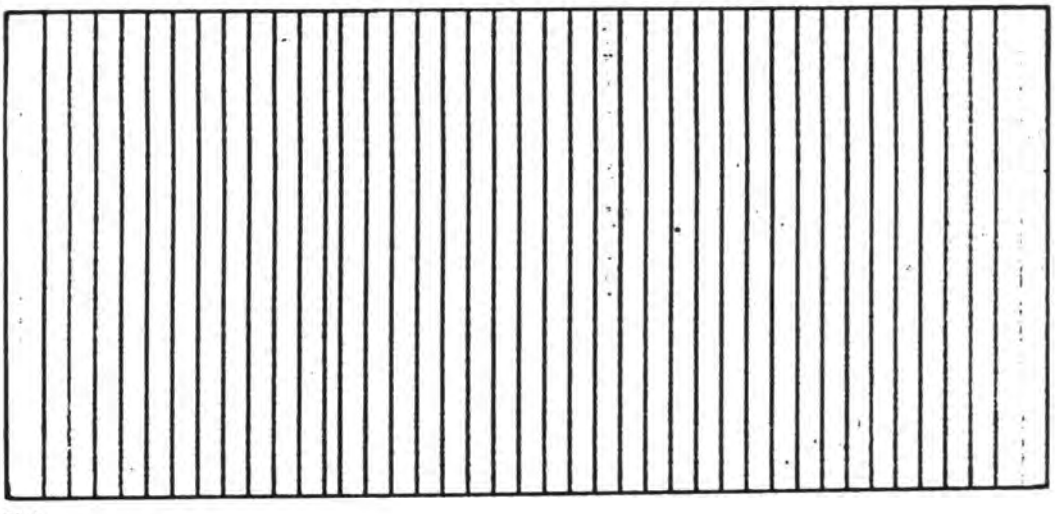
printing slices as the shoe component moves through the printer. Each slice of image produces 512 lines of printing at the drum. With a printing density of 11.81 dots/mm (300 dots/inch), this corresponds to a printing distance along the conveyor of $512/11.81 = 43.35$ mm. The Z8000 Master microprocessor controls the synchronisation of the printing, as explained in section 5.11, and starts the printing of a new slice when the preceding slice has been printed. With a step size of 0.203 mm this corresponds to $43.35/0.203 = 213.54$ steps, which is not an integral number of steps, so the actual number of steps used is 213. Though alternating between 213 and 214 steps would be better when printing over long distances.

To determine how well this technique of transferring slices works in practice, the printing test program was modified. It now generated within the Image RAM parallel lines across the direction of travel spaced 10 pixels apart. This corresponded to a spacing of $(10 \times 4)/11.81 = 3.39$ mm at the print drum. This pattern was printed onto A4 sized pieces of leather, copies of section of which are displayed in fig. 6.10, and the positions and separations of the first 29 lines on each were measured, and are listed in table 6.6. As can be seen from the figure, and the reduced separations between lines 12 & 13 and 25 & 26 listed in the table, the joining of the slices was often imperfect.

The reason for the occasional large error in the spacing of the lines at the join between the image slices is not known. The 1/2 step error in the output of synchronisation pulses over the length of a slice should only have produced a displacement of 0.1 mm. Since ultimately more video RAM will be used, removing the need for printing the image in slices, the causes of the occasional large error were pursued no further.

As described in section 4.6, it was not possible to exactly match the peripheral velocity of the print drum with that of the conveyor belts. Both were driven by 200 step/rev stepper motors, with the conveyor drive having a timing belt reduction of 16:30 to the 15 tooth drive roller for the 1/5 inch pitch toothed conveyor belt. This produced a belt movement per step of the motor of $(15 \times 0.2 \times 16)/(200 \times 30) = 0.008$ inches, or the 0.20320 mm of the digitisation cell size.





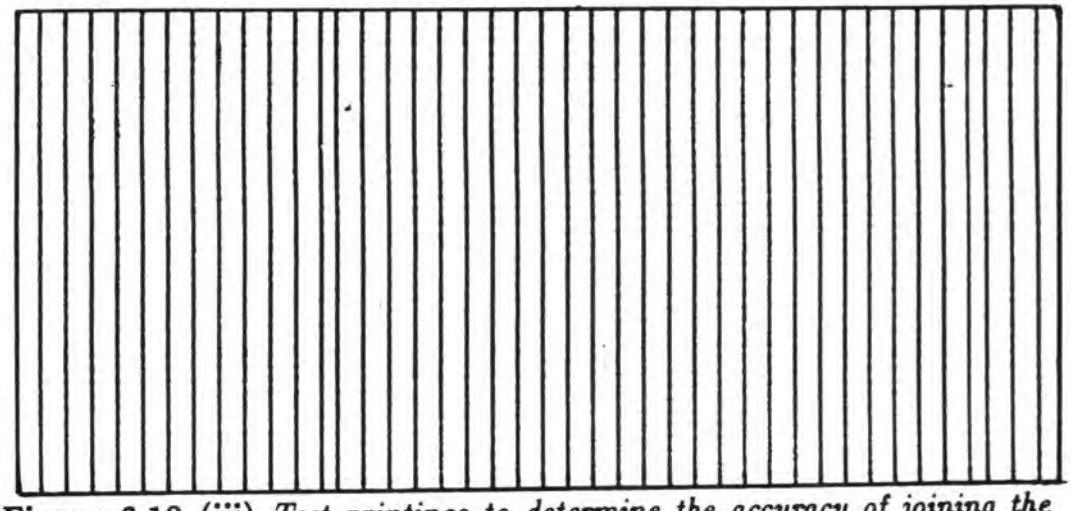


Figure 6.10 (iii) *Test printings to determine the accuracy of joining the image slices.*

Line number	(i) Distance from start (mm)	(i) Distance from previous line (mm)	(ii) Distance from start (mm)	(ii) Distance from previous line (mm)	(iii) Distance from start (mm)	(iii) Distance from previous line (mm)
1	5.1		5.0		4.0	
2	8.5	3.4	8.3	3.3	7.4	3.4
3	11.9	3.4	11.7	3.4	10.8	3.4
4	15.3	3.4	15.1	3.4	14.1	3.3
5	18.8	3.5	18.4	3.3	17.4	3.3
6	22.1	3.3	22.9	3.5	20.8	3.4
7	25.4	3.3	25.3	3.4	24.1	3.3
8	28.8	3.4	28.7	3.4	27.6	3.5
9	32.2	3.4	32.1	3.4	31.0	3.4
10	35.6	3.4	35.4	3.3	34.4	3.4
11	39.0	3.4	38.7	3.3	37.7	3.3
12	42.4	3.4	42.1	3.4	41.1	3.4
13	44.4	2.0	44.2	2.1	43.2	2.1
14	47.8	3.4	47.5	3.3	46.7	3.5
15	51.2	3.4	50.9	3.4	50.1	3.4
16	54.6	3.4	54.3	3.4	53.5	3.4
17	57.9	3.3	57.6	3.3	56.9	3.4
18	61.3	3.4	61.0	3.4	60.1	3.2
19	64.7	3.4	64.3	3.3	63.7	3.6
20	68.0	3.3	67.7	3.4	67.1	3.4
21	71.4	3.4	71.1	3.4	70.4	3.3
22	74.7	3.3	74.5	3.4	73.7	3.3
23	78.0	3.3	77.9	3.4	77.0	3.3
24	81.4	3.4	81.2	3.3	80.4	3.4
25	84.8	3.4	84.6	3.4	83.8	3.4
26	86.2	1.4	87.9	3.3	87.2	3.4
27	88.6	2.4	91.3	3.4	90.6	3.4
28	91.9	3.3	94.8	3.5	94.0	3.4
29	95.2	3.3	98.2	3.4	97.4	3.4

Table 6.6 *Positions and spacing between lines of the slice join test prints.*

The drum diameter was measured to be 100.035 mm, and the closest match to the movement of the conveyor that could be obtained with commercial timing belt components was obtained by using two stages of reduction of 14:42 and 14:36. Hence for each step, the drum moved by $(100.035 \times \pi \times 14 \times 14)/(200 \times 42 \times 36) = 0.20369$ mm. This was slightly larger than the step size of the conveyor belts, and meant that the drum should have had a higher velocity by $100 \times (0.20369 - 0.20320)/0.20320 = 0.24$ %. Without any compensation for this, stitchmarking at the end of the longest possible image of 347 mm would have been printed $347 \times 0.24\% = 0.8$ mm forwards

The velocity of the take–up belt was measured by pressing against it a hard rubber wheel on a free spindle and noting the time for 100 revolutions. The diameter of the wheel was 48.39 mm, giving a circumference of 152.02 mm. The measurement was made 5 times, giving a mean velocity of 180.8 mm/sec, with a range of 179.9 to 181.6 mm/sec. The actual peripheral velocity of the drum was determined similarly by measuring the time for 100 revolutions of the hard rubber wheel when pressed against it. The mean velocity obtained was 183.2 mm/sec, with a range of 182.9 to 183.4 mm/sec. Comparing the two velocities shows that the peripheral velocity of the drum was actually $100 \times (183.2 - 180.8)/180.8 = 1.33$ % higher than that of the belt.

6.6 Measurement of stitchmarking accuracy

Initially the parameters in the operating programs which governed the position of printing had been adjusted to produce printing in the correct position by passing the same shoe component through the machine twice, the second time rotated by 180°, and measuring the average offset between the two images. Although this was satisfactory for demonstration purposes, it could be seen that the image placement did vary occasionally by a millimetre or two.

Experiments to measure the accuracy of printing were made using typical leather to which graph paper was attached with double sided adhesive tape. A trapezium and an isosceles triangle were cut from this since their

features could be easily calculated. The triangle was a 'good' shape with a GCOEFF of 0.64 and the trapezium was a 'bad' shape with a GCOEFF of 0.06. Each shape was taught to the system with its principal axis lying along the centre line of the conveyor belts. Using the graphical stitchmark teaching program straight stitchmark lines were added which were parallel to the sides of the shapes. These were not used during the measurements, but would show if anything odd had occurred during printing. Datum lines for measurement purposes were added by editing the stitchmark data files for the two shapes. A line was added along the principal axis of the shape for checking the accuracy of the orientation, and also a line cutting this at the centroid of the shapes. Also, crosses were added at known places towards the corners of the shapes, as illustrated in figs. 6.11 & 6.12.

The shapes were then passed through the machine at different measured orientations and were printed. The accuracy of the printing was assessed from measurements of the positions of the various lines on the graph paper, measured using a magnified scale. Measurements were made to the assessed centres of the printed lines. Since the graph paper had consistent errors in the printing of its lines in both directions it had to be calibrated and corrections were applied to the stitchmarking positions.

6.6.1 Accuracy of printing along the conveyor.

Table 6.7 lists the results for the positioning along the belt of the first points to be printed on each shape. These show that the first part of each shape was printed with a mean error along the belt of -2.05 mm, and with a range of error from -1.3 mm to -2.5 mm, i.e. 1.2 mm.

It was at this point that the experiments described in section 6.3.2 were performed to measure how these shapes moved relative to the conveyor belt when passing under the top rollers. Movement of the triangle in the range from 0.0 mm to 0.5 mm occurred, with rotation as well, but was not consistent, as discussed in the previous section.

To correct as far as possible the start of printing, the constant in the stitchmark generation program which controls the start of printing was

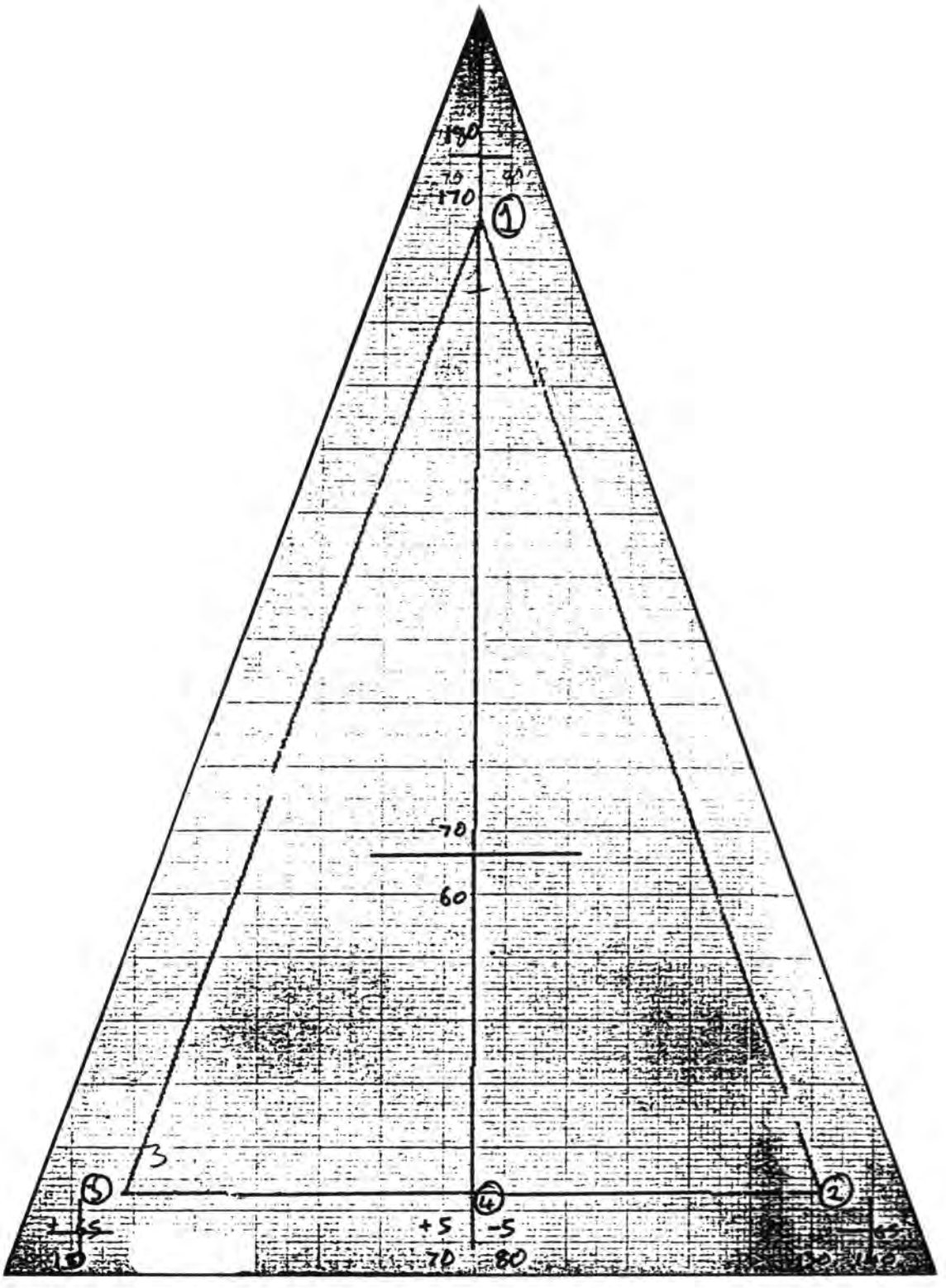


Figure 6.11 *Triangular shape used for accuracy measurements.*

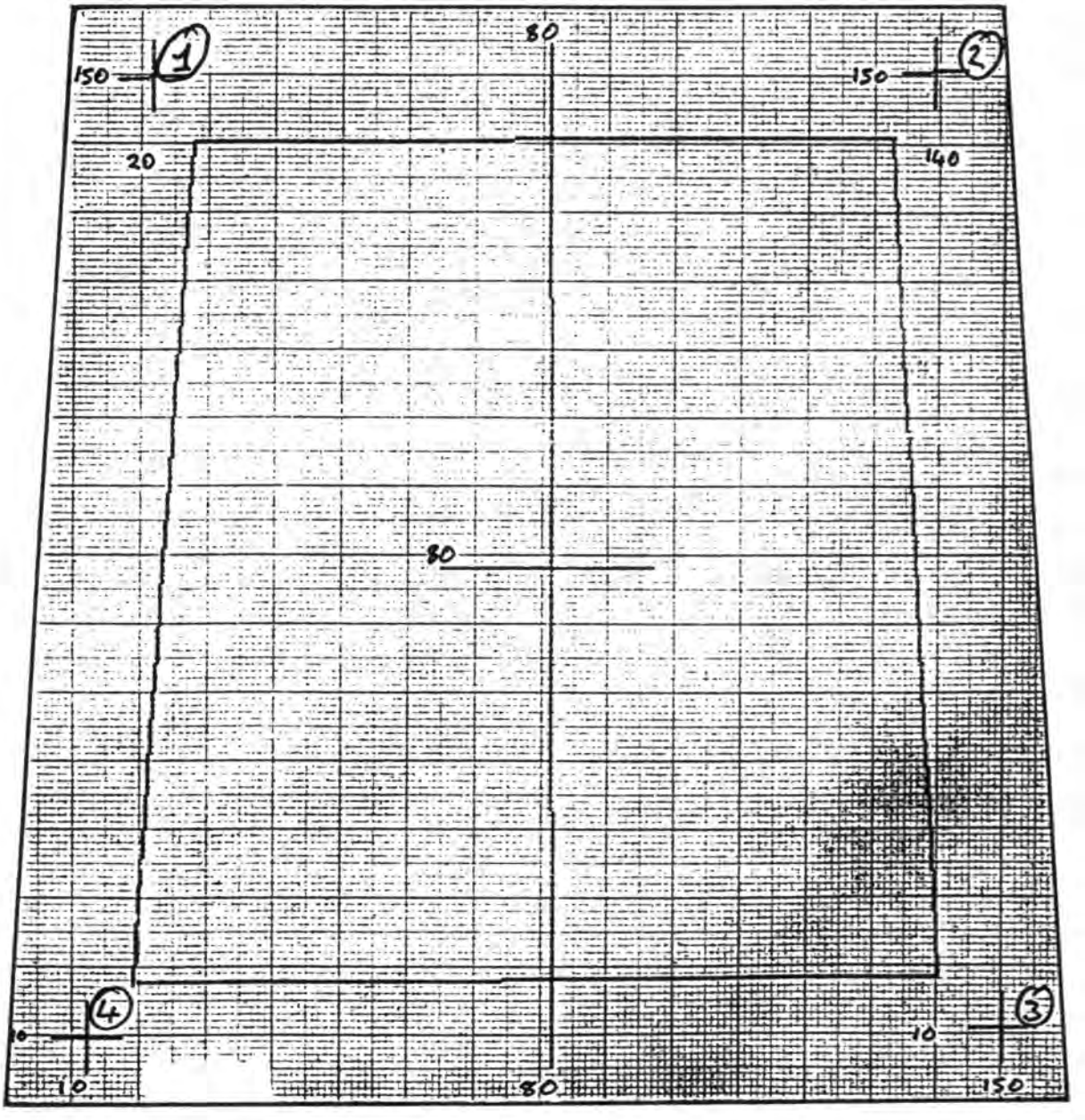


Figure 6.12 *Trapezium shape used for accuracy measurements.*

Shape	Orientation	Measurement point	Positional error (mm.) *
Trapezium	0°	3	-1.3
	90°	1	−2.5
	90°	2	−1.9
	180°	4	−2.5
	270°	3	−2.3
	270°	4	−2.1
Triangle	0°	2	−2.7
	90°	1	−1.6
	180°	3	−1.8
	270°	2	−1.6
	270°	3	−2.3

* A positive error signifies that the printing had occurred too soon.
A negative error signifies that the printing had occurred too late.

Table 6.7 *Accuracy in positioning along the belt of the first points to be printed on a shape.*

altered so that printing started 2.0 mm earlier. This reduced the mean error in the start of printing to almost 0.0 mm, but the range was still 1.2 mm, i.e. from −0.5 mm to +0.7 mm. At least some of this range of values was due to movement of the components relative to the conveyor belt.

Next, the separations between points in the direction along the conveyors were measured to see how they compared with their theoretical values. The results for points on the trapezium are listed in table 6.8. The printing along the belt was consistently foreshortened, with the spread of values being caused by the poor joining of the slices of the image, as described in section 6.5.3. The mean foreshortening was 1.32 % which compared favourably with the prediction from the previously measured speed mismatch between the conveyor belt and the drum that the printing would be 1.33 % foreshortened. However, the mean value of the foreshortening of the printing will have been increased by the poor joining of the slices.

Orientation of shape	Measurement points	Theoretical separation (mm.)	Actual separation (mm.)	Percentage difference *
0°	3–4	140.2	138.8	−1.14
90°	1–4	140.2	138.0	−1.50
90°	2–3	140.2	138.4	−1.21
180°	3–4	140.2	137.9	−1.78
270°	1–4	140.2	139.4	−0.50
270°	2–3	140.2	137.6	−1.78

* A negative percentage difference signifies that the separation of printed points was less than it should theoretically have been.

Table 6.8 *Accuracy in the separation along the conveyor between pairs of points printed on the trapezium.*

The obvious way to overcome the foreshortening was to produce a better match between the velocities of the conveyor belt and the periphery of the drum. However, this is not easy with a stepper motor drive and toothed belt speed reduction, since only discrete speeds are available. The match could have been improved by issuing an extra step to the drum when necessary to increase its mean speed, though this might have caused odd effects to the printing when the extra step occurred. Another method was to use a servo motor drive to the drum, with the speed controlled by the output from an encoder driven by the conveyor belt. However, this would have been a complicated and expensive solution. The simplest solution was to accept the slight speed mismatch and correct for it by scaling the stitchmarking appropriately in the direction along the belt. A scaling factor of 1.0132 (i.e. 1.32 %) was applied to the y coordinates of all the stitchmark data points after they had been transformed for printing on the scanned shoe components. When the trapezium was printed again the mean difference between the actual and theoretical separations was reduced from 1.32 % to +0.29 %, with a variation of −0.14 % and +0.65 % which was probably caused by the poor joining of the slices of image. This shows that compensating for the speed mismatch between conveyor belt and drum by scaling the image in that direction was effective.

6.6.2 Accuracy of printing across the conveyor

To determine the accuracy of the absolute position of printing in the direction across the conveyor the differences between the theoretical and actual positions of printing at the centroid of the shapes was measured. Table 6.9 lists the results which show a range of just -0.3 mm to $+0.3$ mm, which is very close to the specified accuracy of $+/-0.25$ mm.

Shape	Orientation	Error in X–position of centroid (i)	(ii)	(iii)
Trapezium	0°	+0.1		
	90°	+0.1		
	180°	+0.0		
	270°	+0.1		
Triangle	0°	+0.2	−0.1	−0.3
	90°	+0.1	+0.2	+0.0
	180°	+0.1	+0.1	+0.3
	270°	−0.1	+0.2	+0.0

Table 6.9 *Accuracy in the direction across the belt of the printed position of the centroid on the shape.*

To see if there was any distortion in the printing in the direction across the conveyor, the theoretical and measured separation of printed points on the trapezium were compared. As shown in table 6.10, there was some foreshortening with a mean error of 0.20 %. The cause of this is not clear, but it could again have been corrected by a scaling factor in that direction.

Orientation of shape	Measurement points	Theoretical separation (mm.)	Actual separation (mm.)	Percentage difference *
0°	1–4	140.2	139.8	−0.21
0°	2–3	140.2	139.9	−0.21
90°	1–2	119.8	119.6	−0.14
90°	3–4	140.2	139.9	−0.21
180°	1–4	140.2	140.0	−0.14
180°	2–3	140.2	139.8	−0.29
270°	1–2	119.8	119.5	−0.23
270°	3–4	140.2	139.9	−0.21

* A negative percentage difference signifies that the separation of printed points was less than it should theoretically have been.

Table 6.10 *Accuracy in the separation across the conveyor between pairs of points printed on the trapezium.*

6.6.3 Accuracy of the orientation of the printing

The coordinates of the ends of the lines printed along the principal axis gave the orientation of the printing, and the errors from its theoretical direction are listed in table 6.11 for the trapezium, a 'bad' shape, and table 6.12 for the triangle, a 'good' shape.

These tables show a range of errors for the trapezium of −0.3 to −0.6°, and for the triangle of +0.1 to +0.3°, where the positive direction is clockwise. These errors were due to both the determination of the orientation of the shape and errors in generating the stitchmarking, though the former was likely to be the larger. A larger range of shapes would have to be tested to see whether the differences between the accuracies of orientations of 'good' and 'bad' shapes is consistent.

6.7 Conclusions

Errors in the measurement of features are due to contributions from

Orientation	Error in orientation of the printing (degrees)
0°	−0.33
90°	−0.57
180°	−0.45
270°	−0.37

Table 6.11 *Accuracy in the orientation of the stitchmarking printed on the trapezium, a 'bad' shape.*

Orientation	Error in orientation of the printing (degrees)
0°	+0.47
90°	+0.10
180°	+0.34
270°	+0.17

Table 6.12 *Accuracy in the orientation of the stitchmarking printed on the triangle, a 'good' shape.*

many sources. One of the major sources is due to the gap between the conveyor belts through which the components were scanned. In the worst cases the range of measurements of the same feature on different recognition attempts were comparable with the range of the tolerance used on that feature. However, there may be sufficient redundancy within the recognition system to cope with this most of the time, though it would have an effect by increasing the rejection and substitution rates. The effect of the gap could be reduced or eliminated by inserting a supportive transparent rod or bars, as described previously.

The use of slices of the image to overcome the problem of lack of video memory was not perfect since the joins between slices were often visible. Since in future implementations more memory would be used, the reasons for the poor joins was not pursued.

It was difficult to dissociate the effects of errors in the printing from

errors in the detection of the position and orientation of the components, and their movement relative to the conveyor belt. The overall accuracy of printing was improved by the scaling method described, though the placement of the printing along the conveyor had a large range, probably caused by the unpredictable movement of the test components when passing under the top rollers. Further investigations of methods of preventing the components from moving on the conveyors is required.

The accuracy of printing in the transverse direction was much better, had less spread, and was almost within the specified accuracy. The accuracy of the orientation of the printing was less than that mentioned in section 6.2.3 as required for accurately printing stitchmarking at the edge of a large shoe component. Movement of the components as they passed under the rollers would have contributed to this too.

CHAPTER 7

THE PERFORMANCE OF THE RECOGNITION SYSTEM

7.1 Introduction

This chapter deals with an investigation of aspects of the performance of the recognition system with shoe components. Firstly the efficiency of the recognition system as it arrived was determined, then the effects of various attempted improvements were investigated. Also dealt with is an analysis of the distribution of shapes within a large database.

7.2 The recognition efficiency

Ideally, the rejection rate and the substitution rate would be zero, where rejection is the failure to assign a class to a scanned component and substitution is the assignment of the wrong class to the component. Rejection has less serious consequences since the components can be recycled through the machine, whereas substituted components will have incorrect operations performed on them and may be damaged and damage machinery.

Previously, when the recognition efficiency had been investigated at City University [3] databases of up to 180 shapes had been used. For the experiments which were about to take place a database of about 600 shoe component shapes was amassed, which included those shapes used in the City University and BUSM databases. This database included the complete ranges of many sets of components separated from one another by just 1/2

size. The adjacent sizes of these are often difficult to distinguish by eye, and made a good test for the system. The range of components was very wide, and included some small slender shapes. These were expected to cause difficulties, for reasons that will become apparent in section 7.4, due to their large number of small feature–radii.

It has been estimated by others that a typical large shoe factory is likely to have about 50 styles of shoe in manufacture or potentially in manufacture. If each style has 18 sizes, a left and a right, and with 10 different components per shoe upper then there will be a total of 18,000 different shapes to be stored in the database for recognition. Although the actual database used for experimentation was 30 times smaller than this, it was decided that valuable knowledge could be obtained from it before progressing to a larger one.

The components whose shapes composed the database included those obtained from shoe factories which were cut from leather and synthetic materials and also thin card patterns from which the knives to cut out the components are made. Although the card patterns were likely to perform better than actual shoe components, since they are of uniform thickness, are stiff, and do not have ragged edges, they were included to give an indication of the range of shapes that are likely in a shoe factory. All of these shapes were taught to the system in an arbitrary order, with arbitrary positions and orientations, mostly without any stitchmarking details.

The recognition parameters used for the test were those that had been in use at BUSM as the most efficient. These will be referred to in future as the 'Standard Tolerances', and were

i) root–area tolerance = 4,

ii) fourth root of principal second moments tolerance = 4,

iii) radius tolerance = 5, and

iv) minimum distance = 5. A minimum distance of 5 requires that the feature–radius set of the scanned shape and the closest matching candidate shape must have no more than 5 radii which do not match

within tolerance, otherwise the scanned shape is deemed to have been rejected.

For the test all of the components were passed through the machine at least twice. Before each component was scanned its class name was entered into the computer. After recognition the computer printed out the recognised class name, or whether the shape had been rejected or substituted. If there had been a substitution then the scanned shape was examined to check that its class name had been entered correctly. If an incorrect entry had resulted in the apparent substitution, then this was corrected on the printout. Also printed out were the feature–radius set of the scanned shape and the set of the matching class, from the database.

Of the 1249 shapes passed through the machine there were no rejections and 29 substitutions, corresponding to a rate of 2.3 %. Initially there had been more substitutions, but upon examining the shapes concerned it was found that although some pairs of components had been marked as being for different sized shoes they were in fact identical. In these cases the details of one of the pair was erased from the database.

The rate of rejection was excellent, but the rate of substitution was higher than hoped. The reasons for this are discussed in the next section.

7.3 Examination of the rate of substitution

The substitution rate of the recognition system is not as good as that predicted in the early days of this long term project. For example Norton–Wayne & Obray [68] predict a probability of substitution of 0.015% for a system that was a predecessor of the present system.

An analysis of those shapes between which substitution occurred show that they fall into only one of two categories. Either they were the same design, but 1/2 size apart, as shown in fig. 7.1, or they were left and right components which are almost symmetrical about an axis, as shown in fig. 7.2. There were no substitutions between unrelated shapes.

In all of the previous work on substitution rate there was little consideration of the distance (in terms of features) between similar shapes within

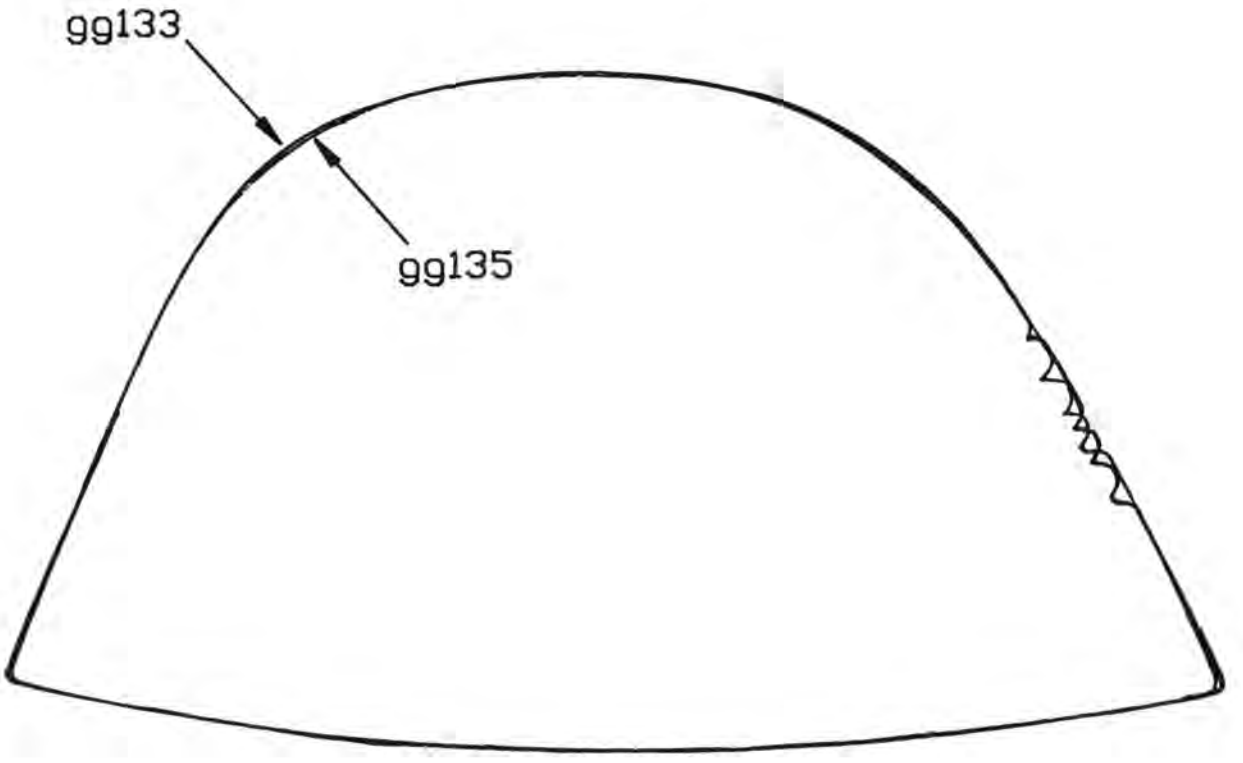


Figure 7.1 *Shapes gg133 and gg135 which differ by 1/2 size and are regularly substituted.*

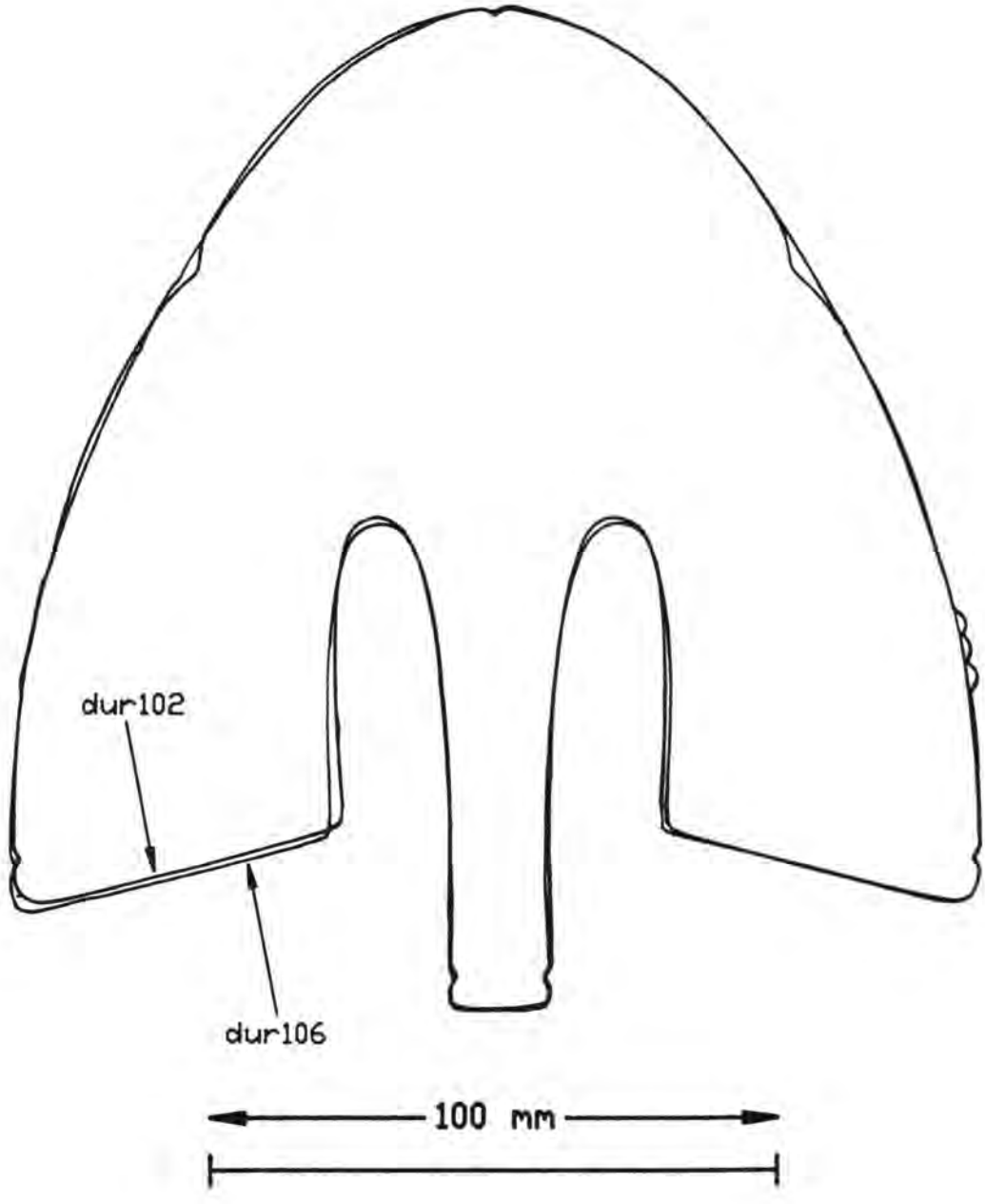


Figure 7.2 *Shapes dur102 and dur106, which are left and right of the same nearly symmetrical design, and are regularly substituted.*

the database in the multidimensional feature space. Obviously, if shoe components are being designed which are very similar there must come a point at which it is acknowledged that the recognition system is not going to be able to distinguish between them.

In order to find how close the shapes might be within the feature space programs were written to analyse the databases. One program analysed the 'good' shapes (since they are simpler to deal with). It checked each shape within the database by searching for others with matching root–area and second moments. Of these others it noted which one had the closest matching set of feature radii compared to the shape being checked, and printed out the name of this shape and the number of radii by which it differed. At the end it printed a summary listing the number of cases of each minimum distance.

The program was run on the data of the 'good' shapes of the City University Database (those prefixed with 'ck'), using the standard tolerances, with the good results summarised as example 1 in table 7.1. This shows that all of these shapes are well separated in the feature space since the smallest minimum distance between any of the shapes is 9, of which there are 4 cases. However, a distinctive characteristic of these shapes is that there is always a whole size between adjacent sizes, never a half size, which accounts for their large separation.

The results for the 'good' shapes of the BUSM database (with names prefixed with 'gg'), on which a lot of work was performed at BUSM, are summarised as example 2 in table 7.1. Within this database there are several sets of shapes separated by a half size with the result that the minimum distances between some shapes are very small or nonexistent. The latter was confirmed from a printout of the features of the shapes within the database made while the machine was at BUSM, and so has not been introduced by alterations to the machine at Durham.

The results for the 'good' shapes of the complete database are summarised as example 3 in table 7.1. This shows that of the 472 'good' shapes within this database 6 shapes cannot be uniquely distinguished using the standard tolerances. However, the recognition algorithm sees the distribu-

Minimum distance	Number of cases of each minimum distance				
	Example 1	Example 2	Example 3	Example 4	Example 5
0	0	4	6	14	24
1	0	12	12	20	38
2	0	9	17	23	26
3	0	3	16	18	19
4	0	9	17	19	8
5	0	6	18	18	8
6	0	2	6	6	5
7	0	2	5	5	0
8	0	2	12	14	17
9	4	2	7	11	11
10	0	0	4	14	14
11	0	0	0	13	10
12	0	0	6	9	21
13	0	0	6	20	25
14	0	0	47	70	71
15	1	1	36	40	30
16	2	41	209	112	99
17	4	2	3	1	1
18	6	0	0	0	0

Example 1 summarises the results for the City University database.

Example 2 summarises the results for the BUSM database.

Example 3 summarises the results for the large database, including examples 1 and 2.

Example 4 summarises the results for the large database with radii reversal allowed.

Example 5 summarises the results for the large database with radii reversal allowed, and radii less than 80 pixels set to zero.

Table 7.1 *Minimum distances between the 'good' shapes of various databases.*

tion of the database shapes in the feature space in an even worse light. After determining the closeness of the match between the set of feature–radii of the scanned shape and the candidate shape it also tries matching with the candidate's set of radii incremented through 180°, which is termed 'radii reversal'. This is required since the orientation algorithm has trouble with some shapes and determines their orientation wrongly by 180°. The results of allowing this reversal of the radius set are summarised as example 4 of table 7.1, which shows that 14 shapes cannot then be uniquely distinguished. Clearly, this is not satisfactory for distinguishing between all of the shapes within the database. Altering the tolerances will have some effect on this, since the substitution rate can be lowered by lowering the tolerances on the features, but at the expense of increasing the rejection rate. This will be investigated in section 7.5. In the next section another phenomenon which affects the substitution rate will be examined.

7.4 Treatment of small radii

In order to determine the feature–radii of a shape described by a set of edge points, stored in the order in which they were received, the feature–radius extraction algorithm searches through the of edge points looking for those which lie within an angular tolerance of the required direction from the shape's centroid. As shown in fig. 7.3, it is possible, in fact quite common, for successive edge points close to the centroid to subtend such large angles that no edge point lies within tolerance of the required direction. In such cases the feature–radius defaults to zero.

The implications of this are that on different complete scans of a shape any short feature–radius may sometimes have a finite value and sometimes be zero. Then, when comparing the short feature–radii of a scanned shape with those of a candidate shape in the database, there is a high probability that they will not match since one will have a finite value and the other will be zero. The mismatch in this feature will increment the 'difference counter' (IDIF) for that candidate shape, and may cause its rejection. Thus these short radii are highly likely to increase the number of mismatches beyond that allowed for recognition.

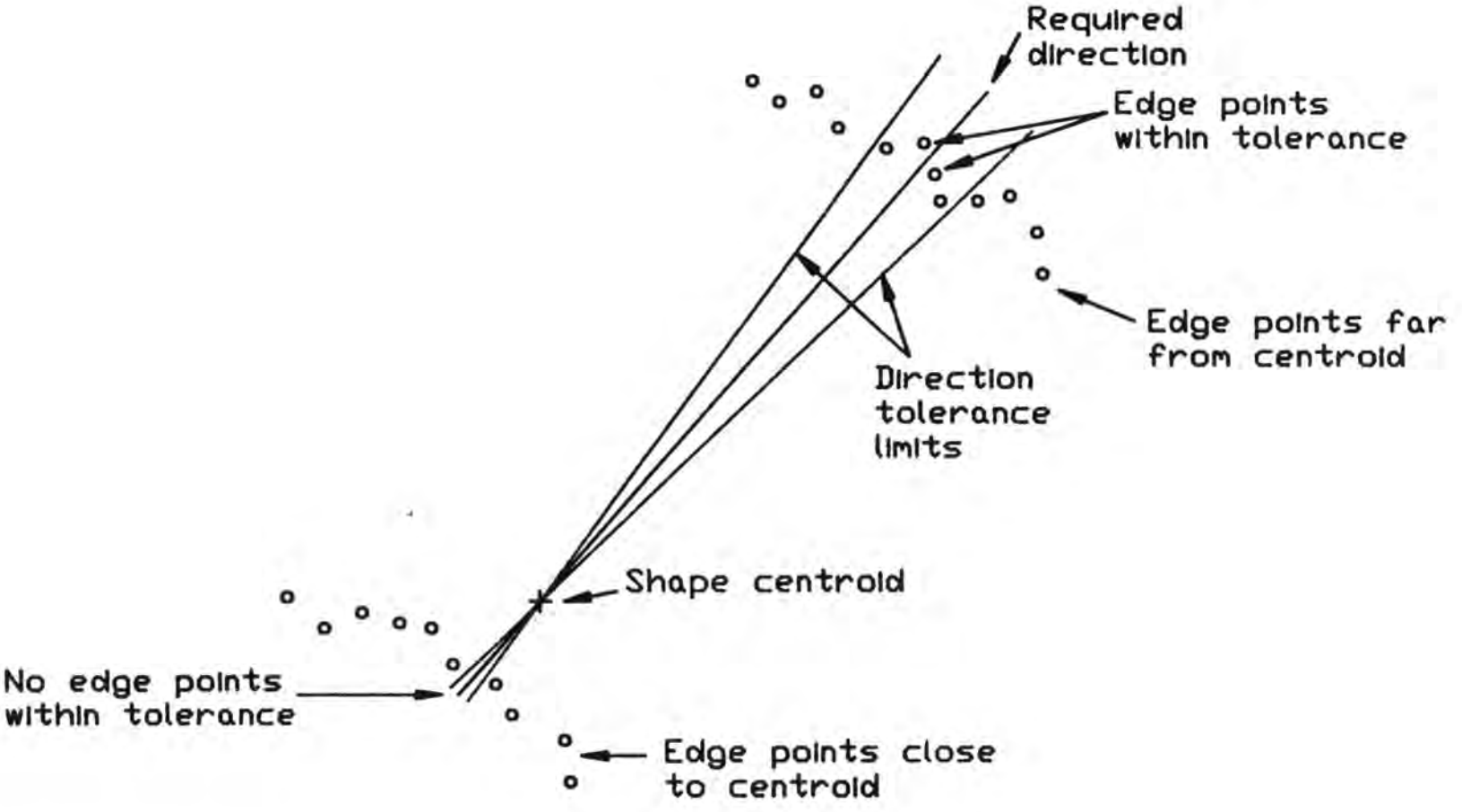


Figure 7.3 *The occurrence of a missing short radius.*

Shapes from the database with many small feature-radii are illustrated in figs. 7.4 & 7.5. Fig. 7.5 shows that such a shape does not necessarily have to be long and slender. Their feature–radii, copied from the database, are listed in table 7.2.

To eliminate this source of mismatches, it had been decided at BUSM to set all feature-radii of less than 80 pixels length to zero. However, this increases the rate of substitution, as can be appreciated by considering the resulting sets of feature-radii for narrow shapes, such as dr62. This shape is illustrated in fig. 7.4 and its feature-radii as initially produced by the 'teaching' scan and after setting those less than 80 to zero are listed in table 7.2. The information carried by those radii set to zero is much smaller than that of the other radii, and this shape is effectively using only 7 radii for recognition. The effect of this on the spacing of shapes within the feature space was determined by altering the program that found the minimum distances between shapes so that it set all radii less than 80 pixels long to zero. The results are summarised as example 5 in table 7.1, which shows that the number of shapes with small minimum distance to another shape is now very large, making their efficient recognition difficult.

One tactic which was suggested to improve the situation was to not set radii less than 80 pixels to zero, but if during recognition there were a mismatch and one of the radii were 0 then this would not be counted as a mismatch of these radii. This would allow the corresponding radii of the scanned and database shapes which were not zero to be used for recognition, but if one were zero and the other were not then they would not be used for matching.

Experiments were performed on 70 shapes which had several short radii. Attempts were made to recognise these in three different ways, with three runs of each shape with each method. The results were

i) with no radii set to 0, there were 16 substitutions and 13 rejections,

ii) with all radii under 80 pixels set to 0, there were 14 substitutions and no rejections,

iii) where any set of corresponding radii in which one was zero was not

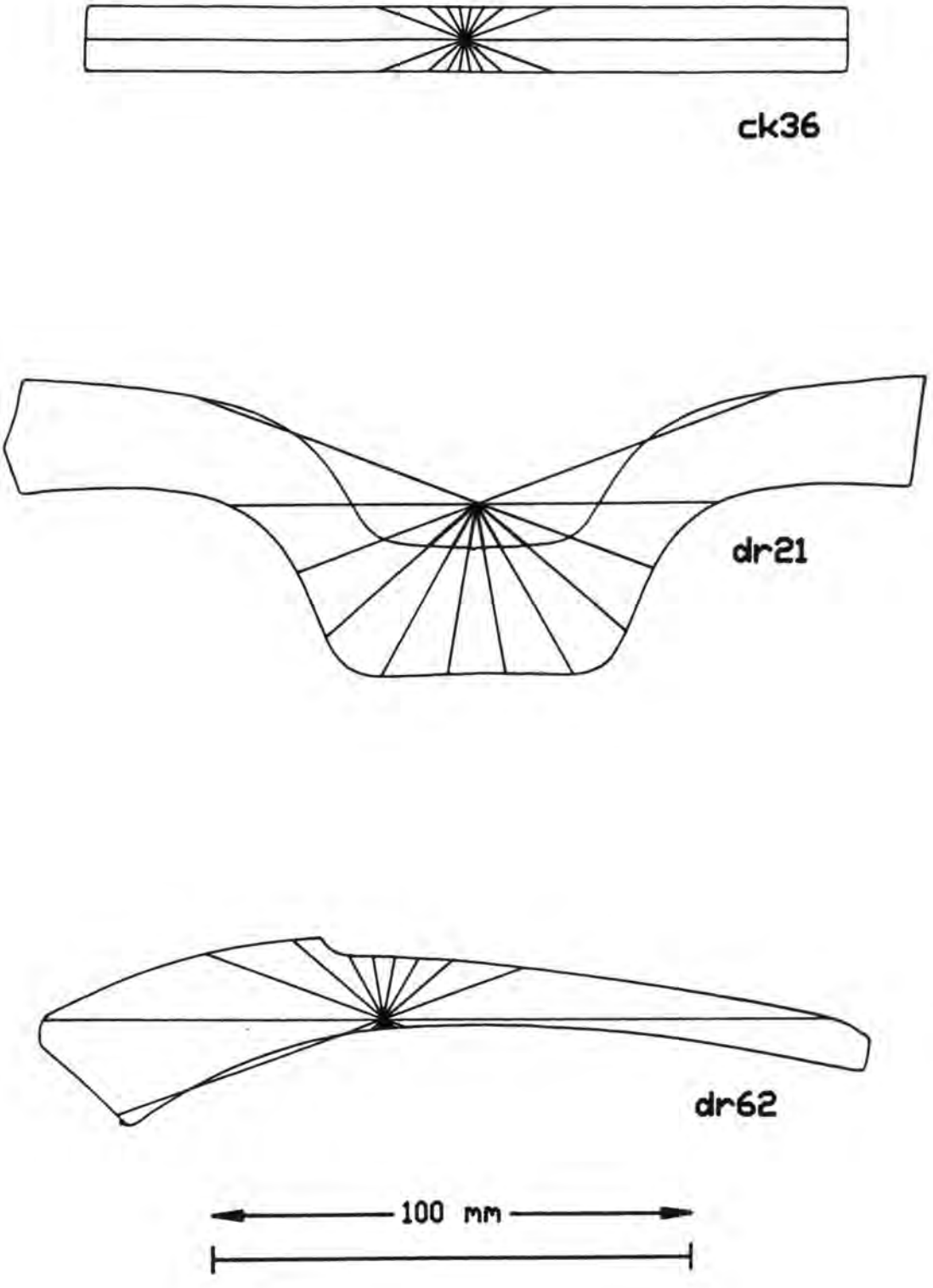


Figure 7.4 *Components with many small feature radii.*

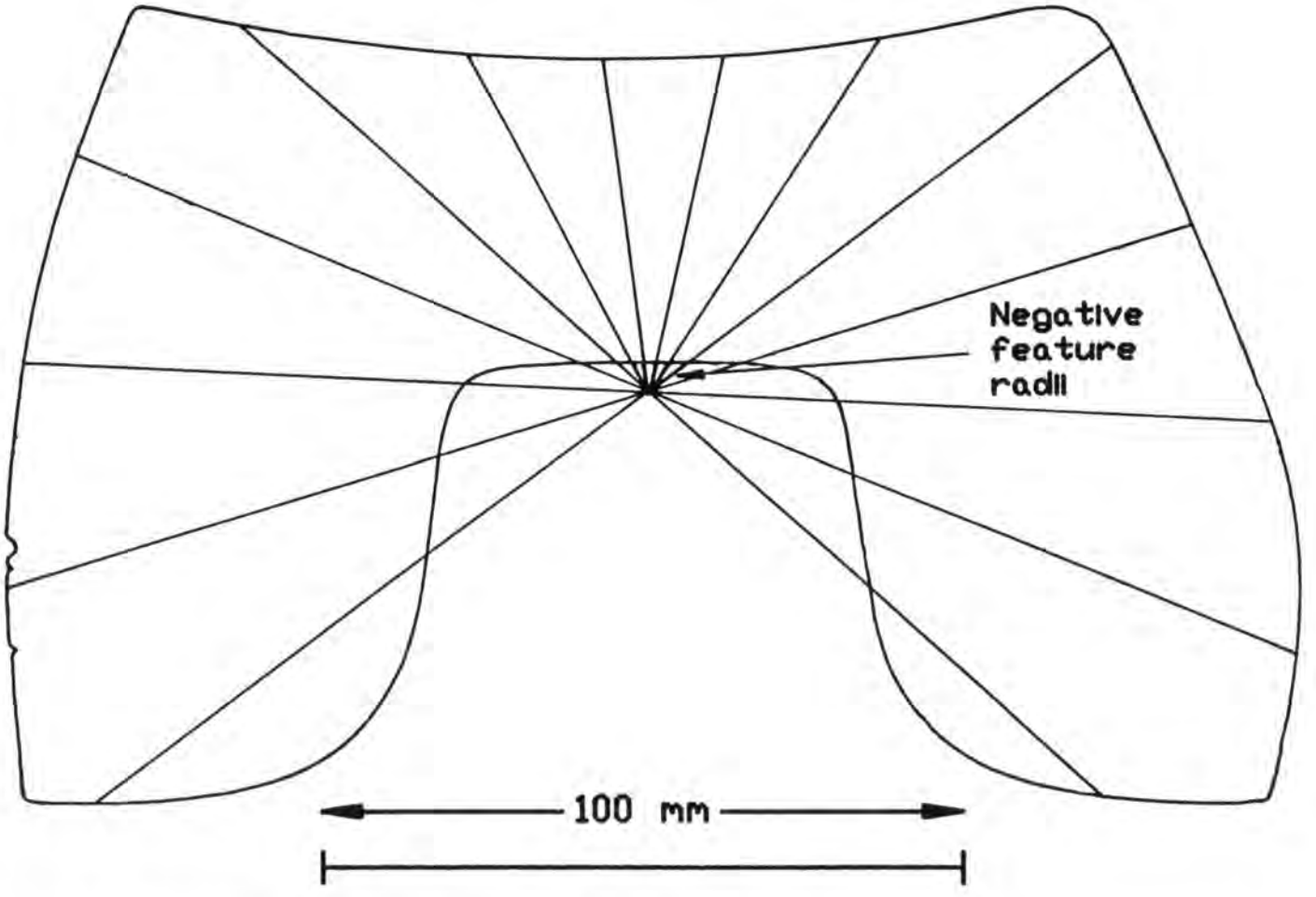


Figure 7.5 *Component with many small negative feature radii.*

Feature–radius no.	Length in pixels for each shape				
	ck36	dr21	gg101	dr62	dr62*
1	392	241	480	420	420
2	89	384	542	139	139
3	47	0	493	83	83
4	35	-69	-23	64	0
5	31	-38	-20	57	0
6	28	-38	-20	0	0
7	34	-44	0	70	0
8	48	-57	530	117	117
9	91	364	513	185	185
10	390	231	471	334	334
11	92	183	464	281	281
12	47	187	426	22	0
13	0	188	291	0	0
14	0	163	252	11	0
15	0	162	258	0	0
16	35	185	308	13	0
17	0	207	450	17	0
18	90	199	441	27	0

* After radii less than 80 pixels have been set to zero.

Table 7.2 *Shapes with many small feature–radii*

allowed to increment the difference counter there were 12 substitutions and no rejections.

This shows that there was some advantage in using the proposed method, though the results were still not perfect. Further improvements could have been made by scanning those shapes prone to having small feature-radii several times during the 'teach' process, ignoring zero radii, and averaging the non-zero ones for storing in the database. However, the single scan of an unknown shape might still produce zero radii. To improve this situation, one or more further tolerances could be added to the direction of a required radius. If no edge point lay within the close tolerance then a point lying within the greater tolerance could be used, if there were one.

7.5 Effects of altering tolerances

The substitution rate should be lowered by decreasing the size of the tolerances used in feature matching, though this should also increase the rejection rate. The effect of altering the tolerances was investigated with the large database by using the program for finding the minimum distances between the shapes with reversal of the radius set not allowed, and with short radii not set to zero, with the results summarised in table 7.3. The first four columns show the result of decreasing the radius tolerance from the standard value of 5 while keeping the other tolerances as standard. As expected, there is a large increase in the minimum distance between shapes. Columns 5 and 6 show that the effect of using smaller values on the other tolerances also has a large effect on the minimum distance.

As an additional test, all of the shapes were passed through the machine for recognition with the radius tolerance decreased to 3. As stated in section 7.2, when 1249 shapes were passed through (some twice) with the standard tolerances there had been no rejections and 29 (i.e. 2.3%) substitutions. With the radius tolerance reduced to 3, each of the 583 shapes were passed through once with 7 (1.2%) rejected and 7 (1.2%) substituted. So, as predicted, the substitution rate was reduced at the expense of a rise in the rejection rate.

Minimum distance	Number of cases of each minimum distance					
	Tol. 1	Tol. 2	Tol. 3	Tol. 4	Tol. 5	Tol. 6
0	6	0	0	0	2	0
1	12	8	2	0	6	0
2	17	9	7	0	8	6
3	16	16	4	0	4	0
4	17	15	11	7	8	4
5	18	18	16	12	2	8
6	6	14	19	6	2	6
7	5	14	11	13	0	2
8	12	9	14	17	12	4
9	7	9	11	19	6	4
10	4	6	4	8	4	2
11	0	6	12	16	0	12
12	6	5	7	7	6	8
13	6	2	2	6	4	0
14	47	40	26	18	53	22
15	36	27	31	34	38	35
16	209	229	238	217	248	277
17	3	5	12	45	1	14
18	0	0	0	2	0	0

The tolerances used were

Tol. 1 Root–area = 4, second moments = 4, radii = 5.

Tol. 2 Root–area = 4, second moments = 4, radii = 4.

Tol. 3 Root–area = 4, second moments = 4, radii = 3.

Tol. 4 Root–area = 4, second moments = 4, radii = 2.

Tol. 5 Root–area = 2, second moments = 2, radii = 5.

Tol. 6 Root–area = 2, second moments = 2, radii = 3.

Table 7.3 *The minimum distances between shapes as the tolerances are altered.*

One significant point to note is that all the 7 rejected shapes were 'bad' and all the substituted shapes were 'good'. Analysing the differences between 'good' and 'bad' shapes shows that of the 210 'bad' shapes that were passed through with the standard tolerances none were rejected and only 2 (0.95%) were substituted. With the decreased radius tolerance, of the 101 'bad' shapes passed through 7 were rejected and 0 substituted. The 'bad' shape technique appears to give a lower substitution rate, and could perhaps be used advantageously with 'good' shapes, though with a time penalty, since the 'bad' shape technique takes longer, as will be discussed in chapter 8. On the other hand, 'good' shapes may be more prone to rejection and substitution since they are long and thin and will have more radii approaching the outline at small angles, where a small variation in the calculated orientation of the shape will have a large effect on the values of the feature radii.

7.6 The distribution of shapes within the feature space

As well as the minimum distance between shapes it is also useful to have a knowledge of the distribution of the shapes within the feature space, since this will indicate for any given root–area the number of shapes that will pass through the binary decision tree and be candidate shapes for radius matching. The number of candidate shapes will govern the recognition time of a scanned shape.

To get an idea of the maximum recognition time it is necessary to have an idea of the maximum number of candidate shapes that there is likely to be for any value of root–area. The worst possible case would occur if all the shapes of the database were to lie within area and moment tolerance of each other, and thus be candidate shapes for radius matching - but this is highly improbable ! A more realistic projection of the worst case was obtained by analysing the 600 shape database. The 'good' shapes and 'bad' shapes were treated separately and their distribution within the feature space as defined by the root–area and the principal 2nd. moments of area was investigated.

Firstly the proportions of 'good' and 'bad' shapes within the database were determined. This revealed that of 598 different shapes, 120 had GCO-

EFF less than the threshold of 0.2, and hence were 'bad'. This number represented 20 % of the total. To assess how representative were the proportions of 'good' and 'bad' shapes within this artificial database, the components of some actual shoes were examined. The shoes comprised 3 pairs of ladies shoes and 5 pairs of mens shoes. The womens shoes were unassembled, and their GCOEFFs were measured by sending them through the machine, but the mens shoes were already assembled so the number of components that were 'bad' had to be assessed visually. The average number of different components in each pair of shoe uppers was 14.25, of which of 3.0 or 21 % were 'bad' shapes, which was reassuringly close to the proportion within the database.

The distribution of the 120 'bad' shapes was then investigated, with the results shown graphically in figs. 7.6, & 7.7. Fig. 7.6 shows the distribution of the 'bad' shapes within the area domain, with the majority, 114 of them, lying within a group with root–areas of values between 616 and 895. Fig. 7.7 shows the maximum number of candidate shapes for all values of root–area. That is, the shapes of each possible value of root–area were examined. For each value the number of shapes whose root–area and second moments matched within tolerance were counted and the maximum was noted. This gave the maximum number of candidates for that root–area, i.e. the maximum number of shapes that will pass the binary decision tree and become candidates for radius matching. The maximum number for any value of root–area was 6. Since it has been estimated, section 7.2, that a shoe factory might have a database that is 30 times larger than the one investigated here, then assuming that the distribution is similar to that of the this database, the maximum number of candidate shapes among the 'bad' shapes is likely to be about $30 \times 6 = 180$

The distributions of the 478 'good' shapes are similarly summarised in the graphs of figs. 7.8 and 7.9. From fig. 7.9 it can be seen that the maximum number of candidate 'good' shapes for any root–area is 10. Projecting to a database that is 30 times larger would give the maximum number of candidate 'good' shapes for any value of root–area as $30 \times 10 = 300$.

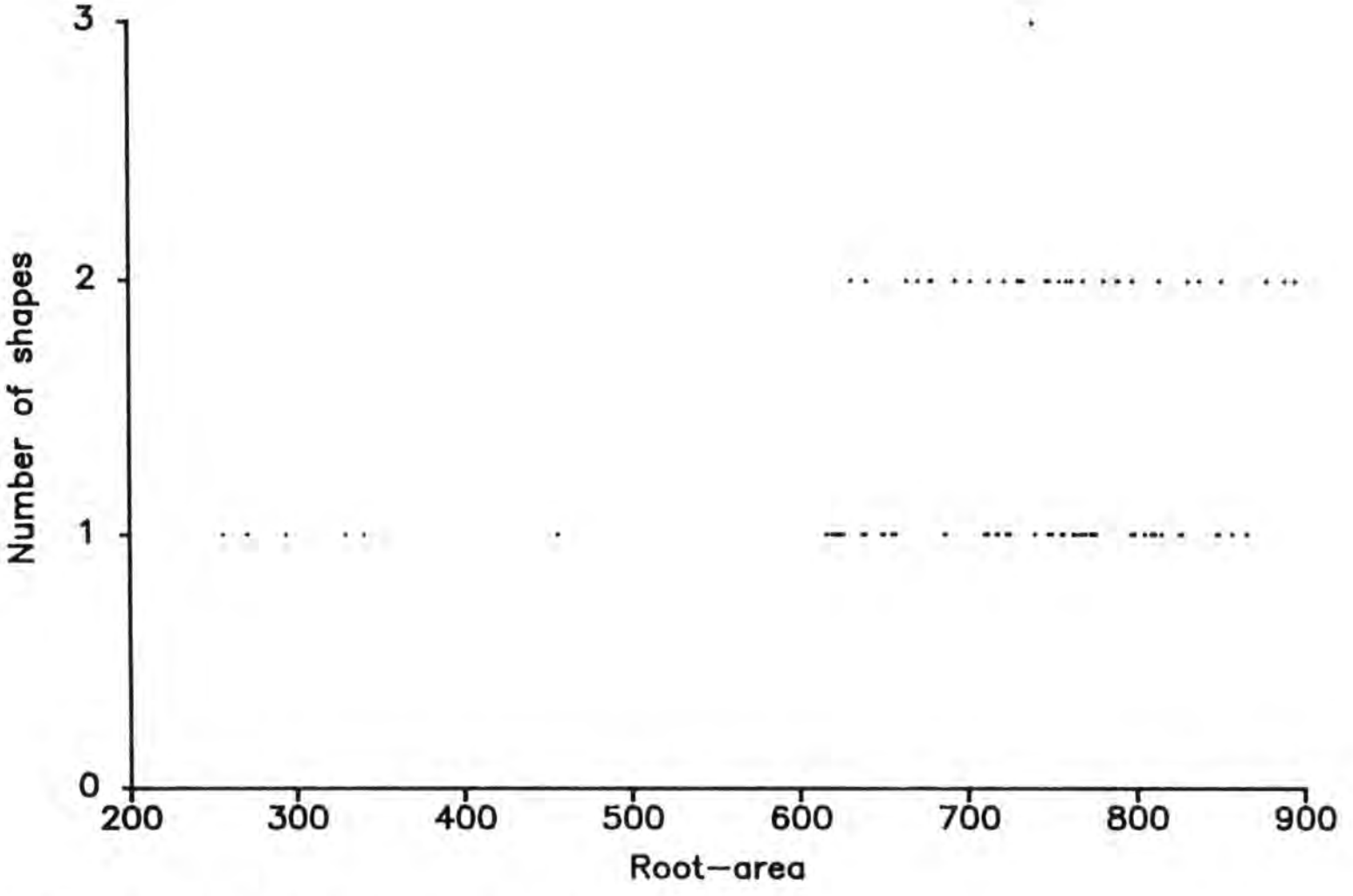


Figure 7.6 *Distribution of the 'bad' shapes within the root-area domain.*

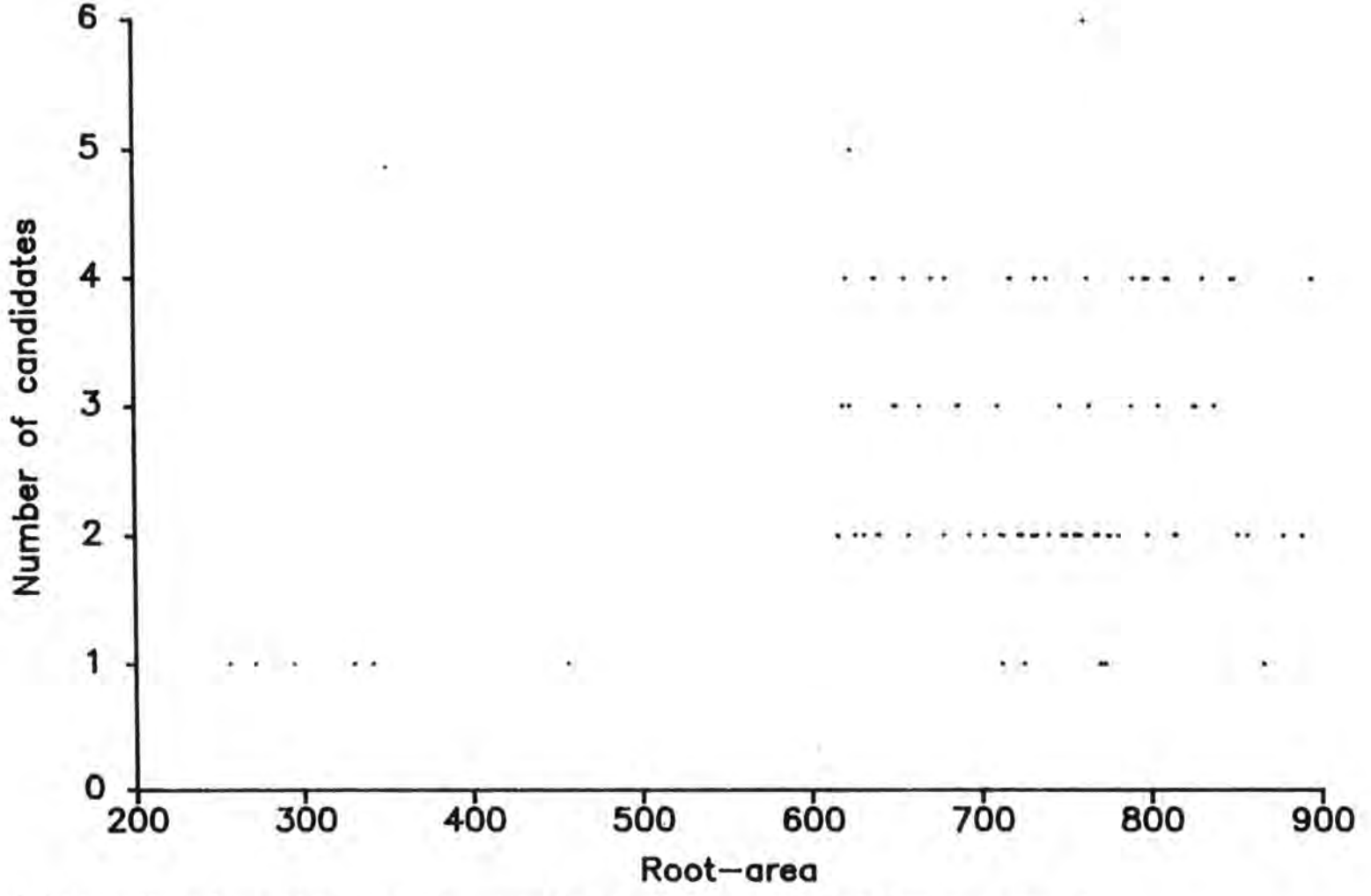


Figure 7.7 *Distribution of candidate, for radius matching, 'bad' shapes within the root-area domain.*

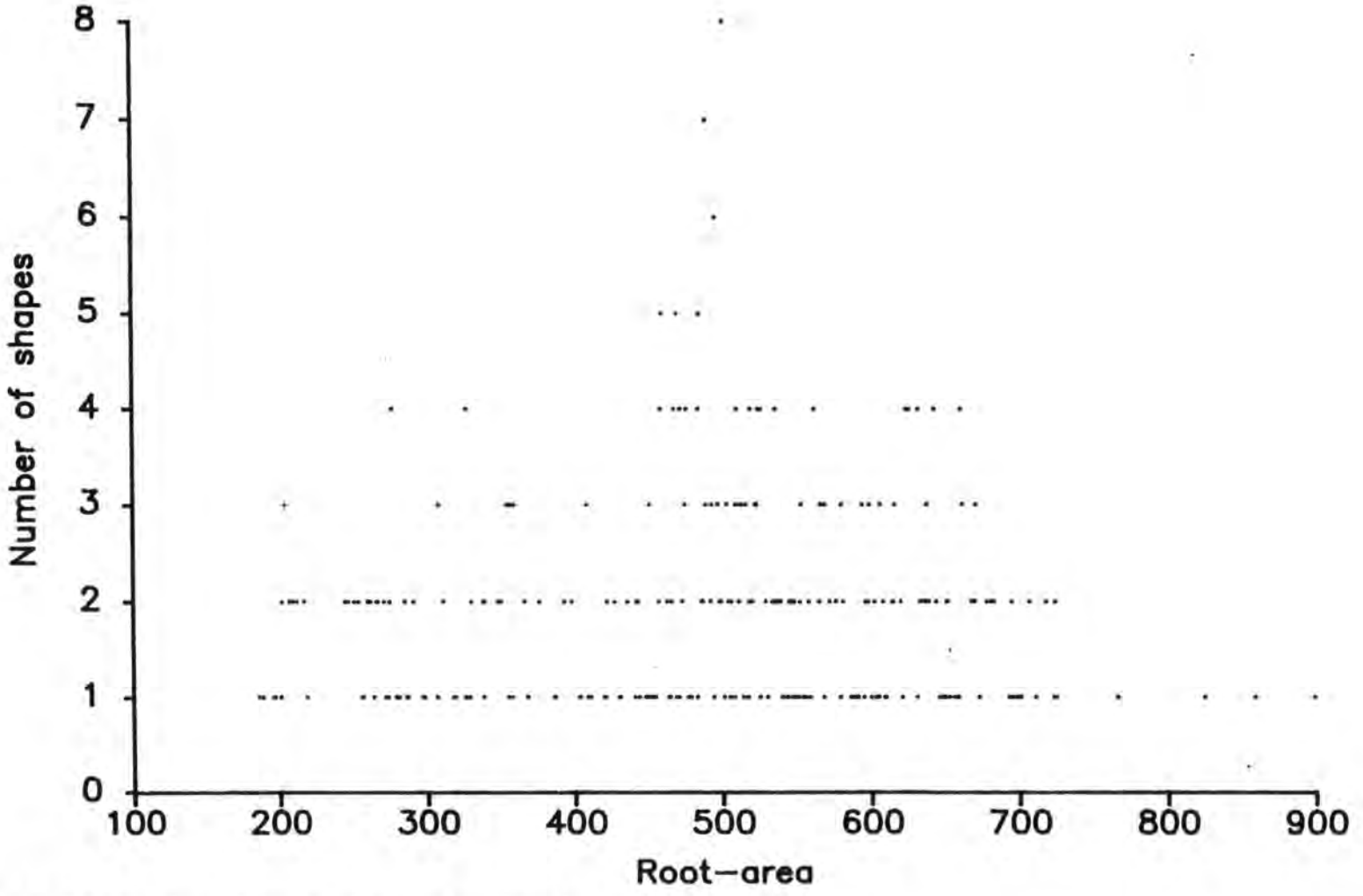


Figure 7.8 *Distribution of the 'good' shapes within the root-area domain.*

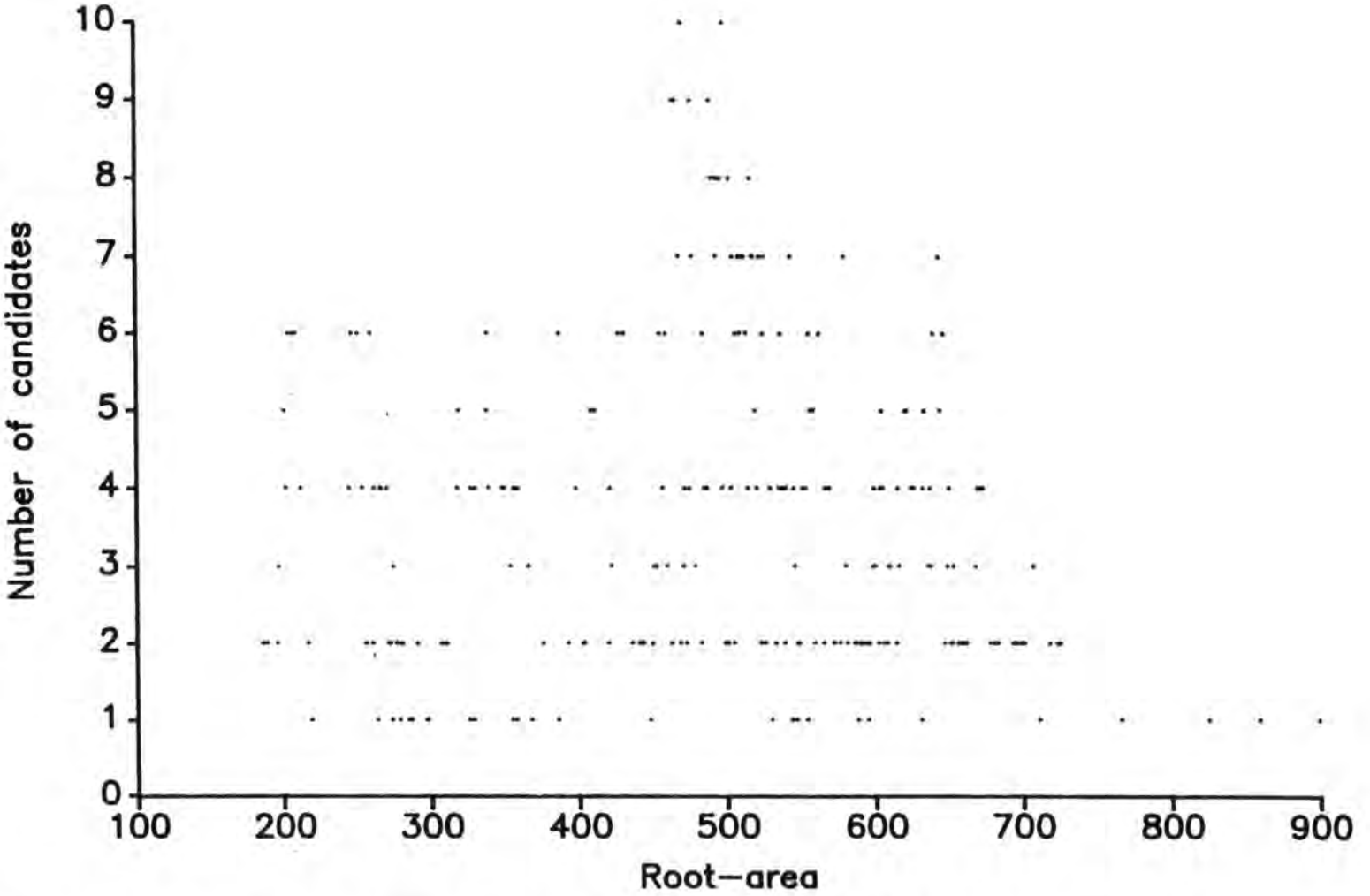


Figure 7.9 *Distribution of candidate, for radius matching, 'good' shapes within the root-area domain.*

7.7 Conclusions

The substitution rate depends primarily on the distance between the taught shapes within the database. What this will be for typical shoe factories is something about which we do not have much information. With the present tolerances on the 600 shape database the substitution rate is larger than desirable. More work is necessary to decide if these are the optimal tolerances for realistic factory databases. The substitution rate can be reduced at the expense of the rejection rate by reducing the tolerance of the features. It is possible that the 'bad' shape technique gives a lower substitution rate, and could perhaps be used advantageously with 'good' shapes, though with the time penalty discussed in chapter 8.

It is difficult to extrapolate results from this database to the much larger database of a typical shoe factory. However, it is likely that there will again be islands of close shapes all of the same design but differing from one another by half size. Problems would arise if two of these islands overlapped. The ideal way to analyse the shapes of shoe factories and the performance of the recognition system with them would be to take a machine to different factories and teach all of the shapes in production. However, this would take a lot of time and effort. Another way would be to make use of factories whose designs are stored on a CAD system and use this to analyse the distribution of the shapes within the feature space.

It could be argued that substitutions between shapes which are separated by 1/2 size and are almost identical are not too important since they would be stitchmarked almost identically, and would probably be stitchmarked conventionally with the same printing matrix. However the lefts and rights which are almost identical may cause problems if they are to be stitchmarked differently. With conventional stitchmarking the operator can distinguish between right and left by the mark in the corresponding side, but this is too small to influence the vision system much.

CHAPTER 8

MACHINE SPEED OPTIMISATION

8.1 Introduction

This chapter starts with a theoretical and practical timing analysis of the stitchmarking system. Methods of shortening processing times are examined and some of these are implemented. Considerations about overlapping the functions of recognition and printing are also discussed, and other aspects governing the throughput of the machine are examined.

A timing analysis of the machine is quite complicated due to the interactions between

i) the mechanical transport and scanning of the shoe components,

ii) the electronics and software of the vision and recognition system,

iii) the distribution of shapes within and the size of the database,

iv) the complexity of the stitchmarking being printed, and

v) the electronics, software, and mechanics of the printer.

This results in a large difference between the process times of the best and worst cases. A further complication is that the average throughput depends on the size distribution of the components passing through the machine, and their positions within the feature space of the shapes in the machine's database. Both of these will depend on the working arrangements

of individual shoe factories when using an automatic stitchmarking machine, which of course can only be conjectured at present.

However, the timings of the individual functions of the machine can be fairly straightforwardly determined and will be examined in turn. After dealing with the functions of the recognition and printing processes an estimation of the overall performance of the stitchmarking system will be made.

8.2 Timing analysis of the stitchmarking system

An analysis of the timing of the stitchmarking system was carried out by considering, where practicable, the code which was being executed. Where this was not practicable, and also to confirm the theoretical times, a Hewlett Packard 1610B logic state analyser was used to measure the process times of actual shoe components. For this, a range of components of various shapes was selected, including the extremes of small, large, 'good', and 'bad'. Representative times for performing each function of recognition and stitchmarking were thus obtained and are highlighted in bold faced text.

8.2.1 The shoe component transportation process

The shoe component transportation mechanism affects the time for recognition since the time to scan a component depends on the length of the component along the conveyor belts and the velocity of the conveyors. The belt velocity is limited by the maximum scanning rate of the camera and the maximum speed of the printer. Since it is desired that eventually scanning and printing will be performed on different components simultaneously, it is necessary for the conveyors to run at the same velocity during these processes.

The camera is driven at a clock rate of 4 MHz. Although the imaging array is quoted [69] as operating at up to a minimum of 5 MHz, and typically up to 8 MHz, use at higher speeds would require the fitting of faster access RAM to the edge point detection and buffer stage. The speed of the conveyor belts is tied to the scanning rate of the camera, as explained in section 5.3.

The 4 MHz clock produces a belt velocity of 183 mm/sec, which is limited by already being slightly above the normal operating speed of the printer.

The time required to scan a component is given by

$$Scanning\ time = \frac{Component\ length\ along\ the\ conveyor}{Conveyor\ velocity}$$

With the allowable range of component lengths from 36 mm to 333 mm, as discussed in section 5.2, this gives a scanning time range from 0.20 sec to 1.82 sec.

The time to scan components is 200 to 1820 msec

It is desirable that the horizontal distance between the scanning position and the printer is as small as possible to keep the length of the machine to a reasonable size. Since it was intended that the conveyor moves at constant velocity, without stopping, the minimum distance allowable between scanning and printing depends on the need to fully scan the longest component and the distance travelled by the component during the longest recognition time. The latter being proportional to the velocity of the conveyor causes a conflict of requirements since a faster belt would produce a longer machine.

In practice, the separation between scanning and printing is limited by the maximum length of the wide, toothed belt that was available, as detailed in section 5.2. This was shorter than desired and requires the belt to stop during recognition, otherwise the components would overrun the printer before recognition were complete.

8.2.2 The edge detection and feature extraction

The edge point detection is performed by hardware which keeps up with the rate of data output from the camera. This introduces negligible time penalty, though as explained previously the fitting of faster RAM would permit a higher camera scan rate, if this were necessary.

There is a time penalty in the way in which the edge coordinate buffer is utilised, since it can only hold the edge points of one shape at a time, and has to be cleared to zero before accepting another shape. The clearing to

zero is controlled by a program written in assembler language running on the Slave Z8000 microprocessor. With a system clock speed of 6 MHz this takes 27.35 msec.

The time to clear the edge point buffer is 27.35 msec

The feature extraction is also performed by assembler language routines on the Slave Z8000 microprocessor. These do not start executing until the whole shape had been scanned, though the summations for the area, 2nd. moments of area, and the centroid coordinates could be performed as the edge coordinates become available during scanning.

The times to perform these summations depend primarily on the number of left/right edge point pairs, rather than the number of top or bottom edges points. From a consideration of the code, the number of processor clock cycles required to produce the summations was calculated to be

$$1613 \times (\textit{No. of left/right edge point pairs})$$
$$+ 103 \times (\textit{No. of other edge points})$$
$$+ 11 \times (\textit{No. of camera scans})$$

The time to perform the summations varies with the number and distribution of the edge points of a shape, so theoretical times are not easily calculated. Instead, measurements were made using the logic state analyser on actual shoe components whose details are listed in Table 8.1.

Shape type	Shape name	GCOEFF	Typical number of edge points
'good'	bg1	0.32	6603
'good'	gg104	0.55	3295
'good'	gg155	0.59	1852
'good'	dr25	0.81	2668
'bad'	gg23	0.14	3948
'bad'	dr34	0.03	1700

Table 8.1 *Details of shapes used in timing the recognition processes.*

With a system clock of 6 MHz, the time to complete the summations was 128 msec for the small shape dr34, with 1700 edge points, and 743 msec for the large shape bg1, with 6603 edge points. The remaining shapes took times in between.

The measured time for the summations for the area, 2nd. moments, and centroid of representative shapes is 120 to 743 msec

The remaining feature calculations have to be performed when the whole of the shape has been scanned.

The times required to complete the calculation of the area, 2nd. moments of area, and the centroid coordinates could not be easily determined by examining the code due to the large number of floating point routines with execution times dependent on the actual numbers involved. These times were, hence, again measured using the logic state analyser with the same selection of shapes. The ranges of times measured are listed in table 8.2.

Calculation	Execution time (msec)
Summations for area & 2nd. moments	0.12 – 0.74
Square root of area	0.41 – 0.43
Centroid coordinates	0.94 – 0.95
Absolute 2nd. moments of area	2.45 – 2.48
2nd. moments relative to centroid	1.23 – 1.25
Mohr's circle (for principal axis)	1.32 – 1.34

Table 8.2 *Range of measured times for root-area, principal second moments, and centroid calculations.*

The calculation of the orientation of the principal axis of a shape varied more widely, and gave a range of measured times for the shapes listed in table 8.1 of 3.91 msec to 5.83 msec.

The time to calculate the orientation was measured to be from 3.9 to 5.8 msec

The set of 18 feature radii is calculated by firstly determining the gradients of the required radii. Each edge point is then examined to see if the gradient of the line joining it to the centroid lies within tolerance of any of the required gradients. The longest such distances to the edge points are recorded as the feature radii. The time for finding the 18 radii was measured in the range 229.8 msec, for shape dr34 with 1700 edge points, to 705.6 msec, for shape bg1 with 6603 edge points.

The time to calculate a set of 18 feature radii of representative shapes was measured to range from 230 to 706 msec

The calculation of the radii completed the set of features of the scanned shape ready for matching. These results show that two processes take much longer than the others. These are the time for the summations for the area, 2nd. moments, and centroid, which takes from 128 to 743 msec, and the time to determine the set of feature radii, which takes from 230 to 706 msec. In both cases each edge point of the scanned shape has to be examined and acted upon.

8.2.3 The feature matching

The feature matching was performed by Fortran routines on the Compaq personal computer. The time taken for this depended on whether the shape was 'good' or 'bad', since these were treated differently, as described in section 5.8. It also depended on the density of distribution of shapes within the feature space around the scanned shape, as was discussed in chapter 7.

Tables 8.3 & 8.4 show the measured times of feature matching for 'good' and 'bad' shapes. Note that the time taken to compare the features of a 'bad' shape is much longer than that for a 'good' shape. This is because the set of 18 feature radii of a scanned 'bad' shape are compared with all 360 radii of each candidate shape in the database, as explained in section 5.8.

Operation	Time (msec)
Read the features of a candidate shape from disc	6 to 33
Compare features	0.38
Repeat for all candidate classes.	

Table 8.3 *Measured times for matching 'good' shapes.*

Operation	Time (msec)
Read the features of a candidate shape from disc	6 to 33
Compare features	140
Repeat for all candidate classes	
Interpolate radii for matching class	21

Table 8.4 *Measured times for matching 'bad' shapes.*

8.2.4 The stitchmark generation and printing

The time taken to transform the stitchmark data and generate the stitchmarking from it, ready for printing, depends on the amount of data, the number of arcs through which the stitchmarking has be interpolated, and the speed of the hardware. The stitchmark data is read from the Winchester disc and loaded into the TMS34010 Software Development Board (SDB) by 'C' language routines. The time required for this is constant at 20 to 30 msec, since the whole of the small amount of data could be read into the disc controller's cache in one go.

The transformation of the stitchmark data so that it prints in the correct position on the component and the interpolation through the transformed points is performed in 'C' and assembler language on the SDB by the TMS34010 graphics processor, as described in section 5.9. The timing of the various functions was measured by inserting into the 'C' code at the relevant

points a TMS34010 assembler language instruction which produced a pulse on an address line of one of the peripheral decoding integrated circuits on the SDB, which was not otherwise used. The pulse was very short (2.8μsec) compared with the times being measured.

The measured times for the stitchmark generation operations are listed in table 8.5.

Operation	Time (msec)
Sine of angle of orientation (non–zero)	2.66
Cosine of angle of orientation (non–zero)	2.90
Transforming the coordinates	$0.96(CRDPRS + 2NCURVS)$
Interpolating the curves	$3.92(CRDPRS - NCURVS)$
Drawing the curves in RAM	$18.02(CRDPRS - NCURVS)$
Transferring the first slice from general purpose RAM to VRAM	50.55

CRDPRS is the total number of taught coordinate points for the stitchmarking.

NCURVS is the number of separate taught lines of stitchmarking.

Table 8.5 *Measured times to transform the data and generate the stitchmarking.*

The times shown total to $5.7+(22.9\times CRDPRS)-(20.0\times NCURVS)$ msec to generate the stitchmarking, plus 50.5 msec to transfer the first slice of the image to VRAM. Table 8.6 shows some examples which illustrate the order of time that this equation gives.

Stitchmarking to be generated	Time (msec)
1 curve with 2 points	31.5
2 curves each with 2 points	57.3
1 curve with 2 points & 1 curve with 3 points	80.2
3 curves each with 5 points	289.2

Table 8.6 *Examples of times to generate typical stitchmarking.*

An extra 50.5 msec is then required in each case for transferring the first slice to VRAM ready for printing.

These times can be compared with the time it would take to transform the stitchmarking if every point of it were stored on disc, rather than storing only the few points and interpolating. The average length of stitchmarking on shoe components has been estimated to be 200 mm [62]. At the resolution of the image in the general purpose RAM of 3 dots/mm (75 dots/inch) this corresponds to between 424 and 600 dots, depending on the orientation of the stitchmarking with respect to the grid of printed pixels. The time to transform these would be $0.96 \times (No.\ of\ points)$ msec, as for the third item in table 8.5, which is between 407 and 576 msec. This is longer than the likely time required for interpolation, and added to this would also be any extra time required to load all of the coordinates of these points from disc onto the SDB. So, it is advantageous to use the interpolation method.

For simplicity, the printing program was written initially so that the printer always printed out the full contents of the image area of the general purpose RAM, even though this was blank after the end of the stitchmarking. This meant that a total of 8 lots of the VRAM's 512 lines were always printed, which covered a length of 347 mm along the belt. Since the printer was synchronised to the belt which moved at 184.7 mm/sec, the printing time was always 1.88 sec.

The time to print the whole contents of the Image RAM is 1880 msec.

After the stitchmarking has been printed the image area in the general purpose RAM has to be cleared ready for the generation of the next image, which takes 222 msec.

The measured time to clear the Image RAM is ~~200 to 2130~~ 222 msec.

8.2.5 The overall timing of the machine

The overall timing of the machine is best seen by reference to an example. The timing for a shoe component of length L along the belt, whether 'good' or 'bad', to be stitchmarked is shown in fig. 8.1. The times in the

figure are derived from the maximum times given in the preceding sections and are summarised in table 8.7.

The total times for stitchmarking (excluding the time to clear the Image RAM) with a conveyor belt velocity of 183 mm/sec are given below. Note that the length of the component cancels out.

The typical time to recognise and stitchmark a 'good' shaped component is $5.7 + (0.03 \times CAND)$ **sec.**

The typical time to recognise and stitchmark a 'bad' shaped component is $5.7 + (0.14 \times CAND)$ **sec.**

It will be appreciated that as the feature space becomes more densely packed so the value of *CAND* will become larger and the time taken to recognise 'bad' shapes will become much greater than that of 'good' shapes.

As can be seen from fig. 8.1, the time allowed for generating and transforming the stitchmarking (S) is the time taken for the conveyor to be accelerated to constant velocity, and for the shoe component to reach the line corresponding to where the print image is deposited on the drum. This gives a time of $0.22 + (333 - \textit{Component length})/(\textit{Belt velocity})$ sec. An adequate time for generating typical stitchmarking is 0.3 sec, as listed in table 8.7, which allows the printing of components up to $333 - (183 \times (0.3 - 0.22)) = 318$ mm long along the conveyor.

An appreciation of the speed improvement from the arrangement with the laser printer to that with the ionographic printer can be obtained by considering the recognition and printing times for printing similar stitchmarking on the same component with the two systems. The timing for the small 'good' shaped component gg104 was discussed in section 4.5 and listed in table 4.1 which is reproduced as table 8.8. These should be compared with times for the system with the ionographic printer which are listed in table 8.9.

The time for recognising and printing this small 'good' component with the system with the laser printer had been 22.5 sec, whereas with the system with the ionographic printer the equivalent time till the component

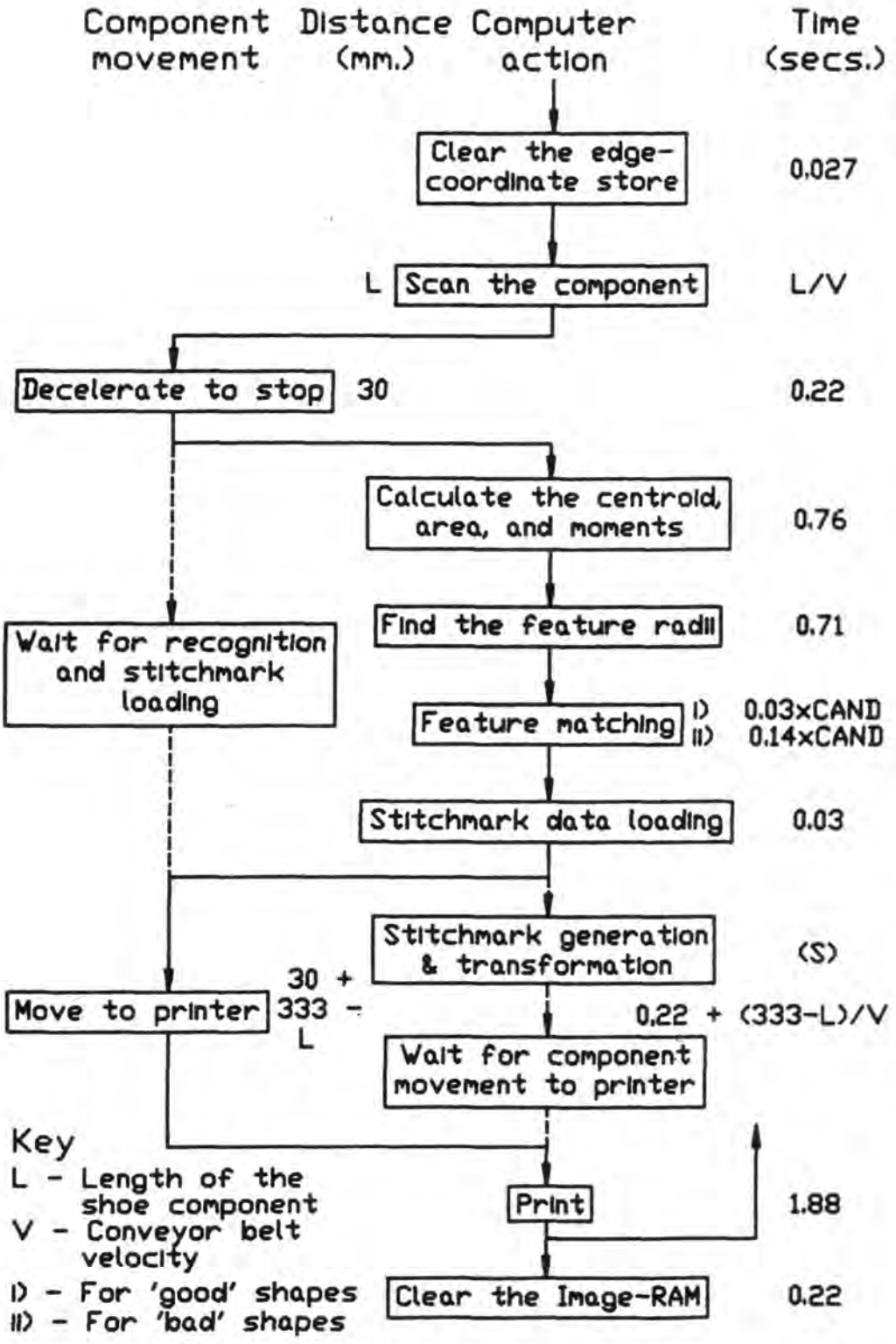


Figure 8.1 *Overall timing of the stitchmarking system.*

Operation	Time (seconds)
Clearing the edge–coordinate RAM	0.027
Scanning the shape	L/V
Deceleration to stop	0.22
Calculating area, moments, centroid, angle	0.76
Finding the 18 feature radii	0.71
Matching the features :– a) 'Bad' shape	$0.14 \times CAND$
b) 'Good' shape	$0.03 \times CAND$
Loading the stitchmarking data	0.03
Acceleration	0.22
Generation and transformation of the stitchmarking (3 curves, each with 5 points) and transferring 1st slice	0.34
Moving the component to the printer	$(D - L)/V$
Printing	1.88

CAND is the number of candidate shapes in the database which match the scanned shape's area and 2nd. moments and who's radii consequently have to be checked for matching.

D is the distance between scanning and the position on the takeup belt that a point on a shoe component reaches when the print image for that point is being deposited on the drum. This is 393 mm, but is further reduced by the distance the component moves during the belt's deceleration and acceleration (30 mm each), and gives a figure of 333 mm.

V is the velocity of the conveyor belts, which is 183 mm/sec.

Table 8.7 *Representative times for the functions of recognising an average shoe component and printing average stitchmarking.*

Operation	Time (sec)
Scan & recognition	3.5
Reading stitchmarking & sending coordinates to Master	2.7
Transforming coordinates	0.5
Transmitting coordinates to printer	3.9
Waiting for printer to start printing	5.0
Printing	7
Giving a total time of about	22.5

Table 8.8 *Time for printing stitchmarking on 'good' shaped component gg104 using the PE-1620 computer for feature matching, the Master Z8000 for stitchmark generation, and the Hewlett Packard laser printer.*

Operation	Time (sec)
Scan & recognition	2.9
Loading stitchmarking onto the SDB	0.03
Generating stitchmarking for 2 curves of 4 points each	0.15
Moving the component to the printer and printing (on the component only)	3.0
Giving a total time of about	6.1

Table 8.9 *Time for printing stitchmarking on 'good' shaped component gg104 using the Compaq 386 for feature matching, the TMS34010 Software Development Board for stitchmark generation, and the ionographic printer.*

emerged printed was 6.1 sec. Although there had been some improvement in the recognition time by the use of the faster 32–bit processor and Winchester disc drive, the main areas of improvement were the generation and transmission of the stitchmarking and the printing.

8.2.6 The worst case timing

To get an idea of the worst case timings for the machine it is necessary to have an idea of the value that CAND might reach. This is a subject which was investigated in section 7.6. There it was projected that for 'good' shapes CAND might reach 300 and for 'bad' shapes CAND might reach 180.

Using the equations given in table 8.7 the maximum time required to search through the candidate 'good' shapes will be $300 \times 30 = 9000$ msec, and the time to search through the candidate 'bad' shapes will be $180 \times 140 = 25200$ msec, which is 25.2 sec.

Thus, it can be seen that with a large database the maximum time to search through the candidate 'bad ' shapes will be an appreciable bottleneck. An investigation of ways of speeding up the recognition of 'bad' shapes is described in chapter 9.

8.3 Optimising the speed of the stitchmarking system

8.3.1 The recognition process

There are four principal methods of improving the throughput of the recognition process

i) by using alternative, faster algorithms,

ii) by using faster executing code,

iii) by using hardware which executes the same code faster, and

iv) by overlapping processes so that several components can be processed simultaneously.

Items (i) to (iii) will be discussed here, while item (iv) will be discussed in section 8.4.

In some cases there is scope for improving the algorithms used for the recognition process. For example the feature matching algorithms, which are written in Fortran and ran on the Compaq 386, have been altered extensively for experimental work, which has left inefficient code, and have had little optimisation performed on them. The execution time could be immediately reduced by rewriting the programs in the more code efficient 'C' language, or even assembler code. At the same time the inefficient algorithms could be improved.

On the other hand, the feature extraction routines which are written in Z8000 assembler language have been extensively optimised at BUSM, to the point where, for example, fast in–line code versions of some arithmetical operations were written, optimised for specific calculations, rather than using the general purpose library functions. Improving the execution times of these algorithms by modifying them is very difficult.

The processes could be speeded up by using faster system clocks for the microprocessors, with the corresponding change in the microprocessor speed. For example the Z8000 microprocessors running with a 6 MHz clock could be replaced by versions capable of running at 8 or 12 MHz, though this may require the use of faster versions of the other devices on the microprocessor boards. Totally different, faster microprocessors, such as the AMD 29000 processor or the transputer, could be used for the recognition processes, though these would require a lot of costly redesign. One other possibility is to use add–on boards containing additional processors within the Compaq 386. For example, boards containing stand alone transputers are available. These could be used for feature extraction and matching, with the Compaq's 80386 processor being used for overall machine control and database management.

Other possibilities for speeding up the recognition process include copying the feature databases from disc to memory when the machine is started up. This would remove the need for reading the disc during match-

ing. However, the maximum likely database size, as estimated in section 7.2, and the storage requirements for each 'good' and 'bad' shape, as listed in section 5.7, would demand the large storage requirement of 6 MBytes.

This large storage requirement could be reduced by utilising a RAM cache for all of the recently recognised shapes, since they would be likely to appear again soon. Additionally, all of the shapes of a design of shoe could be loaded into the cache once any one of them had been recognised, in anticipation that the rest of the design was likely to be processed. However, care would have to be taken to ensure that the overheads in managing the cache do not outweigh the advantage of fewer disc reads.

8.3.2 The generation and printing of the stitchmarking

The time for stitchmark generation on the TMS34010 Software Development Board (SDB) using cubic interpolation is short enough to give overall recognition and printing times within the specification of the experimental machine. For example, to generate a typical stitchmark pattern of 3 curves each with 5 points takes 290 msec. However, in order to reach the ultimately required rate of two to four components per second requires much faster calculations. The stitchmark generation and printing programs had been written in a great hurry to be ready for the exhibition machine, so they were examined to see how they could be optimised.

One obvious change was in the routine 'spline_draw' which set the bits in the general purpose RAM corresponding to the dots to be printed. In this routine the values of 0.1^2 to 0.9^2 and 0.1^3 to 0.9^3 were calculated every time. This was altered so that the values were calculated at compile time and inserted into an initialised array, so they did not have to be calculated by the routine at run time. For 1 curve with 2 points this reduced the execution time for this function from 21.82 msec to 18.02 msec, so that the overall time for stitchmark generation was now given by

$$18.02(CRDSPRS - NCURVS),$$

where $CRDSPRS$ is the total number of coordinate pairs in all the curves, and $NCURVS$ is the number of curves.

All of the arithmetic for stitchmark generation is performed using the standard 'C' language library's double–precision functions. If single–precision routines are used then there would be a reduction in time for the calculations. Unfortunately the single–precision functions provided are not compatible with the standard 'C' function calls. However, by using variables declared as registers it was possible to pass parameters to these functions.

By measuring the difference in execution times between the single– and double–precision routines the speed increase that could be obtained by changing to the former was estimated. However, complications arise because

i) the execution time of the functions depends on the values of the numbers involved, and

ii) the compiler optimises the generated code if there are several of the maths functions on one line by leaving the result of one maths function on the stack if it is to be used by a following one.

To obtain some idea of the likely speed increase, the longest execution times were used, and the times for the double–precision functions were measured from the time after the parameters had been pushed onto the stack to the time before the result had been popped off. The results for the most commonly used multiplication and addition functions are shown in table 8.10.

Operation	Double precision	Single precision	Ratio
Multiplication	110 μsec	37 μsec	3.0:1
Addition	108 μsec	43 μsec	2.5:1

Table 8.10 *Times for double– and single–precision maths functions*

It can be seen that the single–precision routines are in the range of 2.5 to 3.0 times faster than the double–precision routines. Since stitchmark generation involves primarily floating–point maths, changing to single–precision arithmetic is likely to speed up the routines by a factor of about 2.5, though the programming will be more complicated. By this means, the time to generate the typical stitchmark pattern of 3 curves each with 5 points will be usefully reduced from 290 msec to about 120 msec, though the effect

of the reduction in precision on the accuracy of printing would have to be investigated.

Eventually, more video memory will be provided, allowing the stitchmark image to be directly generated there. This will remove the need to transfer slices of the image, so saving the 50.55 msec that this requires. The overall time for stitchmark generation will then be suitable for use in a system stitchmarking several components per second.

8.3.3 Optimising the overall operation of the machine

As mentioned in the introduction to this chapter, it is difficult to pin down a universal overall time for the automatic stitchmarking process. There are several reasons for this. Firstly, the distance between scanning and printing has to be fixed. Its minimum distance depends on the length of the longest shoe component to be stitchmarked and the maximum recognition and stitchmark generation time. The maximum recognition time will probably occur for a medium sized 'bad' component in a dense region of the feature space, but the actual maximum time cannot be determined, since this would vary from shoe factory to shoe factory.

To overcome this problem of not knowing the maximum recognition time, the distance between scanning and printing could be made much longer than would ever likely be needed. This would make the overall process time very long, but the average throughput could be increased by allowing several small, quickly recognised components to be travelling from scanning to printing simultaneously. In this way, some of the stitchmarking processes would be performed concurrently.

8.4 Concurrent processing

The specification of the requirements for the stitchmarking machine resulting from this project, see section 1.5, states that processes of the system should be designed to ultimately enable concurrent operation for greater throughput. This was implemented by configuring the system with independent processors for machine control, edge detection, feature extraction,

feature matching, and printing. However, when the detail of overlapping the processes was considered, it was found that the large range of execution times of some of the processes, combined with the constant velocity of the shoe components through the machine, made this a complicated operation.

Consider the stitchmarking system as outlined in fig. 8.2 (i) & (ii). In the ideal case of fig. 8.2 (i) every process executes in the same time on every shoe component. When all of the processes have completed then each can pass on the parameters that it has obtained to the succeeding process. If, however, one of the processes, for example the feature matching, were to take a long time for a certain type of component, then, as shown in fig. 8.2 (ii), that process would not be ready to accept the next shape from the preceding process, and a log jam would ensue. The implications of this are that early, fast processes are restricted by later slower ones.

This could be overcome by restricting the interval between the arrival of shoe components to the maximum process time of all of the processes. The throughput of the system would then be drastically reduced, however. This is not really feasible, since the feature matching is likely to have the maximum process time, but no figure can be put on this since it will depend on the distribution within the shape database of individual shoe factories.

Another approach to this problem would be to have a data buffer between the processes, so that, for example, the features of several shapes could be stored, waiting for matching while the features of the following shapes are being extracted. There would, however, be two problems with this. Firstly, since the shoe components are arriving at random there will be the possibility that dozens of small quickly processed shapes, but requiring a long time for feature matching, might arrive one after the other, and cause overflow of the data buffers. The second problem would be that the components would always be travelling at constant velocity, so the length of the machine for the worst case would be likely to become very long.

A further possibility is that the feed conveyor belt would normally be stationary, and only run to feed another component when it was predicted that the processes had sufficient capacity to deal with it. This would be

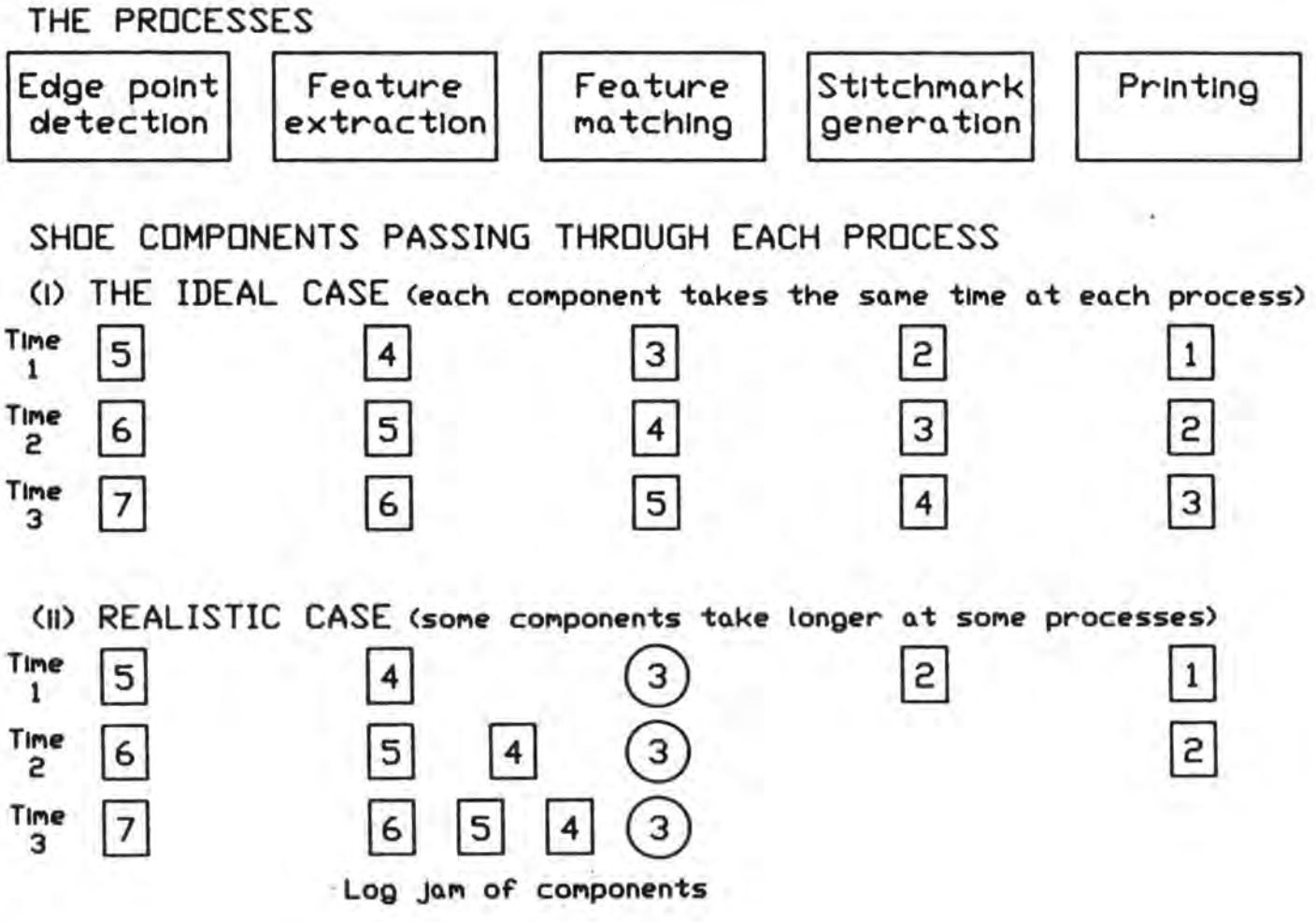


Figure 8.2 *Ideal and real concurrent processes.*

determined by polling the status of each process and its associated buffer. Even this would not be perfect, however, unless some prediction capability were built into the polling. Probably the most workable method is to accept that occasionally certain processes would become overloaded, and at such times some shoe components passed on from previous processes would have to be ignored and be recycled through the machine.

8.5 General considerations

The speed might could also be increased by other means. For example the printing resolution of 11.8 dots/mm (300 dots/inch) gives higher quality printing than is required for stitchmarking, so there would be scope for increasing the speed by reducing the amount of data that has to be handled in the stitchmark generation. However, reducing the resolution also reduces the accuracy of the printing of individual points of the stitchmarking, as is discussed in section 6.5.2.

8.6 Conclusions

The throughput of the automatic stitchmarking system was within the specification of a component every 5 to 10 seconds, though the times for some 'bad' shaped components might become longer than this. There is scope for optimising the algorithms of the feature matching process and rewriting in a more efficient language. Otherwise speed increases could be obtained by using faster processors. Overlapping of processes to give concurrency is possible, but is not easy to implement reliably. However, ways around this have been indicated.

The major bottleneck that was foreseen would be in the feature matching of 'bad' shapes in dense areas of feature space within a large shape database, where the number of candidate shapes for radii matching would be high. Methods of overcoming this problem were investigated, and are reported in chapter 9.

CHAPTER 9

ACCELERATING THE PROCESSING OF 'BAD' SHAPES

9.1 Introduction

From the estimate of the likely number of candidate 'bad' shapes of section 7.6 and the timing analysis of chapter 8 it can be appreciated that the time to examine each candidate 'bad' shape is a critical element in the automatic stitchmarking process. In an attempt to overcome this potential bottleneck methods of speeding up the processing of 'bad' shapes were investigated. However, none of the methods investigated were found to be better than the polar coding and correlation technique, already employed, in terms of speed and accuracy.

9.2 The cause of the bottleneck with 'bad' shapes

The bottleneck in the recognition of 'bad' shapes is caused by the time taken to examine each candidate shape (i.e. database shapes whose root–area and principal 2nd. moments match within tolerance those of the scanned shape, and whose radii must be checked for matching) and the large number of likely candidate shapes.

In section 7.6 it was estimated that when stitchmarking with a typical 18,000 shape database there could be scanned 'bad' shapes which have 180 candidate shapes. Each of these would require comparison of radii by the polar coding and correlation method, as described in section 5.8. For 'good'

shapes, which were 4 times more abundant, it was estimated that there could be 300 candidate shapes, i.e. not quite twice as many as the 'bad' shapes.

The reason why the number of candidate shapes was proportionally more for the 'bad' shapes is that the area of a shape and its principal 2nd. moments are not completely independent. Both of the 2nd. moments of 'bad' shapes tend towards the value $\frac{1}{4}\pi r^4$, where r is the radius of the circle with the same area as the shape. This increases the density of shapes within the feature space, based on area and 2nd. moments, around 'bad' shapes.

Use of the polar coding and correlation method of comparing the set of 18 feature radii of the scanned shape with the 20 sets of 18 feature radii of each candidate shape is necessary since the orientation of a 'bad ' shape as given by the principal axis is not reliably accurate enough for stitchmarking. The polar coding and correlation method as well as producing a matching class also gives the orientation of the scanned shape. However, the time taken to compare the radii of candidate shape is given in table 8.7 as 140 msecs. So to examine the radii of 180 candidate shapes would take the excessive time of 25.2 seconds.

9.3 Speeding up the recognition algorithm

The obvious methods of reducing the time to compare the radii by the polar coding and correlation method are to alter the algorithm so that it performs the same task but in less time, or to increase its execution speed. There is some scope for improving the algorithm. For example, all of the 360 possible sets of 18 radii of each candidate shape are compared with the set of the scanned shape, before the best fit is announced. This could be made more efficient by starting with the set of radii based on the inaccurate direction given by the principal axis. Then the sets of radii on each side could be examined until the set with a minimum mismatch is found. The comparisons would then stop, without having to examine the rest of the radius sets.

The execution speed of the algorithm could be further improved by rewriting it in a more efficient programming language. For example, pro-

grams written in the 'C' language generally produce less executable code than the Fortran used by the matching algorithms, and their passing of arrays by the use of pointers is quicker than the passing of whole arrays, as in Fortran. For further speed the programs could be rewritten in assembler language, at the expense of portability in the event of the need to run the program on another computer. Finally, a faster computer system could be used, i.e. one using the same processor, but with a faster system clock, or one that uses a different processor with a faster architecture.

9.4 Alternative methods for dealing with 'bad' shapes

Another way of approaching this problem, and with which the rest of this chapter is concerned, is to use an alternative, faster method of dealing with 'bad' shapes. The three basic methods of doing this that were tried were

i) to use a better method of determining the orientation of 'bad' shapes, then use the standard recognition method with the single set of 18 radii of candidate shapes, based on this orientation,

ii) to determine the orientation in two stages, roughly first for recognition, then accurately for printing, and

iii) to add another feature to the binary decision tree so that more shapes are filtered out leaving fewer candidate shapes to be matched by the polar coding and correlation method.

A check was made of the methods discussed in section 2.3.6 for specifying the orientation of a shape to see if the usefulness of any of these had been overlooked. Also, a further literature search was performed and various knowledgeable people were consulted.

Three alternative methods of determining the orientation of 'bad' shapes then looked worthwhile for further investigation. These were

i) use of the axis of maximum symmetry of the shape (section 9.4.1);

ii) use of the phase shift of the angle of Fourier harmonics of the outline of the shape (section 9.4.2), and

iii) use of higher order moments of area of the shape (section 9.4.3).

These methods were investigated, as detailed in the indicated sections, as possible better alternatives to the polar coding and correlation technique, but were found to have various deficiencies which rendered them unsuitable.

The two other methods round this problem were also investigated, though again with no overall advantage over the previous method. These were

iv) determining the orientation by a two–stage method (section 9.4.4), and

v) filtering out more candidate shapes using additional features in the binary decision tree (section 9.4.5).

9.4.1 Axis of maximum symmetry

The vast majority of the bad shapes are 'vamps', which are used in the forepart of the shoe upper. An example is illustrated in fig. 9.1, and it can be seen that they are nearly symmetrical about an axis along the centre of the foot. Since they have such a distinctive axis of symmetry, it should be possible to specify their orientation very accurately by calculating its direction. The method would hopefully be valuable in determining the orientation of other shapes too. The use of this method is noted by Kammenos [26], and a method similar to his was used.

The usefulness of this method was assessed by using the 20 sets of 18 radii that were stored in the database when a 'bad' shape was taught. By modifying the teaching program these 20 sets of radii were stored for 'good' shapes too. These sets of radii provided 360 radii at 1° intervals. The axis of maximum symmetry was determined by finding the minimum of $X(\alpha_j)$, where

$$X(\alpha_j) = \sum_{i=0}^{K-1} |R(\theta_i + \alpha_j) - R(\theta_i - \alpha_j)|$$

Effectively, a mirror image of the shape was generated by reversing the order of the stored radii, $R(\theta_i)$. Then the array of radii of the mirror image was shifted, by α_j, and compared with the radii of the original and the difference

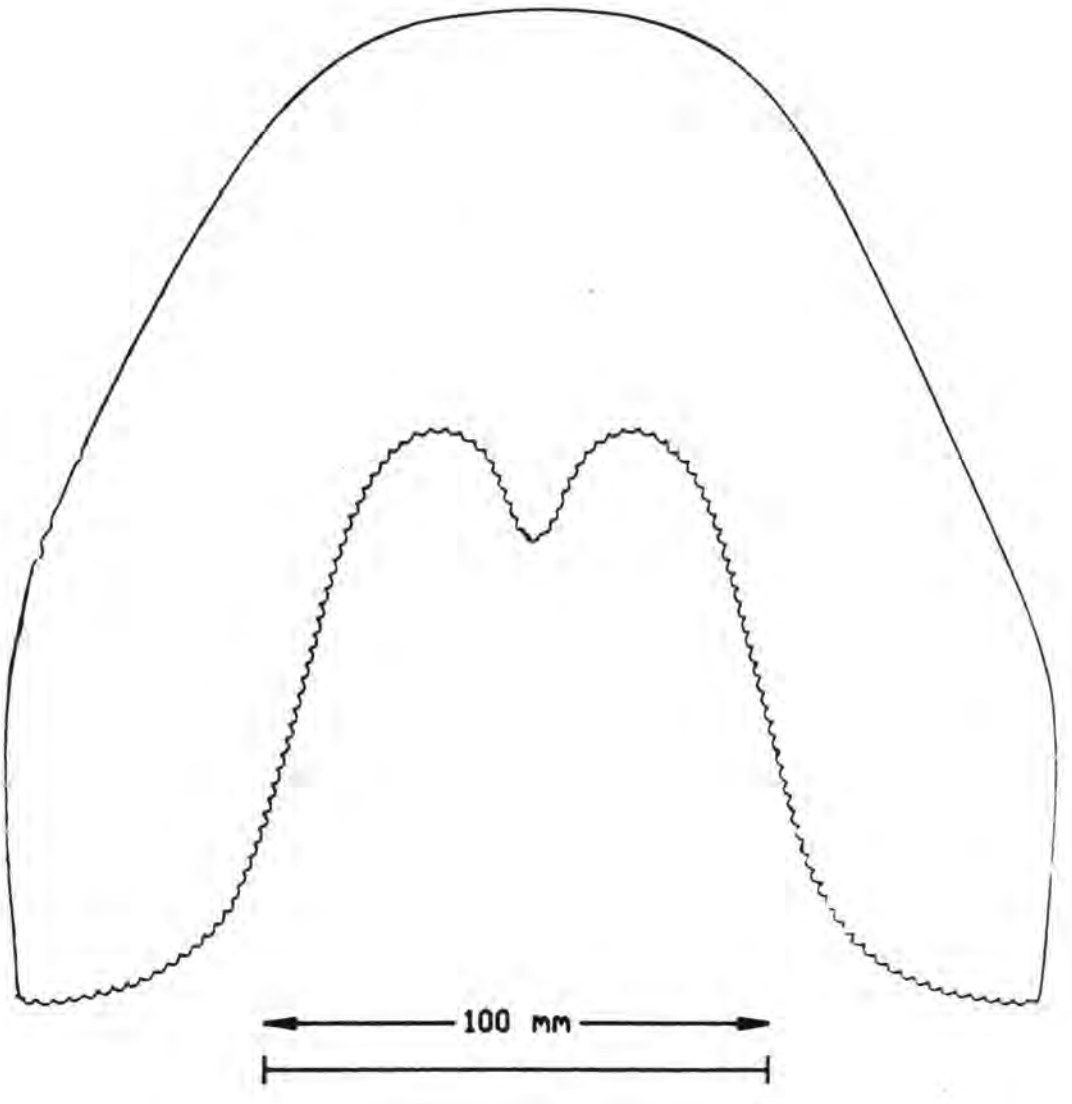


Figure 9.1 *Typical 'vamp', shape gg79.*

between the corresponding radii at each orientation was noted. This was repeated for all K, in this case 360, intervals. The sum of the differences at any rotation are at a minimum when the mirror image most closely matches the original image. The rotation of the mirror image required to do this was taken to be the orientation of the original shape. To get the orientation to finer than 1° an interpolation technique with the radii could be used.

This technique was tested on several shapes, both 'good' and 'bad', and the typical results are illustrated in the plots of radii difference against rotation angle shown in figs. 9.2 to 9.4. These show a very deep and sharp minimum for vamp shapes as expected, though less useful minima are displayed for most 'good' shapes, and very blunt minima for a shape such as ck74 which was approaching a circle.

It was decided that since the flatness of the minima for non–vamp shapes spanned several degrees this method might not be sufficiently accurate for them. It would be restricted to vamp type shapes with their sharp minima, with 'good' shapes continuing to use the principal axis and other 'bad' shapes using the polar coding and correlation method, as before.

However, the major shortcoming of this method was the time taken to calculate the radii required for the scanned shape. The time taken by the Slave Z8000 processor to calculate 360 radii using the assembler coded program was estimated at (Number of edge points $\times 0.24$) msecs.

For a large shape with 8000 edge points this gave a time of about 2.0 seconds. The time taken to calculate the angle of maximum symmetry using a program written in Fortran on the Compaq computer was also 2.0 seconds. This was reduced to 0.4 seconds by initially calculating the minimum of the radii differences by rotating the mirror image in 4° steps rather than 1° steps. The program then searched for a lower minimum in 1° steps on both sides of the first minimum. This gave an overall maximum time for this method of 2.4 seconds, which was still too long. An analysis of the Z8000 program for calculating the radii showed that it was already very efficient and that there was very little that could be done to speed it up apart from increasing the processor clock speed or replacing the software calculation with hardware.

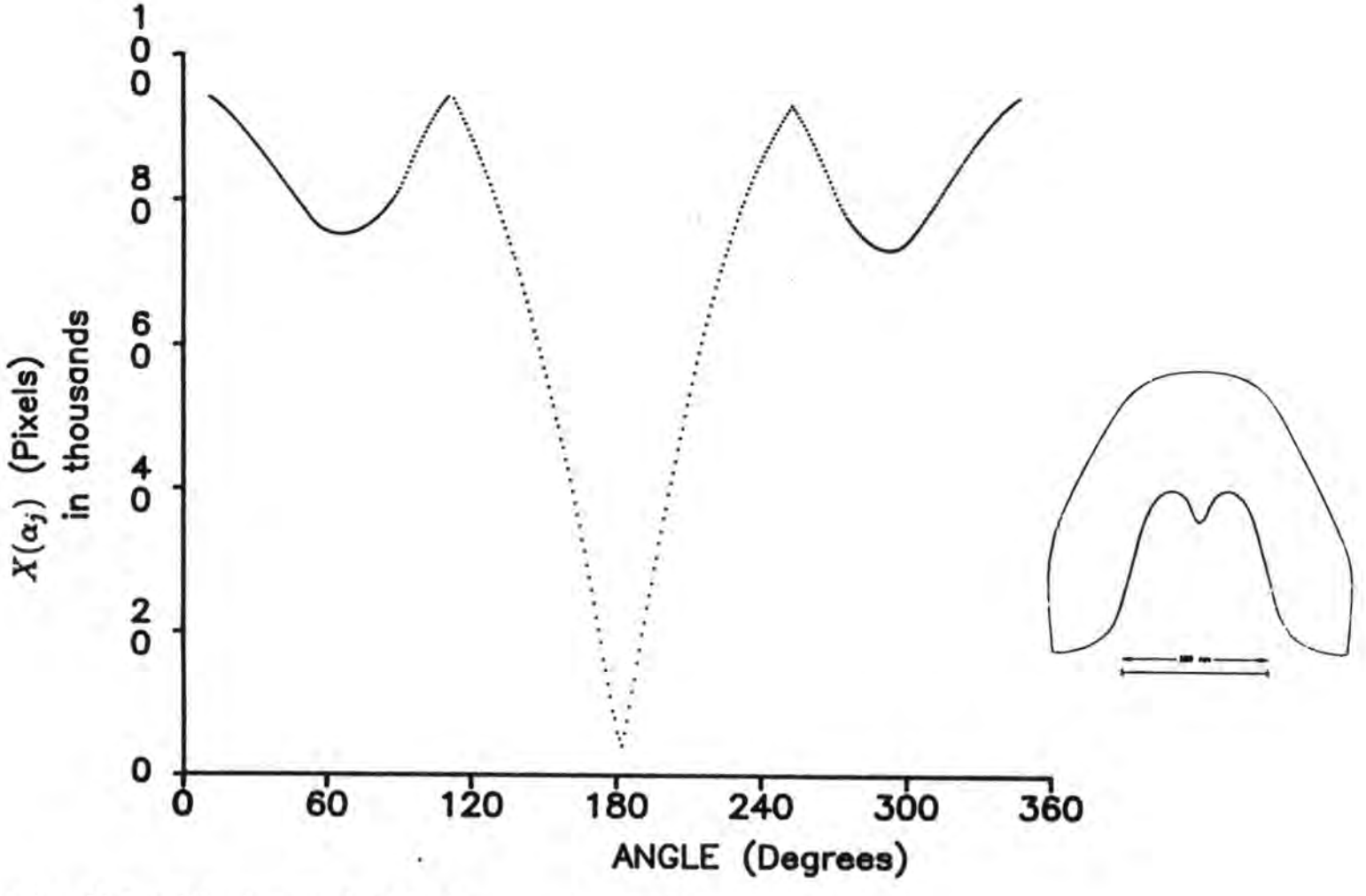


Figure 9.2 *Plot of $X(\alpha_j)$ against angle for vamp shape gg79.*

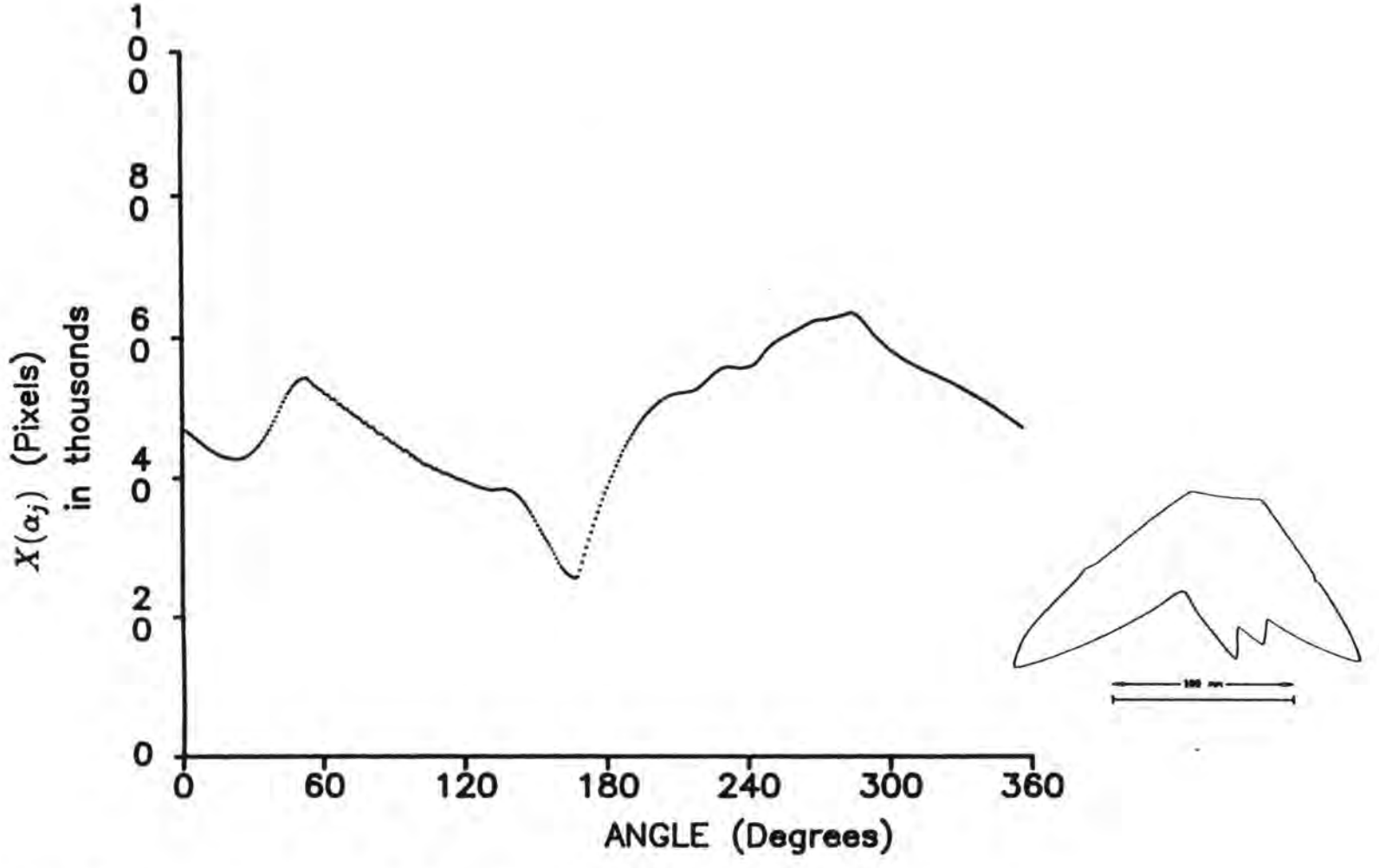


Figure 9.3 *Plot of $X(\alpha_j)$ against angle for 'good' shape dur1070.*

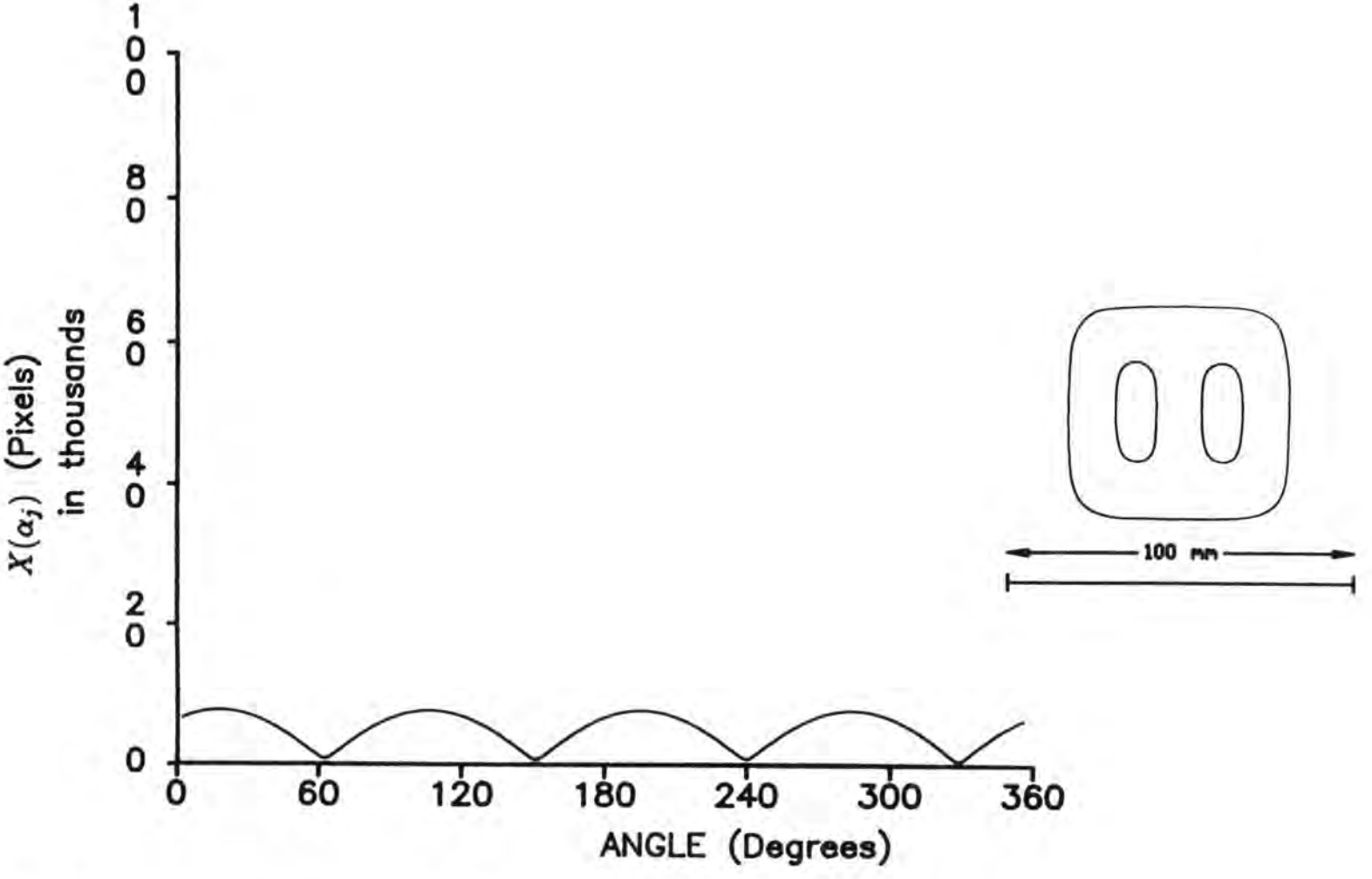


Figure 9.4 *Plot of $X(\alpha_j)$ against angle for shape ck74.*

So, although this method looked very promising for determining the orientation of vamp shapes, it took longer than could be allowed. Hence, work on it was stopped while the other methods were investigated.

9.4.2 Phase angle of Fourier harmonics

The basis of another method which was suggested was to analyse the outline of a shape by decomposing it into the Fourier harmonics from which it can be reproduced. The rotation between a scanned and a database shape could then be determined by calculating the phase difference of one or more of the corresponding harmonics of each. This, it was suggested, might allow the orientation to be determined using a reduced number of radii of the scanned shape than the symmetry technique. It would also be possible to use fast Fourier programs or integrated circuits to increase the speed of the calculations if the technique were feasible.

Fourier analysis has previously been used as a basis for describing complex shapes by encoding the curvature of the outline [16] [70] [71]. Since shoe components are fairly simple shapes a more appropriate method was used in this case which made use of the sets of radii stored in the database by the polar coding technique.

Fourier's theorem states [72] that a function, $x(t)$ may be expanded over the interval $[0, T]$ as a series of the form

$$x(t) = C_0 + C_1 \sin(\omega_0 t + \theta_1) + C_2 \sin(2\omega_0 t + \theta_2) + , ..., + C_n \sin(n\omega_0 t + \theta_n)$$

where $\omega_0 = \frac{2\pi}{T}$.

This may be considered as the sum of the fundamental frequency component, or first harmonic, $C_1 \sin(\omega_0 t + \theta_1)$, having a fundamental frequency of $f = \frac{\omega_0}{2\pi}$, together with its harmonics and a constant term C_0. The coefficients $C_1, ..., C_n$ represent the peak amplitude excursion of the fundamental and harmonic components of the series. The angles $\theta_1, ..., \theta_n$ represent the phase relationship between the initial vector value for the fundamental and those of the harmonics at time t.

A feature of such a series is that the value of the function exactly repeats itself at regular intervals. The interval required for one complete

repetition is known as the period T and the number of cycles per unit time is the fundamental frequency, where

$$f_0 = \frac{\omega_0}{2\pi} = \frac{1}{T}$$

The equation can be converted to the more convenient form

$$x(t) = a_0 + a_1 \cos \omega_0 t + b_1 \sin \omega_0 t + , ..., + a_n \cos n\omega_0 t + b_n \sin n\omega_0 t$$

which can be written as

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos k\omega_0 t + \sum_{k=1}^{n} b_k \sin k\omega_0 t$$

This is known as the Fourier series of the function $x(t)$, and the terms a_k and b_k are known as the Fourier coefficients.

The Fourier coefficients can be determined from

$$\frac{a_0}{2} = \frac{1}{T} \int_0^T x(t).dt$$

$$a_k = \frac{2}{T} \int_0^T x(t) \cos k\omega_0 t.dt$$

$$b_k = \frac{2}{T} \int_0^T x(t) \sin k\omega_0 t.dt$$

The coefficients a_k and b_k represent the same frequency of the Fourier spectrum, and can be combined into a single vector sum to represent the modulus (C_k) and phase (θ_k) of each of the harmonic amplitudes

$$C_k = \sqrt{(a_k^2 + b_k^2)}$$

and

$$\theta_k = arctan \left[\frac{b_k}{a_k} \right]$$

The Fourier coefficients of a sampled periodic waveform can be evaluated using the following well known approximate method. Consider the function $f(x)$ $(0 < x < 2\pi)$ which is specified by a set of points on a graph. Let the range $0 \leq x \leq 2\pi$ be divided into p equal subdivisions, each of width $h = \frac{2\pi}{p}$. Then

$$a_k = \frac{2}{p} \sum_{k=0}^{p-1} f(x_k) \cos nx_k$$

and

$$b_k = \frac{2}{p} \sum_{k=0}^{p-1} f(x_k) \sin nx_k$$

If the silhouette of the shape is polar coded at equal angles then the lengths of the resultant radii can be plotted against angle, as shown for the large 'bad' shape gg23 in figs. 9.5 & 9.6.

The method was evaluated by inserting the 360 radii obtained from shape gg23 into a spreadsheet program which calculated the Fourier coefficients. Since the calculation of 360 radii of the scanned shape had already been ruled out as taking too long, section 9.4.1, the method was repeated using less radii. Table 9.1 lists the amplitudes and phases of the first two harmonics calculated from different numbers of radii. It shows that their values alter as the number of radii used in the calculations is increased. This illustrates that the frequency of sampling affects the accuracy of the result.

Number of radii used	Sampling frequency	1st. harmonic Amplitude	1st. harmonic Phase	2nd. harmonic Amplitude	2nd. harmonic Phase
45	8°	25.78	-1.47	28.92	0.01
90	4°	19.50	-1.43	30.15	0.02
180	2°	17.36	-1.37	30.72	0.05
360	1°	17.46	-1.29	30.72	0.11

Table 9.1 *Variation of the calculated harmonic amplitude and phase angle with the sampling frequency.*

Table 9.2 shows the Fourier coefficients of the first 20 harmonics calculated using 90 radii, i.e. sampling at 4° intervals, with the shape unrotated and rotated through 1° and 4°. From these results the phase differences of the harmonics were calculated, with the results shown in Table 9.3. Note that the phase, as given by $arctan\left[\frac{b_k}{a_k}\right]$, is indeterminate by π radians, i.e. 180°.

Table 9.3 shows that the calculated phase angle did not give the correct rotation when the rotation angle was less than the sampling frequency,

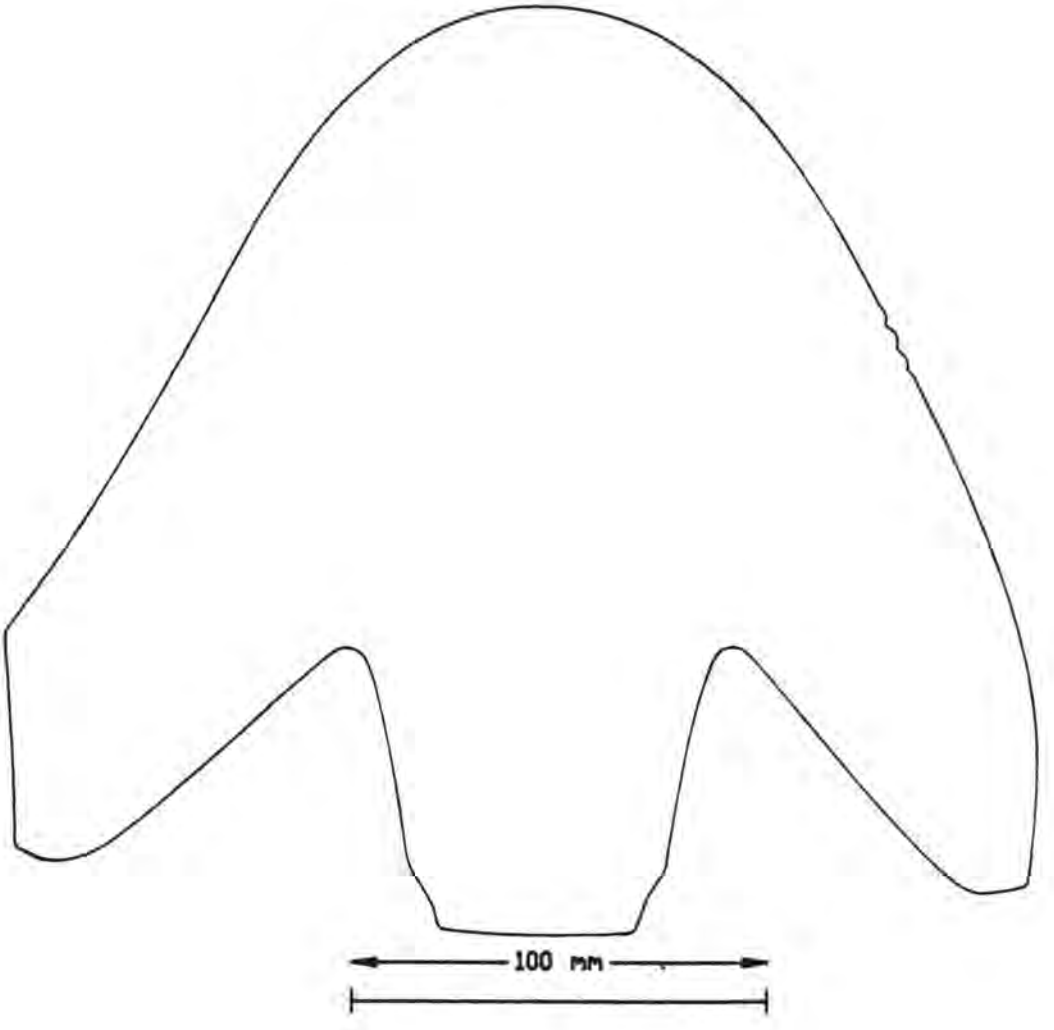


Figure 9.5 *Component gg23, used in the investigations of the determination of orientation using Fourier harmonics.*

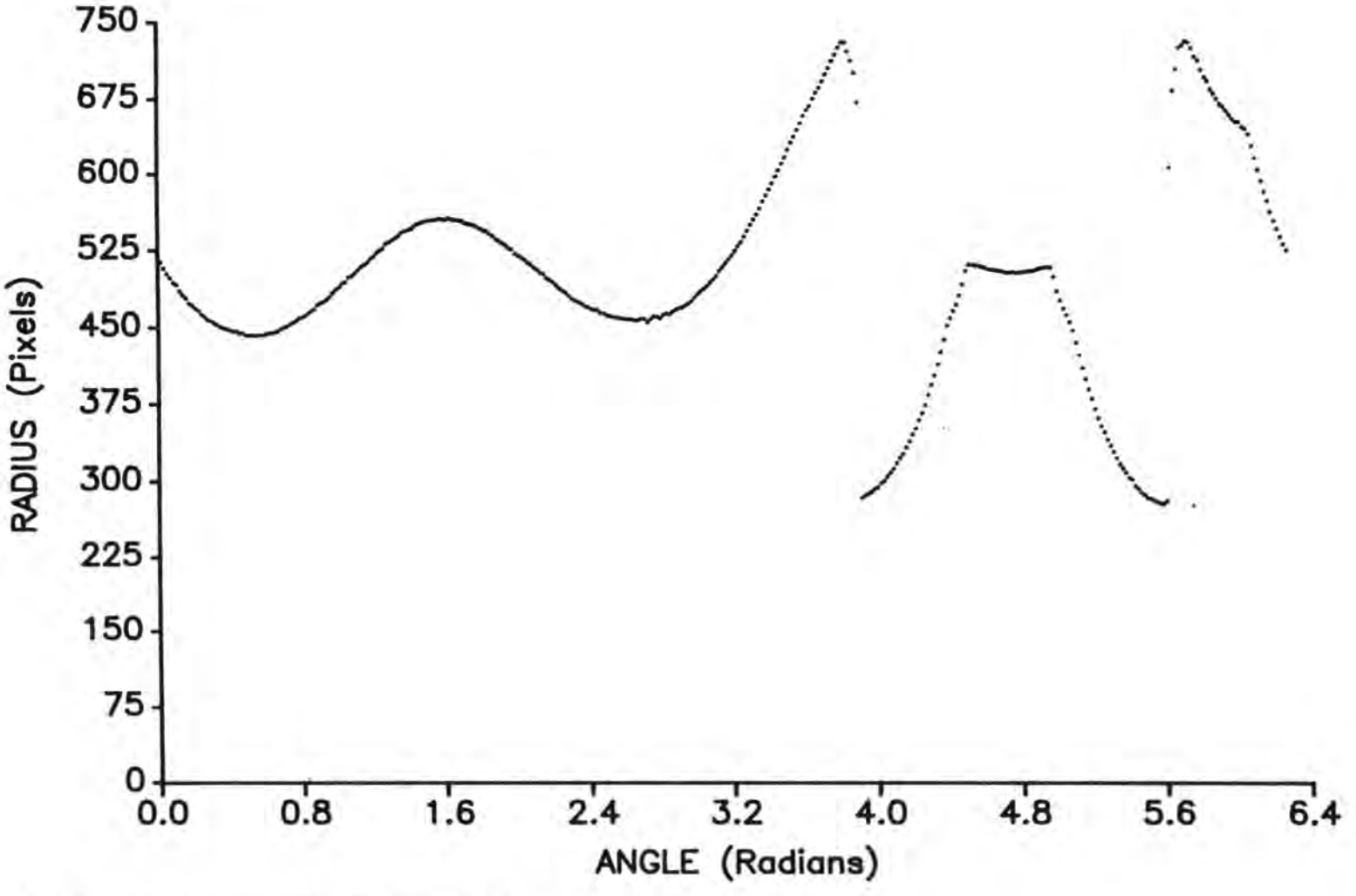


Figure 9.6 *The polar coding radii of shape gg23.*

Harmonic	Unrotated shape				1 degree rotation of shape				4 degree rotation of shape			
(k)	a_k	b_k	C_k	θ_k	a_k	b_k	C_k	θ_k	a_k	b_k	C_k	θ_k
1	-2.68	19.31	19.50	-1.43	-5.97	19.33	20.23	-1.27	-1.33	19.45	19.50	-1.50
2	30.15	0.61	30.15	0.02	29.93	3.94	30.19	0.13	29.94	-3.59	30.15	-0.12
3	10.14	-58.00	58.88	-1.40	9.27	-58.49	59.22	-1.41	-2.14	-58.84	58.88	1.53
4	44.68	-1.20	44.69	-0.03	44.25	-2.71	44.33	-0.06	42.62	-13.47	44.69	-0.31
5	11.99	-73.83	74.79	-1.41	9.48	-73.83	74.44	-1.44	-13.99	-73.47	74.79	1.38
6	-55.10	-14.35	56.94	0.25	-55.93	-12.21	57.25	0.21	-56.17	9.30	56.97	-0.16
7	-3.72	8.35	9.14	-1.15	-3.15	9.01	9.55	-1.23	0.63	9.11	9.14	1.50
8	-26.40	-8.89	27.86	0.32	-27.07	-7.83	28.18	0.28	-27.10	6.45	27.86	-0.23
9	-12.82	35.22	37.48	-1.22	-11.54	35.93	37.73	-1.26	10.32	36.03	37.48	1.29
10	21.42	6.85	22.49	0.31	21.79	6.01	22.60	0.27	20.81	-8.52	22.49	-0.39
11	-2.51	0.24	2.52	-0.09	-2.72	-0.07	2.72	0.03	-1.64	1.91	2.52	-0.86
12	15.95	7.99	17.84	0.46	15.96	8.28	17.98	0.48	16.61	-6.50	17.84	-0.37
13	8.39	-17.07	19.02	-1.11	8.19	-17.43	19.26	-1.13	-8.28	-17.12	19.02	1.12
14	-5.23	-3.21	6.14	0.55	-5.44	-3.52	6.48	0.57	-5.59	2.54	6.14	-0.43
15	5.03	-10.49	11.63	-1.12	3.93	-10.58	11.29	-1.22	-6.57	-9.60	11.63	0.97
16	-18.06	-10.37	20.82	0.52	-18.38	-9.09	20.51	0.46	-17.24	11.68	20.82	-0.60
17	-8.66	12.26	15.01	-0.96	-8.08	12.99	15.30	-1.01	8.12	12.62	15.01	1.00
18	-0.59	0.58	0.83	-0.77	-0.18	0.71	0.74	-1.32	0.36	0.74	0.83	1.12
19	-8.60	10.80	13.81	-0.90	-7.77	11.24	13.66	-0.97	8.40	10.96	13.81	0.92
20	11.80	10.93	16.08	0.75	12.43	10.41	16.21	0.70	12.81	-9.73	16.08	-0.95

Key :

a_k & b_k are the Fourier coefficients of the harmonic k.

C_k is the modulus of the harmonic k.

θ_k is the phase angle of the harmonic k.

Table 9.2 *Fourier coefficients, magnitudes, and phase angles of the harmonics of the polar coded outline of shape gg23, when unrotated and rotated through* 1° *and* 4°.

Harmonic (k)	Unrotated shape θ_k	1 degree rotation of shape θ_k	$\Delta\theta_k$ (rads)	$\Delta\theta_k$ (deg)	4 degrees rotation of shape θ_k	$\Delta\theta_k$ (rads)	$\Delta\theta_k$ (deg)
1	-1.4329	-1.2712	+0.1617	+9.26	-1.5027	-0.0698	-4.00
2	+0.0203	+0.1308	+0.1105	+6.33	-0.1193	-0.1396	-8.00
3	-1.3977	-1.4136	-0.0159	-0.91	+1.5345	+2.9322	+168.00
4	-0.0268	-0.0611	-0.0343	-1.97	-0.3061	-0.2793	-16.00
5	-1.4099	-1.4431	-0.0332	-1.90	+1.3827	+2.7926	+160.00
6	+0.2548	+0.2149	-0.0399	-2.29	-0.1641	-0.4189	-24.00
7	-1.1516	-1.2345	-0.0829	-4.75	+1.5013	+2.6523	+151.97
8	+0.3247	+0.2815	-0.0432	-2.48	-0.2338	-0.5585	-32.00
9	-1.2216	-1.2599	-0.0383	-2.19	+1.2917	+2.5133	+144.00
10	+0.3097	+0.2693	-0.0404	-2.31	-0.3885	-0.6982	-40.00
11	-0.0944	-0.0266	+0.0678	+3.88	-0.8624	-0.7680	-44.00
12	+0.4647	+0.4788	+0.0141	+0.81	-0.3731	-0.8378	-48.00
13	-1.1137	-1.1315	-0.0178	-1.02	+1.1203	+2.2340	+128.00
14	+0.5504	+0.5745	+0.0241	+1.38	-0.4270	-0.9774	-56.00
15	-1.1234	-1.2150	-0.0916	-5.25	+0.9710	+2.0944	+120.00
16	+0.5213	+0.4593	-0.0620	-3.55	-0.5957	-1.1170	-64.00
17	-0.9555	-1.0146	-0.0591	-3.39	+0.9993	+1.9548	+112.00
18	-0.7685	-1.3193	-0.5508	-31.56	+1.1164	+1.8849	+108.00
19	-0.8985	-0.9658	-0.0673	-3.86	+0.9167	+1.8152	-104.00
20	+0.7468	+0.6970	-0.0498	-2.85	-0.6495	-1.3963	-80.00

Key :

θ_k is the phase angle of the harmonic k.

$\Delta\theta_k$ is the difference in phase angle of the harmonic k compared to that of the unrotated shape.

Table 9.3 *Phase differences of the Fourier harmonics, calculated from a* 4° *sampling interval, of the coded outline of shape gg23 when rotated through* 1° *and* 4°.

as can be seen by examining the $\Delta\theta_k$ (degrees) columns. Note that the calculated rotation for the higher harmonics is proportional to their harmonic number and is sometimes shifted by +180°.

Table 9.4 lists the phase differences of the first two harmonics, calculated from the Fourier coefficients, for various angles of rotation of the shape. It emphasises that these phase differences are only correct when the rotation angle is greater than the angle of the sampling interval. By examining the phase difference (°) columns it can be seen that accurate phase differences were only obtained for rotations of the shape which were multiples of the 4° sampling interval.

Rotation	1st. harmonic				2nd. harmonic			
(°)	Amplitude	Phase	Diff. (rads)	Diff. (°)	Amplitude	Phase	Diff. (rads)	Diff. (°)
0	19.50	-1.43	-	-	30.15	+0.02	-	-
1	20.23	-1.27	+0.16	9.19	30.19	+0.13	+0.11	+6.30
2	15.31	-1.32	+0.11	6.30	31.22	+0.01	-0.01	-0.57
3	15.11	-1.22	+1.21	69.33	31.45	+0.05	+0.03	+1.72
4	19.50	-1.50	-0.07	-4.01	30.15	-0.12	-0.14	-8.02
5	20.23	-1.34	+0.09	+5.16	30.27	-0.01	-0.03	-1.72
8	19.50	+1.57	+3.00	+171.89	30.23	-0.26	-0.28	-16.04
60	19.50	+0.66	+2.09	+119.7	30.12	+1.07	+1.05	60.16

Table 9.4 *Variation of calculated rotation angle with actual rotation angle for a sampling frequency of* 4°.

These results confirm that the orientation cannot be determined by this technique to a smaller interval than the sampling frequency. Since the orientation is required to an accuracy of better than 1° and the time to calculate 360 radii of the scanned shapes takes too long, then this method was also discarded.

As a check on the method the shape gg23 was reconstructed from the calculated harmonic parameters sampled at 4° intervals, with the resulting curves, which closely resembled the original coded shape, shown in appendix C.

9.4.3 High order moments of area

The use of second moments of area to determine the orientation of a silhouette shape is quite common, as noted in section 2.3.6. It was decided to investigate the use of higher order moments for defining the orientation of a shape and see whether they have any advantages over the 2nd. order moments. This method promised to be fast since it could be implemented as a series of summations in a similar manner to the calculations of the 2nd. order moments.

As explained in chapter 2, the moments of area of a two dimensional shape are defined by

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx.dy$$

For shapes binarily digitised by a square grid, the moments can be approximated by summations of the coordinates of the pixels over the area quantised by $m \times n$ pixels, where the function $f(x,y)$ takes the value 1 within the silhouette of the shape and 0 outside it. The mathematics is simpler if the origin of the coordinate system passes through the centroid of the shape.

The second order moments are used to find the principal axis of the silhouette by the following method [67]. The three second order moments are given by

$$M_{20} = \sum_{1}^{m} \sum_{1}^{n} x^2 f(x,y)$$

$$M_{02} = \sum_{1}^{m} \sum_{1}^{n} y^2 f(x,y)$$

$$M_{11} = \sum_{1}^{m} \sum_{1}^{n} x.y f(x,y)$$

The way in which the 2nd. moments vary as the coordinate system is rotated by an angle θ about the shape's centroid can be determined by considering how the coordinates of points will alter. A pixel with coordinates (x,y) in the xy system will now have coordinates (x',y') in the $x'y'$ system. Coordinates in the two systems are related by

$$x' = x\cos\theta + y\sin\theta$$

$$y' = y\cos\theta - x\sin\theta$$

The 2nd. order moments are now given by

$$\begin{aligned} M'_{20} &= \sum_1^m \sum_1^n (x\cos\theta + y\sin\theta)^2 f(x,y) \\ &= M_{20}\cos^2\theta - 2M_{11}\cos\theta\sin\theta + M_{02}\sin^2\theta \end{aligned}$$

$$\begin{aligned} M'_{02} &= \sum_1^m \sum_1^n (y\cos\theta - x\sin\theta)^2 f(x,y) \\ &= M_{02}\cos^2\theta + 2M_{11}\cos\theta\sin\theta + M_{02}\sin^2\theta \end{aligned}$$

$$\begin{aligned} M'_{11} &= \sum_1^m \sum_1^n (x\cos\theta + y\sin\theta).(y\cos\theta - x\sin\theta) f(x,y) \\ &= M_{20}\sin\theta\cos\theta + M_{11}(\cos^2\theta - \sin^2\theta) - M_{02}\cos\theta\sin\theta \\ &= \frac{1}{2}(M_{20} - M_{02})\sin(2\theta) + M_{11}\cos(2\theta) \end{aligned}$$

The principal axes are defined as those for which $M_{11} = 0$. Rearranging the equation for M'_{11} gives

$$\tan(2\theta) = \frac{2M_{11}}{M_{02} - M_{20}} \tag{9.1}$$

This equation gives two values of θ differing by 90°, one for the major and one for the minor principal axes. The angle of the major principal axis is the one which has been used to specify the orientation of a shape. This relationship means that whatever the scanned angle of the shape its orientation can be simply determined from its 2nd. moments at that angle.

The values of the 2nd. moments at 5° intervals were calculated from the edge point coordinates of scanned shapes. Those for the 'good' shape gg151 are illustrated graphically in fig. 9.7 and those for the 'bad' shape shoe3–1 (which has the very low GCOEFF of 0.068) are illustrated in fig. 9.8. These show that M_{20} and M_{02} are at maximum and minimum when M_{11} is zero. Note that for the 'bad' shape, fig. 9.8, M_{20} and M_{02} are nearly equal, which is the source of the unreliability of the orientation given by them.

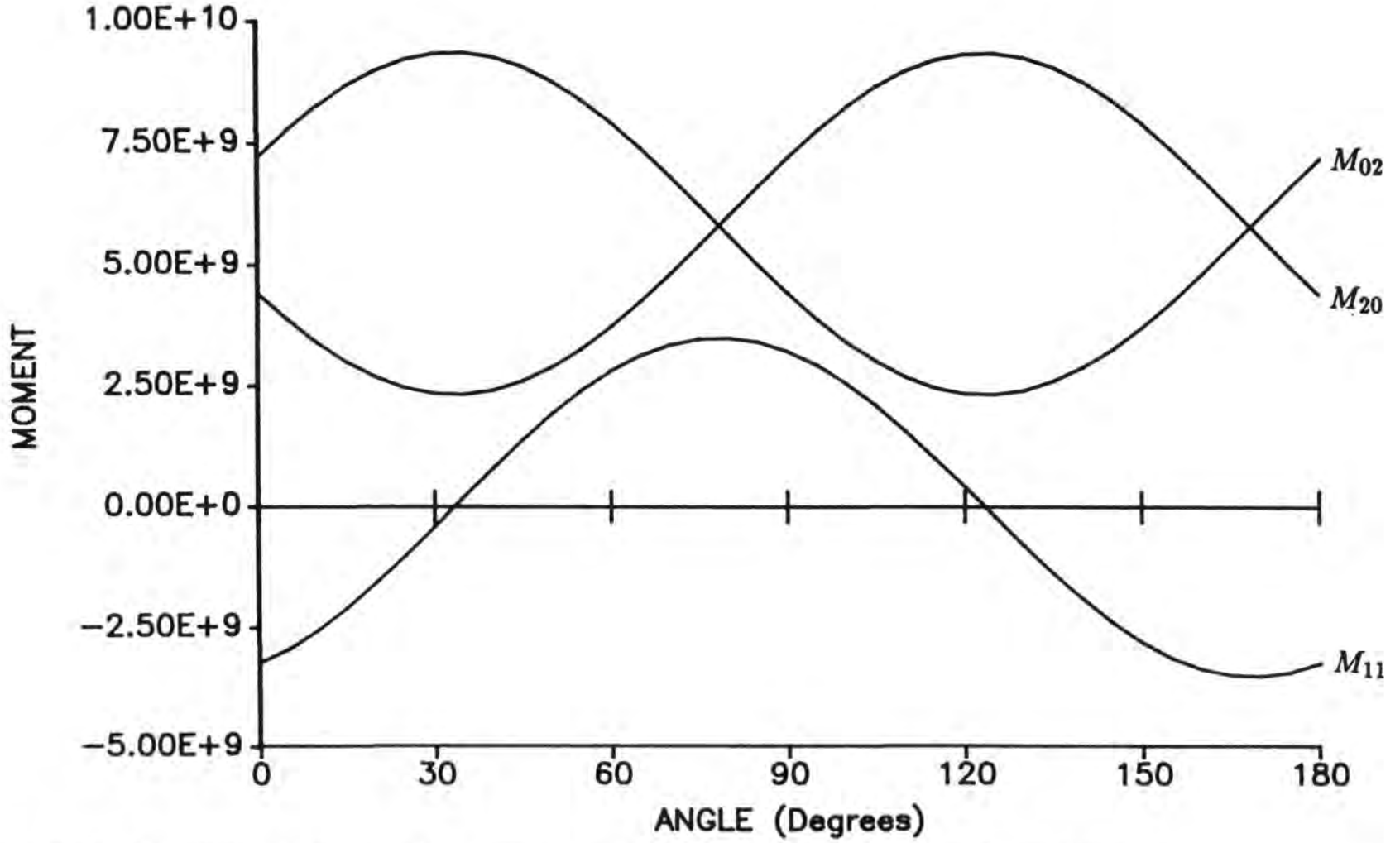


Figure 9.7 *The variation of the second moments of area of the 'good' shape gg151 with orientation.*

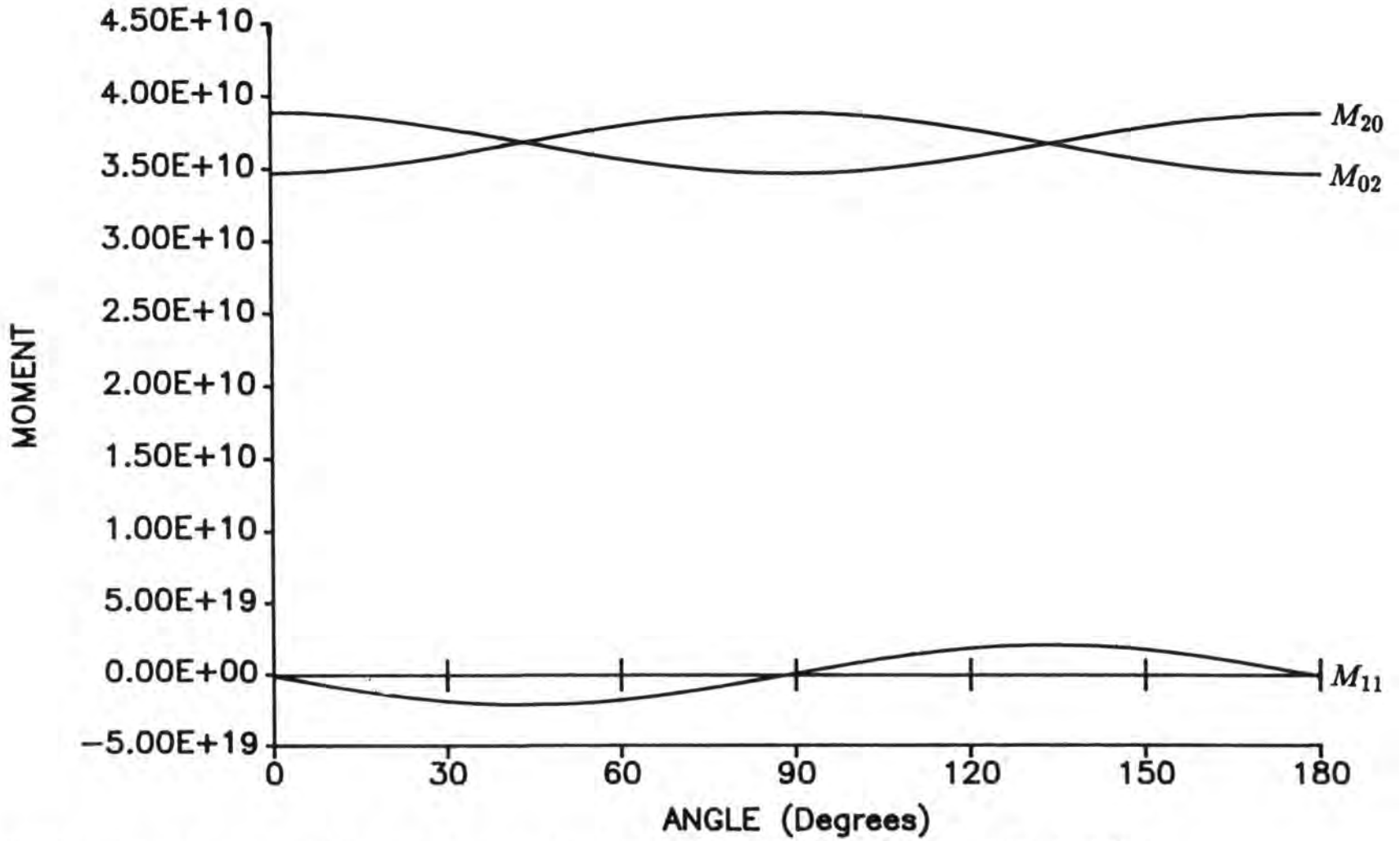


Figure 9.8 *The variation of the second moments of area of the 'bad' shape shoe3_1 with orientation.*

It had been suggested that higher moments may be more useful for determining the orientation of 'bad' shapes. This would be even more useful if there existed a simple relationship to determine the angle as equation 9.1 for the 2nd. moments. The four third order moments are given by

$$M_{30} = \sum_{1}^{m}\sum_{1}^{n} x^3 f(x,y)$$

$$M_{21} = \sum_{1}^{m}\sum_{1}^{n} x^2.y f(x,y)$$

$$M_{12} = \sum_{1}^{m}\sum_{1}^{n} x.y^2 f(x,y)$$

$$M_{03} = \sum_{1}^{m}\sum_{1}^{n} y^3 f(x,y)$$

Values of the third moments of 'good' shape gg151 and 'bad' shape shoe3–1 at 5° intervals were calculated and were plotted for comparison purposes with those of the 2nd. moments as shown in figs. 9.9 & 9.10. However, no relationship comparable to equation 9.1 for the 2nd. order moments could be discerned.

The equations for the 3rd. moments when the coordinate system is rotated are given in appendix D. Although these were manipulated to find a simple solution of θ in terms of the third moments, none was found. It was hence decided to examine the fourth order moments instead.

The five fourth order moments are given by

$$M_{40} = \sum_{1}^{m}\sum_{1}^{n} x^4 f(x,y)$$

$$M_{31} = \sum_{1}^{m}\sum_{1}^{n} x^3.y f(x,y)$$

$$M_{22} = \sum_{1}^{m}\sum_{1}^{n} x^2.y2 f(x,y)$$

$$M_{13} = \sum_{1}^{m}\sum_{1}^{n} x.y^3 f(x,y)$$

$$M_{04} = \sum_{1}^{m}\sum_{1}^{n} y^4 f(x,y)$$

As before, graphs of these against angle of orientation for shapes gg151 and shoe3–1 were drawn, to see if any relationship similar to that for the 2nd.

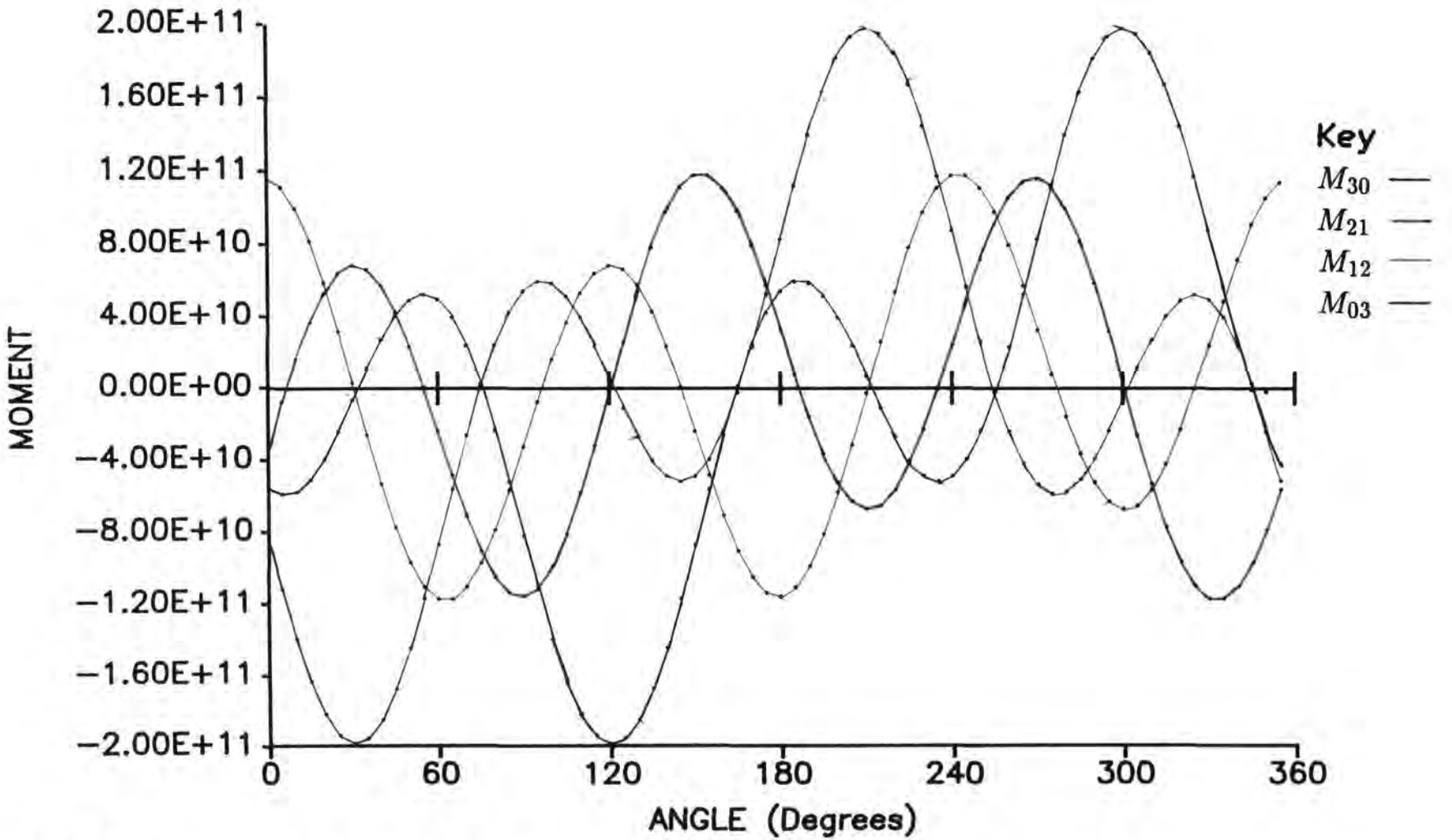


Figure 9.9 *The variation of the third moments of area of the 'good' shape gg151 with orientation.*

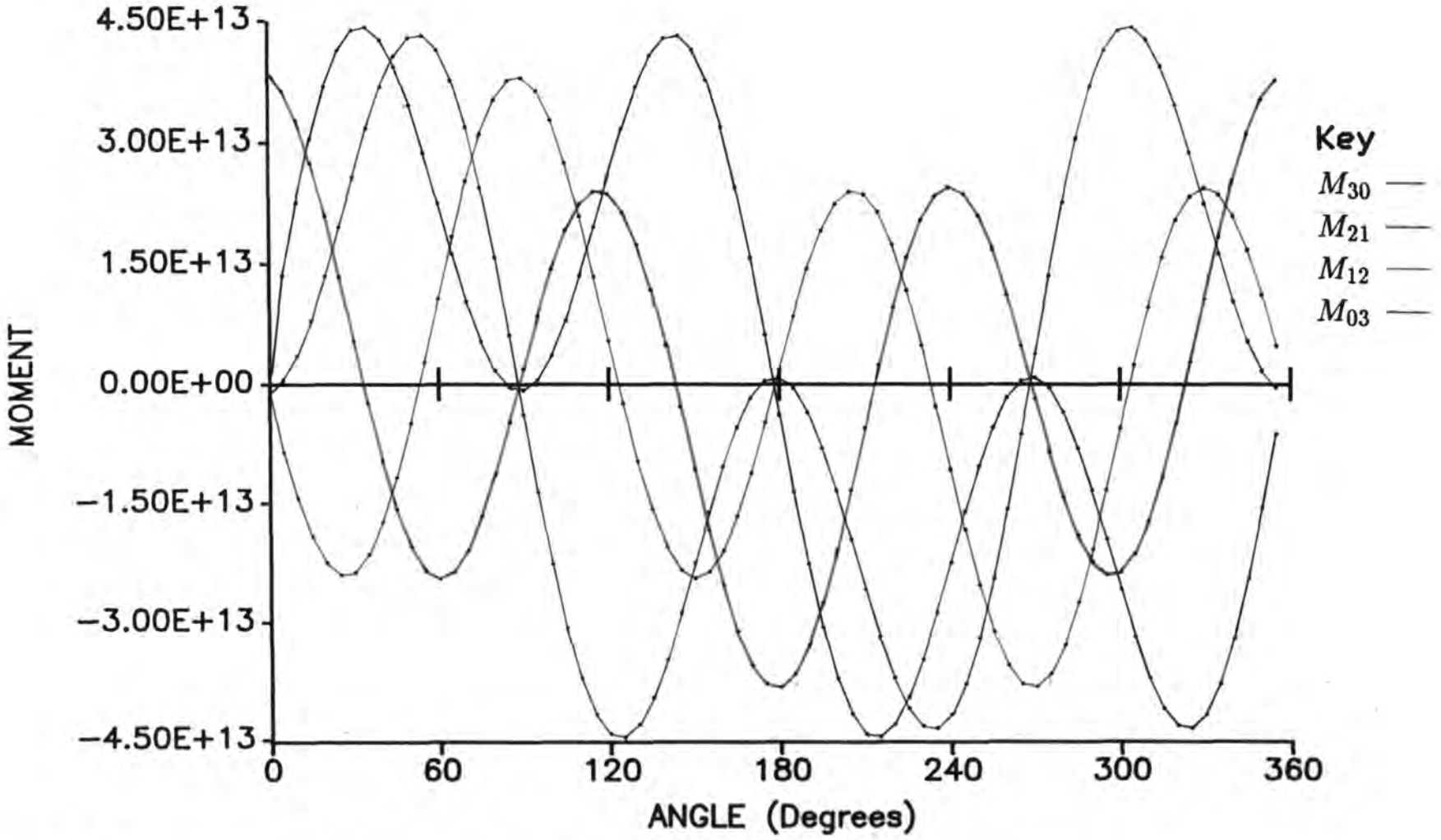


Figure 9.10 *The variation of the third moments of area of the 'bad' shape shoe3_1 with orientation.*

moments could be discerned, and are shown in figs. 9.11 & 9.12. Also the equations for the 4th. order moments when the coordinate system is rotated, as given in appendix D, were manipulated, unsuccessfully, to find a simple relationship between θ and the fourth moments.

No equation comparable to equation 9.1, for the 2nd. moments, was apparent from the complex equations for the variation of the higher moments with rotation. In the absence of such equations it would be necessary to use other less satisfactory methods for finding a maximum, minimum, or zero crossing point of one of the higher moments to specify the orientation. It was also considered that the reason that the 2nd. moments do not give the orientation of the shape reliably when the two principal moments are nearly equal is that the difference between them, as required by equation 9.1, is markedly affected by edge noise, caused both by the rough edge and also by quantisation noise. Since this noise would also be impressed on the coordinates used to calculate the higher moments, these would also exhibit unreliability in the specification of the orientation, and would require the use of another method at low values of GCOEFF. The use of higher moments was not completely ruled out, but it was decided that other methods ought to be investigated with a higher priority.

9.4.4 Two stage method

In order to stitchmark accurately the orientation of the component has to be determined precisely. However, for recognition purposes it is only necessary to know the orientation of the scanned shape sufficiently accurately to determine its set of feature radii. For 'bad' shapes the orientation given by the principal axis is not accurate enough for printing stitchmarking, though it may be accurate enough as the datum for the feature radii. The slower polar coding and correlation method would then only be used on the recognised shape class to get the accurate orientation for stitchmarking. Thus, in this 'two stage' technique the orientation would be determined roughly initially for recognition, and then accurately for printing stitchmarking.

Experiments were performed to test this hypothesis by determining the performance of the standard recognition technique for 'good' shapes,

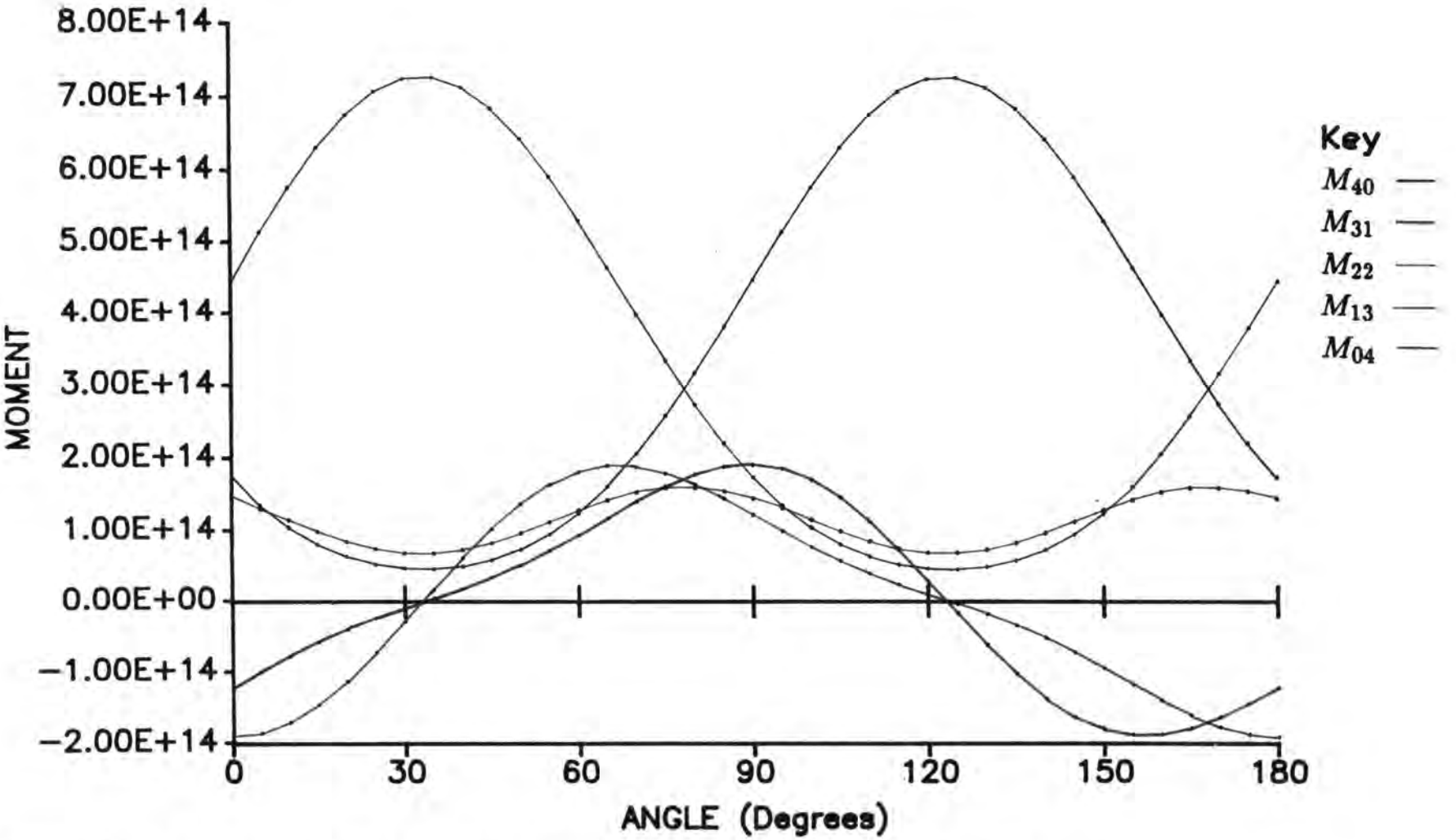


Figure 9.11 *The variation of the fourth moments of area of the 'good' shape gg151 with orientation.*

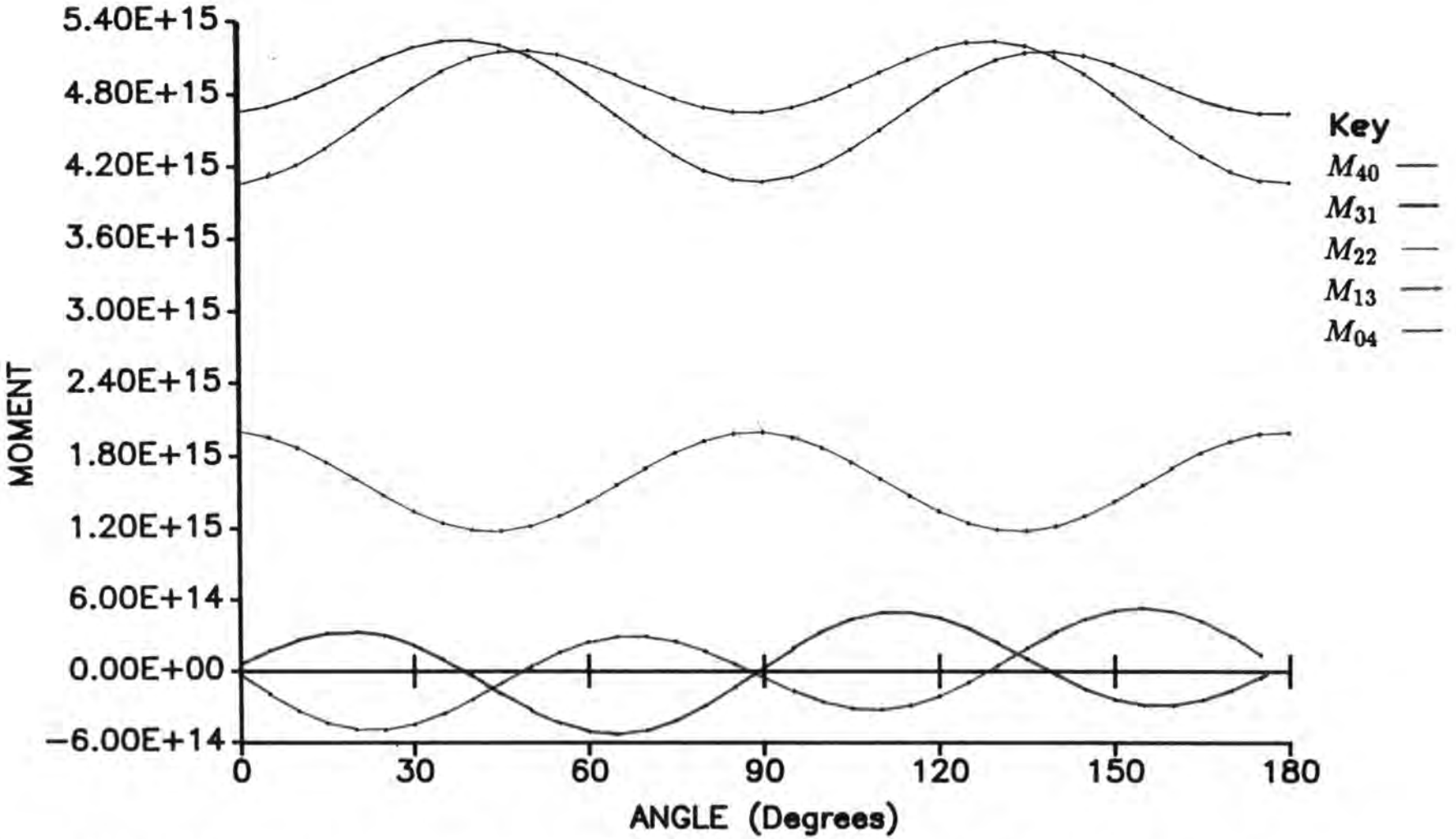


Figure 9.12 *The variation of the fourth moments of area of the 'bad' shape shoe3_1 with orientation.*

using the principal axis, when it was used on 'bad' shapes. A total of 118 'bad' shaped components were taught to the machine. Each was then fed through for recognition, twice using the polar coding and correlation technique and twice using the principal axis technique.

The usual polar coding and correlation method performed 2 substitutions, though this was between two pairs of shapes which were from a set of different width fittings of 3 half sizes of one design of component and were very similar. The proposed principal axis method made 9 substitutions, but only among this same group of shapes. It also gave one rejection.

Detailed results are listed in appendix E, which also shows how the number of mismatched radii when using the principal axis technique was reduced once the shape had been recognised and the polar coding and correlation technique had been used on it. The table also lists the difference in orientation as given by the two methods, and how this difference reduces as GCOEFF increases, i.e. the difference between the principal 2nd. moments of area becomes greater.

Although this two stage method performs reasonably well with this small database, the large number of mismatching radii at the first stage gave cause for concern. The performance with a large database was likely to be poor, and would have outweighed the advantages of any speed increase.

9.4.5 Additional features in the decision tree

The number of candidate shapes which have to be examined using the polar coding and correlation technique could be reduced by using further features in the binary decision tree. Many features have been suggested and discarded as not suitable, as outlined in section 2.3.2. It was decided that the features which warranted further consideration were the Standard deviation and the Mean deviation of the distances of all the edge points from the centroid. Either calculation could be performed simultaneously with the calculations of the second moments and the radii and would not incur too large a time penalty.

Several shapes were scanned and their standard and mean deviations

were evaluated as shown in table 9.5.

Shape	Standard deviation	Mean deviation
gg151	70.57	64.48
gg151 rotated 45 degrees	71.46	65.25
gg43	99.59	77.49
gg43 rotated 45 degrees	97.28	74.71
gg23	109.83	84.73
shoe3_1	140.82	113.88
Calibration disc	0.42	0.33

Table 9.5 *Standard and Mean deviations of the edge points of some typical shoe components*

Shape gg151 is a 'good' shape and gg23 & gg43 are 'bad' vamp shapes of the same design but large and small, respectively. Shoe3_1 is also a 'bad' vamp shape whereas the calibration disc is circular. Both the standard deviation and the mean deviation are promising as features for shape discrimination since different shapes have different deviations, and they are largely invariant to rotation.

However, analysis of the shapes which were close within the feature space, as described in chapter 7, and were likely to be candidate shapes for matching revealed that they consisted either of shapes of the same design but scaled by a half size, or shapes that were very similar anyway. Since these deviations did not give values which could reliably discriminate between such shapes they would not have helped much to filter out unwanted shapes. However, one use of the deviation would be when adjusting the camera to obtain a square and orthogonal quantisation grid using the calibration disc [4]. The camera could be adjusted to give the minimum deviation.

9.5 Conclusions

The conclusions from these experiments was that the polar coding and correlation method was the best method currently available for recognising and determining the orientation of 'bad ' shapes, since the other methods of calculating the orientation that were investigated took longer. No suitable, quickly calculated feature that would be efficient for all shoe shapes was found that could be used to filter out shapes and so reduce the number of candidates. Hence, the most satisfactory way of reducing the bottleneck caused by the projected large number of candidate 'bad' shapes appears to be to make the current algorithm more efficient, to rewrite the program in a faster executing code, and to use a faster processor.

CHAPTER 10

CONCLUSION

10.1 Introduction

This work has successfully demonstrated the feasibility of electrographic printing linked to a vision system for high speed automatic stitchmarking of shoe upper components. Before summarising the position attained with the stitchmarking system there are some aspects of its operation, especially with regard to the printer, which require recording.

10.2 Printer operation

The printer performed very well, standing up to heavy use and some abuse. However, one problem which was first really apparent on the exhibition machine was the high rate at which the toner was carried over from the developing station to the cleaning blade, even when the machine was just idling and not printing an image. This had not been so obvious with the experimental machine since the original printer drive motor had been retained to drive the developer unit. The motor was controlled by a signal from the master Z8000 microprocessor, and only operated when a shoe component was being printed. Due to a lack of space on the exhibition machine, the developer unit was driven by a small stepper motor which took its drive pulses from the conveyor belt driver. This resulted in developing action whenever the conveyors were operating.

A further contributory cause on both machines was that the erase rod was only energised while a component was actually being printed. It was found that the lack of erasing and the continuation of developing when the machine was running with the printer idling caused the excessive carry over of toner on the drum from the developer unit to the cleaning blade. This toner is retained by a catcher tray, which soon became full. Also the cleaning blade is not 100 % efficient, which, with the high rate of toner carry over, resulted in a build up of toner in small areas on the erase rod, which is only spaced 0.25 mm from the drum. Although the erase rod is heated to below the melting point of the toner, there is a high frequency high voltage applied to the windings and the toner is conductive. This resulted in the toner fusing to the erase rod in those areas where it collected. There the erasing action ceased, causing even more toner deposition along that strip of the drum during development, and so producing positive feedback to exacerbate the problem. Eventually the cleaning blade became overwhelmed and allowed past vast quantities of toner which penetrated the ion cartridge. This completely stopped the operation of the ion cartridge, and it was never found possible to clean one so that it worked completely properly again.

This problem was partially cured by wiring the printer so that the erase rod was permanently energised. However this only reduced the problem so that about 20 g/hr of toner was carried over during idling, allowing the printer to run for several hours before toner started to fuse to the erase rod. It was reported from BUSM that with the exhibition machine the transfer corona greatly affected the rate of toner carry over. If the corotron was not energised then the carry over rate dropped markedly, but the image was very feint. Tests on the experimental machine confirmed this and also showed that there was a threshold at -5.0 KV on the corotron above which the carry over rate increases very markedly, as shown in fig. 10.1. This coincides with the minimum voltage at which consistently good image transfer to the shoe components occurs. By operating at a corotron voltage of -5.0 KV, instead of the previously used -6.0 to -6.5 KV, good quality printing was obtained with a carry over rate of about 0.6 g/hr. The rise in the rate of toner carry over with increase in the corotron voltage may be caused by additional charge being deposited on the image drum by the corona, which

is not completely neutralised by the erase rod.

The problem of efficiently cleaning the drum was made worse by debris from leather and elsewhere becoming trapped between the drum and the cleaning blade, allowing toner to pass there. For reliable continuous operation a more efficient drum cleaning mechanism is needed to minimise the possibility of contamination of the erase rod and ion cartridge. Such mechanisms, involving rotating brushes and electrostatics are discussed by Nebenzahl et al [73] and in the IBM Technical Disclosure Bulletin [74].

Concern had also been expressed about the possibility of a continuous build up of electrical charge on the take–up conveyor belt from the transfer corotron, in the same way as in a Van de Graaff generator. The charge on the belts was measured using an Autelco static locater spaced 100 mm from the belts at various locations along them. With the machine operating with the corotron unenergised no charge was measured at a position half way between the scanning station and the printer. Where the belt emerged from the printer a potential with respect to earth varying from zero to -2 KV as different parts of the belt passed by was measured after a few seconds. This pattern of voltages remained the same after the first few seconds of belt movement, with no further increase, and the -2 KV was always measured as the same part of the conveyor passed by. With the transfer corotron energised to -6.3 KV there was again no charge measured between scanning and printing, but the voltage emerging from the printer had altered to range from -0.5 KV to 1.5 KV, where the part of the belt that was previously -2 KV was now 1.5 KV. This patch may correspond to a certain type of construction within the belt. These results indicate that the take–up conveyor belt was becoming charged during its passage through the printer, but was then losing the charge, probably through the earthed metal drive rollers, before it arrived there again. The varying electrostatic properties of the conveyor may effect the efficiency of toner transfer from the print drum to the shoe components, but there was no noticeable variation in the density of the printed image.

The availability of black toner only for the printer was unacceptable since it is very difficult to see when printed on black shoe materials, which

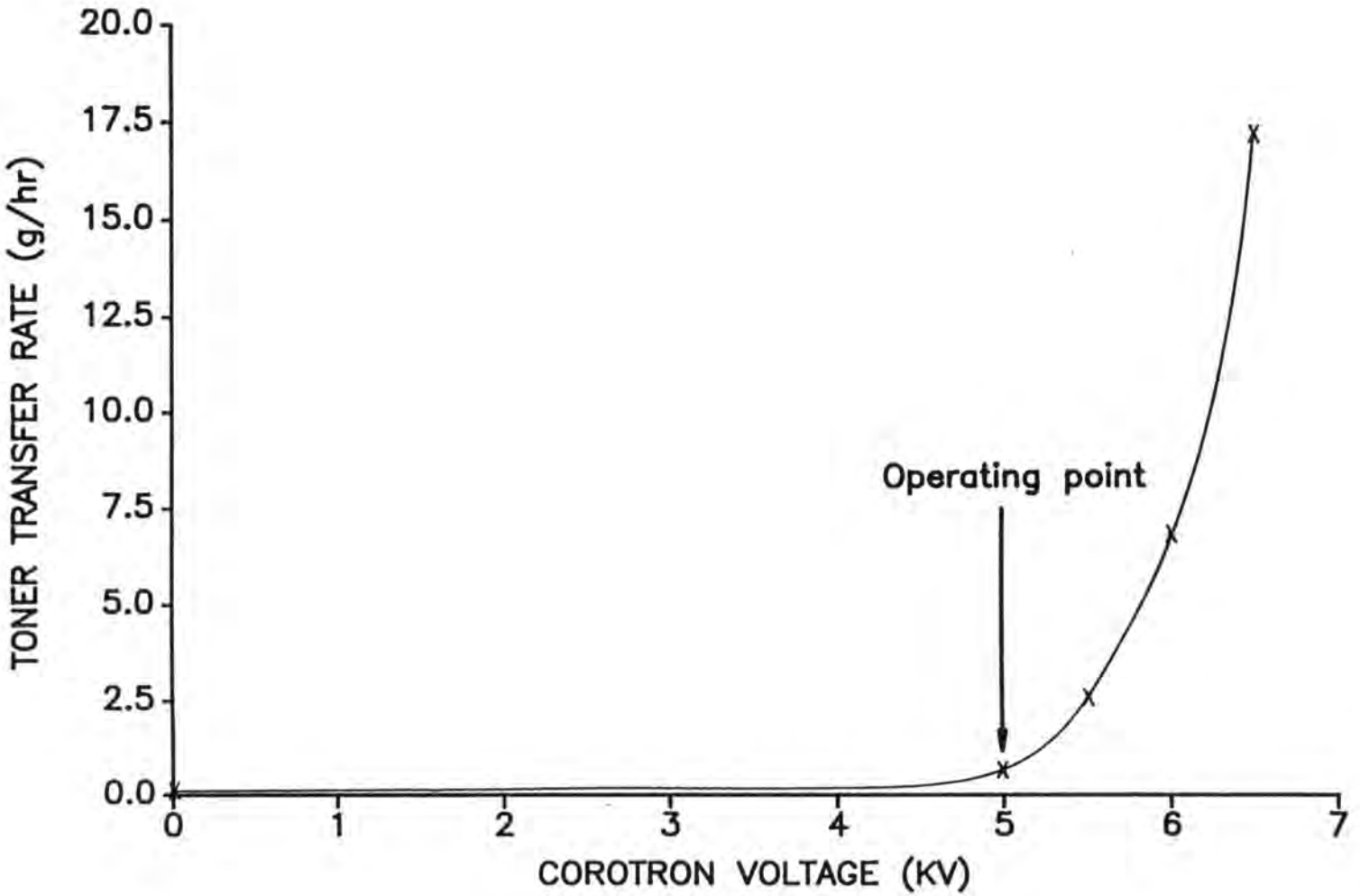


Figure 10.1 *Variation of the rate of toner carry over with the corotron voltage, during idling.*

are quite common! The solution to this problem was thought to lie in the use of a grey toner, since this would give the required contrast on black materials and would also be visible on the majority of grey materials, since it is very difficult to match grey exactly both in colour and shade. In order to demonstrate this about 10 % by volume of titanium dioxide white pigment was carefully blended into the black toner. The toner was turned dark grey by the individual particles retaining some of the white pigment on their surfaces. Stitchmarking was successfully printed on a variety of coloured shoe materials with this toner, though the titanium dioxide tended to separate out in the developer unit and contaminate the conveyor, top rollers, and shoe components. A sample of proper grey toner has been made by the toner manufacture and has been successfully tested for stitchmarking.

By–products of the ion generation and corona processes are ozone and nitrogen compounds. Since these are harmful at high concentrations they ought to be removed by drawing the air from around the printer through an activated charcoal filter, as used in many photocopiers and the Hewlett Packard laser printer which was used for the early stitchmarking experiments.

10.3 General machine operation

The mechanical parts of the machine worked very well, helped by their simplicity and the lack of any reciprocating masses, and in particular, the toothed conveyor belts ran completely trouble free, with no signs of wear. Their only slight shortcoming is that they do not lie very flat across their widths when pulled between the drive rollers. However, the belts are forced to be straight at the scanning gap where they are tight on the drive rollers, and between scanning and printing where the top rollers press the take–up belt flat.

With three large stepper motors bolted directly to the framework the machine was rather noisy in operation. This could be greatly reduced by resiliently mounting the motors so that less vibration is passed on to the framework, and by applying acoustical damping material to the framework to reduce its ringing.

10.4 Summary

This work has involved the choice of a printing technology for integration with a previously developed recognition system for high speed automatic stitchmarking of shoe upper components. From a consideration of the possible configurations of recognition systems it was shown that the most appropriate was being used, utilising the best of the known shape features for use with large numbers of shoe components. The evaluation of the recognition system showed that it performed very well with the 600 shape database. With the standard recognition system parameters the rejection rate was zero, though the substitution rate was higher than had been hoped. However, this was still impressive considering the similarity of some pairs of the shoe components. The similarity between shapes which have to be recognised as different was not listed in the specification, and is not an easy thing to quantify. Positive use could be made of this by scanning the card patterns of a new design of shoe and seeing if any are matched to previously used components within the database. If so, it would probably be possible to use the previous component in the new design, thereby reducing the number of different components, their cutting knives, and other tools. Even so, more work is needed to determine if the recognition parameters are optimum for use with the large database that would result from teaching the large number of shapes of a typical shoe factory.

The stitchmarking machine met the required specification in terms of of the overall process time, though a potential bottleneck with 'bad' shapes when the database is large was identified. An investigation of other methods of dealing with the orientation of 'bad' shaped components showed that there is no technique presently known that is better in terms of speed, accuracy, and range of applicability than the polar coding method. A speed improvement can be made to this method by modifying the algorithm, by employing a more efficient computer language, and by using a faster microprocessor.

The machine demonstrated the effectiveness of using a gap between the conveyor belts through which the components are scanned with the elimination of problems with dirt or scratches. However, there was a cost to this as was demonstrated in the increase in the spread of the features of scanned

shapes which did not remain flat as they crossed the gap. If this loss of performance proves unacceptable then the machine could be improved by the methods described to keep the components flatter during scanning.

The reasons for the selection of the ionographic printer, and its modification for printing onto shoe materials were described in detail. The printer has operated successfully on the variety of shoe materials with the range of thicknesses normally encountered in shoe manufacture. Although there were problems concerned with the high rate of toner carry over these have been largely overcome.

The work with the machine also demonstrated the advantages of using a dedicated graphics processor with a video frame store for the fast manipulation and interpolation of image data, and in transferring the data at high speed to a video device.

The accuracy of printing the stitchmarking on the shoe components was not quite within the specification, though this appeared to be largely due to the shoe components moving relative to the conveyor belts. The method of preventing the components moving on the belts by using top idler rollers was unsatisfactory and a more effective method is required, as was explained.

10.5 Future directions

Details have been obtained of a newly available ionographic printer with a print width of 365 mm (14.4 inch), a resolution of 7.9 dots/mm (200 dots/inch), and a printing speed of up to 1.5 m/sec. The printing mechanisms are being marketed for retrofitting to high speed presses by the Idax division of the Dennison Manufacturing Company. Use of one of these would allow the required width specification to be met and would have plenty of scope for increasing the operating speed of the machine. It is proposed to use a graphics processor with a larger video memory to interface to the printer so that its full resolution can be used and so the image does not have to be sent in slices.

The throughput of the machine will be increased by using faster microprocessors, recoding where appropriate, and by implementing concurrent

processing. It should then be in a suitable state for commercial production as a stand–alone automatic stitchmarker. Future processes which could be linked to the vision system are being considered. The conveyor could be extended from the printer to a following process while preserving the knowledge of the class names, positions, and orientations of shoe components upon it. In this way the machine would form the basic component of an automatic shoe upper assembly line.

REFEFERENCES

1. Cole, A.J. **"Survey of preparation–for–closing operations."** Internal report *BUSM Ltd.* Nov. 1978.

2. Crouch I.P. **"Shape recognition of shoe uppers."** Internal letter *BUSM Ltd.* Nov. 1979.

3. Koulopolos, C. **"Automated vision for identification of silhouette shapes with an application to industrial automation."** PhD thesis City University, London. 1982.

4. Browne, Arthur, and Norton–Wayne, Leonard. **"Vision and Information Processing for Automation."** *Plenum Press.* New York & London. 1986.

5. Pavlidis, T. **"A review of algorithms for shape analysis."** *Computer graphics image processing.* Vol 7. 1978. 243–258.

6. Pavlidis, T. **"Algorithms for Shape Analysis of Contours and Waveforms."** *IEEE. Trans. Pattern Analysis and Machine Intelligence.* Vol. PAMI 1–2. 1980. 301–312.

7. Freeman, H. **"On the encoding of arbitrary geometric configurations."** *IRE. Trans. Electron. Comput.* Vol. EC–10. No. 2. 1961. 260–268.

8. Ho, C–S. **"Precision of digital systems."** *IEEE. Trans. Pattern Analysis and Machine Intelligence.* Vol. PAMI–5. No. 6. Nov. 1983. 593–601.

9. Dessimoz, J–D. **"Sampling and smoothing curves in digitised pictures."** *Proc. 1st. EUSIPCO.* Lausanne. Sept. 1980.

10. De Coulon, F. and Kammenos, P. **"Polar coding of planar objects in industrial robot vision."** *Neue Technik.* No. 10. Oct. 1977. 663–671.

11. Teague, M.R. **"Image analysis by the general theory of moments."** *J. Opt. Soc. Am.* Vol. 70. No 8. Aug. 1980. 920–930.

12. Grogan, T.A. and Mitchell, O.R. **"Shape recognition and description: a comparative study."** *Purdue University.* Lafayette, In. Report TR–EE83–22. July 1983.

13. Ming–huey, H. **"Visual pattern recognition by moments invariants."** *IRE. Trans. on Information Theory.* Feb. 1962. 179–187.

14. Smith, F.W. and Wright, M.H. **"Automatic ship photo interpretation by the method of moments."** *IEEE. Trans. on Computers.* September 1971. 1089–1095.

15. Dudani, S.A., Breeding, K.J., McGhee, R.B. **"Aircraft identification by moment invariants."** *IEEE. Trans. Computers.* Vol C–26. No. 1. Jan. 1977.

16. Zahn, C.T. and Roskies, R.Z. **"Fourier descriptors for plane closed curves."** *IEEE. Trans. on Computers.* Vol. 21. 1972. 269–281.

17. Stearns, S.D. **"On selecting features for pattern classifiers."** *The third international joint conference on pattern recognition.* Nov. 8–11. 1976. Coronado, Calif. 71–75.

18. Norton–Wayne, L. **"Workpiece identification apparatus."** *British Patent 2067326.*

19. Kirsch, R.A., Cahn, L., Ray, C., Urban, G.H. **"Experiments in processing pictorial information with a digital computer."** *Proc. Eastern Joint Computer Conf.* Vol. 12. 1957. 221–229.

20. Tsai, J.S. and Sheng, C.L. **"A preprocessing method to identify**

pattern classes with a deterministic pattern function." *Dept. Elec. Engrg. University of Ottawa.* Ontario. Tech. Rept. Nov. 1967. 67–72.

21. Horn, B.K.P. **"A problem in computer vision: orienting silicon integrated circuit chips for lead bonding."** *Computer Graphics and Image Processing.* 1975. 4. 294–303.

22. Tsuboi, Y., Shiraishi, T., Kosaka, N. **"Positioning and shape detection algorithms for an industrial robot."** *Systems, computers, controls.* Vol. 4. 1973. 8–16.

23. Kitchin, P.W. and Pugh, A. **"Processing of binary images."** *First SERC. Vacation School on Robotics.* University of Hull. Sept. 1981.

24. Armbruster, K., Martini, P., Nehr, G., Rembold, U., Olzmann, W. **"A very fast vision system for recognizing parts and their location and orientation."** *Proc. 9th. Int. Symp. on Industrial Robots.* Washington D.C. Mar. 1979. 265–280.

25. Martini, P. and Nehr, G. **"Recognition of angular orientation of objects with the help of optical sensors."** *Industrial Robot.* Vol. 6. June 1979. 62–69.

26. Kammenos, P. **"Performance of polar coding for visual localisation of planar objects."** *Proceedings of the 8th. International Symposium on Industrial Robots.* Boblingen. May 1978. 143–154.

27. Zurcher, L–M. **"An Industrial Robot with Visual Perception – Conception d'un syteme de perception visuelle pour robot industriel."** Edited translation of the report. *Ecole Federale Polytechnique de Lausanne.* 1980.

28. Blum,H. **"A transformation for extracting new descriptors of shape."** in "Models for perception of speech and visual form." *M.I.T. press.* Boston. 1967.

29. Agin, G.J. **"Computer vision systems for industrial inspection and assembly."** *Computer.* Vol. 13. No 5. May 1980.

30. Birk, J.R. Kelley, R.B. Martins, H.A.S. **"An orienting robot for feeding workpieces stored in bins."** *IEEE. Trans. on Systems, Man, and Cybernetics.* Vol. SMC–11. No. 2. Feb. 1981. 151–160.

31. Vuylsteke, P., et al. **"Video rate recognition of plane objects."** *Sensor Review.* July 1981. 132–135.

32. Blower, N. **"Determination of shape orientation using coordinates of edge points."** Internal calculation sheet. *BUSM Ltd.* Dec. 1981.

33. Foith, J.P., Eisenbarth, C., Enderle, E., Geisselmann, H., Ringhauser, H., Zimmermann, G. **"Real–time processing of binary images for industrial applications."** in 'Digital Image Processing Systems'. *Springer–Verlag.* Berlin. 1981.

34. Cronshaw, A.J. **"Automatic chocolate decoration by robot vision."** *12th. International Symposium on Industrial Robots.* Paris. 9–11th. June. 1982.

35. **"La manipulation robotisee des pieces souples."** Press release *Centre Technique Cuir.* Lyon. 1988.

36. **"No. 5 Stitchmarking machine – BUUR/5SM."** Data sheet 3/46. *BUSM Ltd.*

37. Sweet, R.G. **"High frequency recording with electrostatically deflected ink jets".** *Rev. Sci. Instr.* Vol. 36. Jan. 1965. 131–136.

38. Smith, B. and Simonson, E. **"Ink jet printing for textiles."** *Textile chemist and colorist* Vol. 19. Part 8. Aug. 1987. 23-29.

39. **Epson M3740 Thermal Print Mechanism** Specification sheet *Epson (UK) Ltd.* Wembley. 1984.

40. Keeling, M.R. **"Ink jet printing."** *Physics in technology.* Vol. 12. 1981. 196–203.

41. Carr, W.W., Tincher, W.C., Cook, F.L., Lanigan, W.R. **"New technologies for fabric printing."** *Book Pap. – Int. Conf. Exhib.,*

AATC. 1986. 176–177.

42. Thorn, Geoffrey. **"Bubbles for speed."** *Systems International.* April 1986. 41.

43. Ewer, Paul. **"Tailored thermal mechanisms."** *Systems International.* May 1986. 57–61.

44. Anczurowski, E. and Sanders, D.J. **"Materials for thermal transfer printing."** *Journal of Imaging Technology.* Vol. 13. No. 3. June 1987. 97–102.

45. Frohbach, Hugh F. **"Electrostatic electrographic printing."** *IEEE. Trans. on Electron Devices.* Vol. ED–19. No. 4. April 1972.

46. Maebara, Akiyoshi. **"Electroprinting Tubes and Their Applications to Television Facsimile."** *IEEE. Trans. on Electron Devices.* Vol. ED–19. No. 4. April 1972.

47. Stone, J. James. **"The Videograph electrostatic printing process."** *IEEE. Trans. on Electron Devices.* Vol. ED–19. No. 4. April 1972.

48. Sukegawa, C. et al. **"Telenewspaper receivers for facsimile broadcast with electrostatic recording."** *IEEE. Trans. Electr. Dev.* Vol. ED–19. No. 4. April 1972. 548–556.

49. Schaffert, R.M. **"Electrophotography."** *Focal Press.* London. 1975.

50. Kimura, M., Kondo, I., Horie, M. **"High speed electrostatic recording for facsimilies."** *Fujitsu Sci. Tech. J.* Vol. 17. No. 1. March 1981. 73–76.

51. **CalComp 5754 electrostatic plotter."** Information sheet. *CalComp Ltd.* Bracknell. 1985.

52. Vahtra, U., Wolter, R.F. **"Electrophotographic process in a high speed printer."** *IBM. Journal of research and development.* Vol. 22. No. 1. Jan. 1978. 34–39.

53. Elzinga, et al. **"Laser electrophotographic printing technology."**

IBM. journal of research and development. Vol. 25. No. 5. 1981.

54. **"LZR–2600 Product description."** *Dataproducts Ltd.* Reading. Rev. 1. 6.7.85.

55. **"MP6090 90 page–per–minute non–impact printer."** Information sheet. *Bull Peripherals U.K.* Sunninghill, Berks. 1986.

56. Rumsey, J.R. and Bennewitz, D. **"Ion printing technology."** *Journal of Imaging Technology.* Vol. 12. No. 3. June 1986. 144–151.

57. Cook, F.L. **"No–solvent finishing: a coming reality."** *Textile World.* Vol. 136. Jan. 86. 81–82.

58. Carr, W.W., Tincher, W.C., Cook, F.L., Lanigan, W.R. **"New technologies for fabric printing."** *Book Pap. – Int. Conf. Exhib. AATC.* 1986. 176-177.

59. Beard, J. **"Photocopied textiles march into Georgia."** *New Scientist.* No. 1659. 8 April 1989, 32.

60. Firth, R.V.D., Goff, W., Jenkins, W.M. **"Vacuum belt picker seperator."** *IBM. Technical Disclosure Bulletin.* Vol. 25. No. 11B. April 1983. 5983–2984.

61. Newman, William M. and Sproull, Robert F. **"Principles of Interactive Computer Graphics."** 2nd. ed. *McGraw–Hill.* New York. 1979.

62. Graveling, F.J. **"A study on the use of computer controlled X–Y plotters for stitchmarking shoe components."** Internal report B/77/2 *BUSM Ltd.* Dec. 1977.

63. Chasen, Sylvan H. **"Computer Principles and Procedures for Computer Graphic Applications."** *Prentice–Hall.* New Jersey. 1978.

64. Giloi, Wolfgang K. **"Interactive computer graphics."** *Prentice–Hall.* New Jersey. 1978.

65. Blower, Nigel. **"Computer simulation using ellipses, rectangles**

& triangles to find orientation using moments of area." Internal calculation sheet 11/82 *BUSM Ltd.* April 82.

66. Landheer, D. and Devitt, E.B. **"A high current density ion generator."** *Photographic Science and Engineering.* Vol. 27. No. 5. Sept./Oct. 1983.

67. Case, J. and Chilver, A.H. **"Strength of materials."** *Edward Arnold.* London. 1959.

68. Norton–Wayne, L. and Obray, C.D. **"Prediction of error rates for coding pattern recognition."** *Unpublished paper.* 1982.

69. **"CCD145 2048–element linear image sensor."** Data sheet. *Fairchild Camera & Instrument (UK) Ltd.* Potters Bar, Herts.

70. Persoon, E. and Fu, K–S. **"Shape discrimination using fourier descriptors."** *IEEE. Trans. on Systems, Man, and Cybernetics.* Vol. SMC–7. No. 3. March 1977.

71. Crimmins, T.R. **"A complete set of fourier descriptors for two–dimensional shapes."** *IEEE. Trans. on Systems, Man, and Cybernetics.* Vol. SMC–12. No. 6. Nov./Dec. 1982, 848–855.

72. Beauchamp, K.G. and Yuen, C.K. **"Digital methods for signal analysis."** *George Allen & Unwin.* London. 1979.

73. Nebenzahl, L. et al. **"Forces involved in cleaning of an electrographic printer."** *Photographic Science and Engineering.* Vol. 24. No. 6. Nov/Dec. 1980. 293–298.

74. Anon. **"Corona design for fiber brush photoconductor cleaning system."** *IBM. Tecnical Disclosure Bulletin.* Vol. 29. No. 4. Sept. 1986. 1836–1837.

APPENDIX A

The interface to the printer

An interface board was required in order to connect the TMS34010 Software Development Board (SDB) to the ionographic print engine's de–skew board, as explained in sections 4.6 & 5.11. A schematic diagram of the interface that was designed and used is illustrated in fig. A.1, and the waveforms present are represented in figs. A.2 & A.3.

The first, and simplest requirement of the interface is to receive the differential vertical (VSYNC) and horizontal (HSYNC) synchronisation signals from the de–skew board and pass them on to the SDB. The use of differential signals for communications with the de–skew board allows these lines to be long, whereas the lines to the SDB, involving signals between TTL (transistor–transistor logic) devices, were made as short as practical. The length of the HSYNC pulse originating from the de–skew board is variable since it is used to buffer the rate of data reception at the board against its rate of usage, depending on the precise speed of the image drum. Since the falling edge of HSYNC performs the synchronisation process it has to be inverted, as depicted in fig. A.2, to produce exact synchronism of the transmitted data.

The second, and more important requirement of the interface is to take the 4–bit parallel video data from the SDB and send it serially to the de–skew board. The data has to be synchronised with RCLCK (the return clock), which is derived from DOTCLCK (the dot clock) transmitted by the

de–skew board. Also DOTCLCK has to be divided by 2 for use as the clock input to the SDB. This latter task is performed by the counter IC4.

The video data is changed from parallel to serial using the 4–line to 1–line data selector, IC1. The input line which is selected as output is determined by the code on the A & B inputs, derived by counter IC3 from DOTCLK. RCLCK, which latches the data into the de–skew board during its high to low transitions, as depicted in fig. A.3, is derived directly from DOTCLK, but is gated by the AND gate. This gating, and also the gating of the serial data and the initialisation of counter IC3, is derived from the data–enable (DATEN) output of the SDB, which performs blanking during the synchronisation periods.

The integrated circuits used are :–

IC1 – 74LS153, 4–line–to–1–line data selector.

IC2 – 26LS31, Differential driver.

IC3 – 74LS163, Synchronous 4–bit driver.

IC4 – 74LS163, Synchronous 4–bit driver.

IC5 – 26LS32, Differential receiver.

7404, Hex inverter.

7408, Quad AND gate.

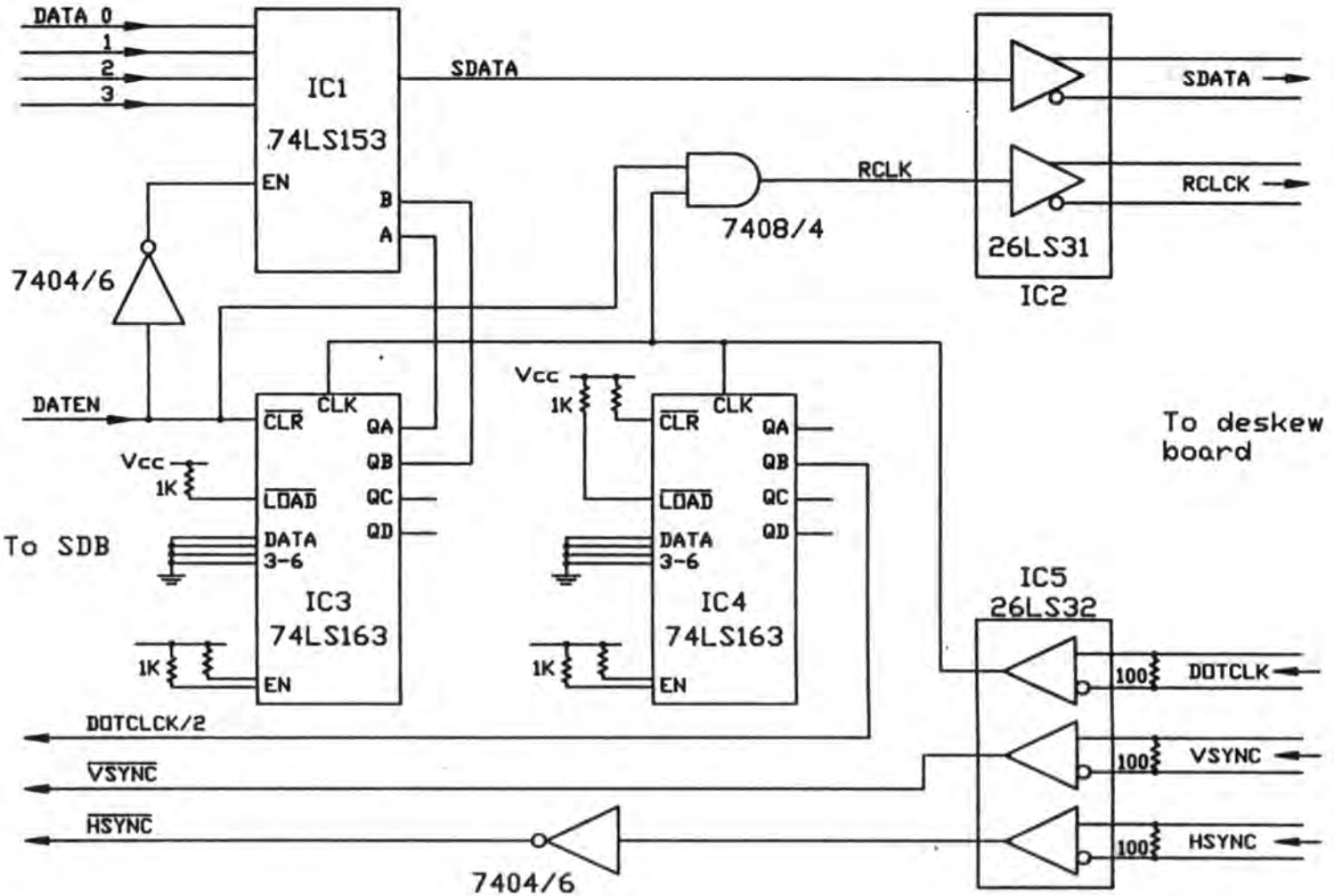


Figure A.1 *Schematic diagram of the SDB–printer interface board.*

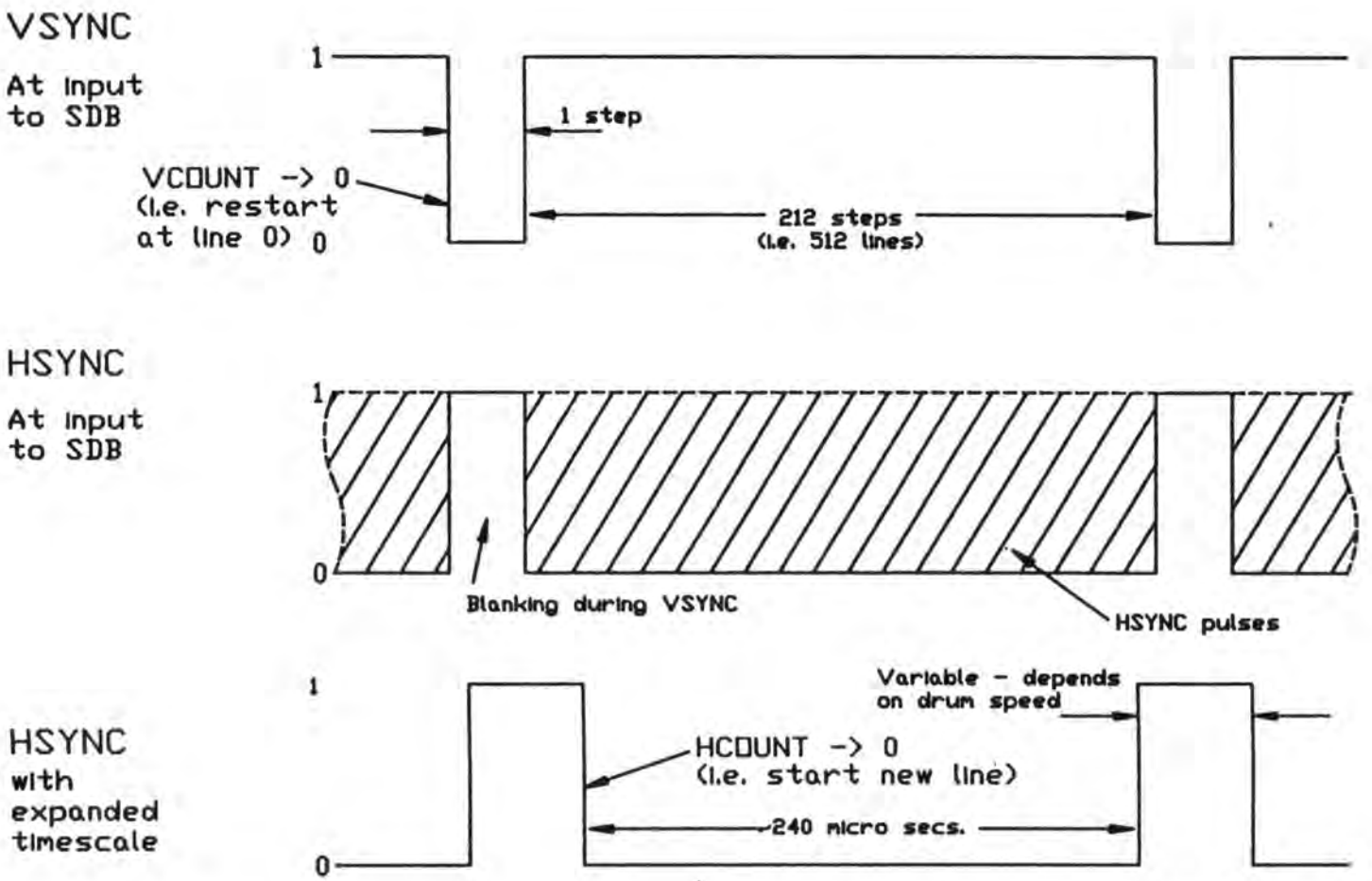


Figure A.2 *Synchronisation pulse waveforms at the SDB–printer interface.*

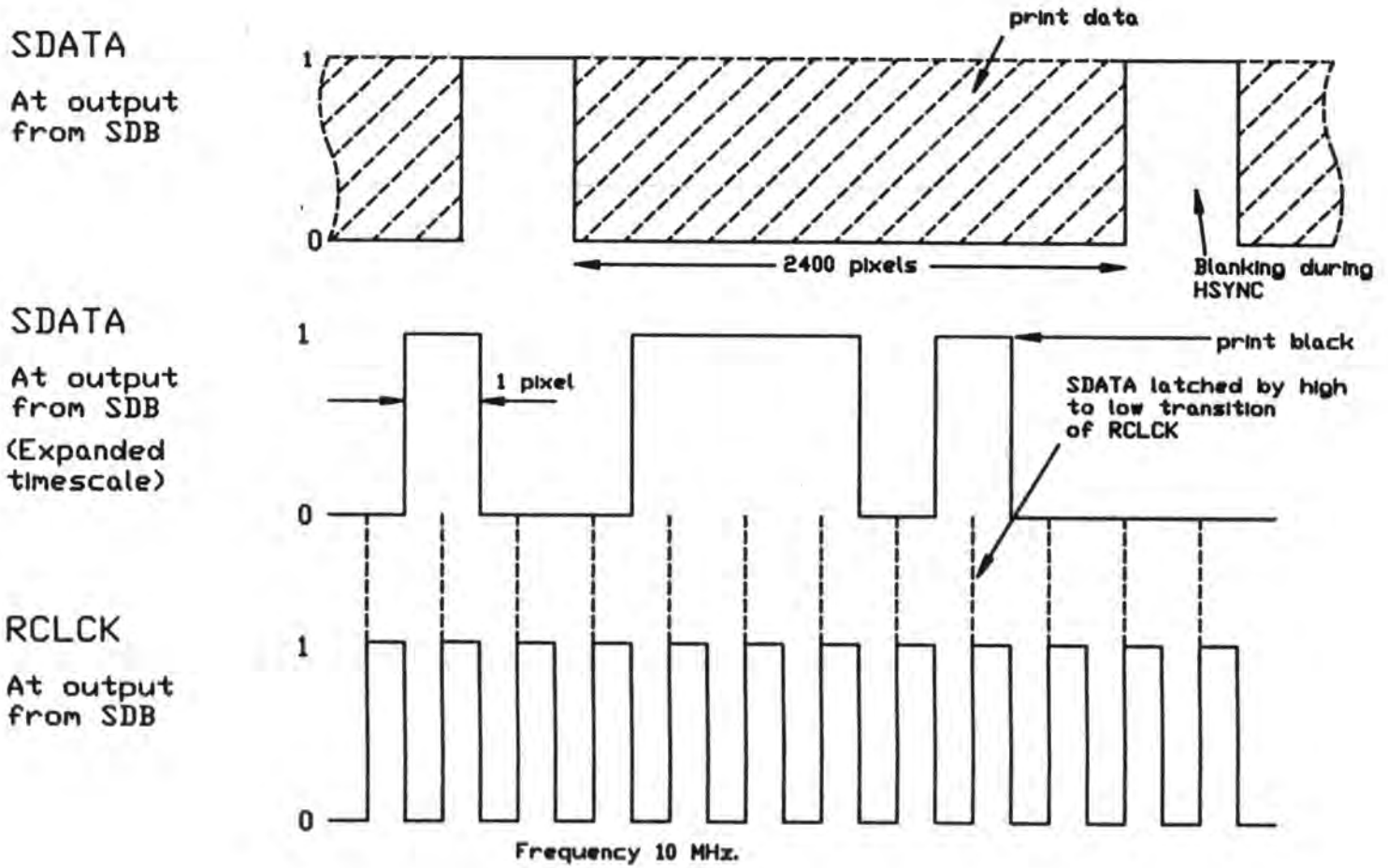


Figure A.3 *Data and clock waveforms at the SDB–printer interface.*

APPENDIX B

Stitchmark generation & printing programs

This appendix contains listings of the stitchmark generation and printing programs which run on the TMS34010 Graphics System Processor (GSP) Software Development Board (SDB). This board was modified, as explained in sections 4.6 and 5.11 to operate at 1 bit/pixel with external synchronisation.

The programs are :-

Listing B.1 *gsp_stch* – overall stitchmarking control program.

Listing B.2 *gspregs.h* – register definitions.

Listing B.3 *constant.h* – constant definitions.

Listing B.4 *gsphead.h* – constant definitions.

Listing B.5 *init_dimg* – function which initialises the GSP for creating images within the Image–RAM.

Listing B.6 *spline_solve* – function which evaluates the constants for the cubic interpolation of the splines of curves.

Listing B.7 *spline_draw* – function which sets up the stitchmarking as a pattern of 1's within the Image–RAM.

Listing B.8 *init_dzoom* – function which initialises the GSP for zooming and transferring a slice of stitchmarking image from the Image–RAM to the VRAM.

Listing B.9 *zoomdslice* – function for zooming and transferring a slice of image.

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

-------------------------------------------------------------------
 Program gsp_stch.c                                      N.R.Tout.
-------------------------------------------------------------------

        C program for generating stitchmarking and controlling printing
on a Delphax print engine. Executes on TMS34010 Graphics System Processor
(GSP) Software Development Board (SDB) modified to provide 1 bit/ pixel.

        The interpolation routines are based on those used in the graphical
stitchmarking teaching program written by R. Yardley of BUSM.

        The program is loaded from the PC, which starts it executing. The
memory is initially loaded with zeroes and, after various initialisations,
the program detects that there is no data and loops round indefinitely. After
a shoe component has been recognised the PC reads the appropriate stitch-
marking file off its disc and transfers the data to the array 'coord[ ]'
on the SDB. The rotation required for the stitchmarking and the coordinates
of the component's centroid are also loaded.

      A Non-Maskable Interrupt (NMI) is generated by the PC, and sets this
program executing again. The coordinates of the splines are transformed so
that the stitchmarking will be printed in the correct position on the
component and are loaded into the 'structure' 'spline'. The program reads
from the data the type of interpolation required and deals with the data as
appropiate. At present only cubic interpolation is used and the necessary
dummy points are loaded with the stitchmarking data and do not have to be
calculated. The positions of intermediate points in the splines are
calculated by the function 'spline_solve( )' and straight lines are drawn
between them in the low-resolution Image-RAM by the function 'spline_draw( )'
All data is passed to functions using the 'structure' 'spline', the size of
which limits each curve to 18 points plus dummy points.

        The first of the 8 slices of the stitchmarking image is expanded
and loaded into the Video RAM (VRAM) by the function 'zoomdslice( )'. The
function 'wait_scan( )' only allows the program to procede when the image
line number specified in its parameter list has been output by the TMS34010.
The subsequent slices of the image are dealt with in the same way, until
all 8 have been output, when both Image-RAM and VRAM are cleared ready for
printing the next component.

                ---------------------------------------                 */


#include "delib\gspregs.h"
#include "delib\constant.h"
#include "\h\gsp_head.h"


extern void init_dimg();
extern void init_dzoom();
extern void wait_scan();
extern void zoomdslice();
extern spline_solve();
extern spline_draw();
extern double cos();
extern double sin();

struct spline
{
        short N;                /* Number of points in this curve (max 18). */
        short x[20];            /* X-coords (up to 18 + 2 dummy points). */
        short y[20];            /* Y-coords (up to 18 + 2 dummy points). */
```

Listing B.1 (i) *gsp_stch – overall stitchmarking control program.*

```
        double ax[20];
        double bx[20];
        double cx[20];
        double ay[20];
        double by[20];
        double cy[20];
} sp, *sd;

main ()
{
long ii, num_to_clear, num_coords, num_ycoords, num_xcoords, slices;
short x_new, y_new, coord_index, type;
unsigned long source_address, *p_vram;
double sin_angle, cos_angle;
short *p_coords;
float *p_orient;
long *p_image;


/* Initialise variables. */
sd = &sp; /* Pointer to the structure. */
p_coords = (short *)SDB_COORD_RAM;  /* Pointer to the coordinate array. */
p_orient = (float *)SDB_ORIENT_RAM;  /* Pointer to the rotation required. */
p_image = (long *)IMAGE_RAM;  /* Pointer to the Image-RAM. */


/* Inititalise the processor. */
init_dimg();  /* Initialise the GSP for using the Image-RAM. */

/* Clear the Image-RAM ready for next time. */
num_to_clear = 32768;  /* Index to give one above hihest address. */
while (num_to_clear)
        {
        --num_to_clear;
        *(p_image+num_to_clear) = 0;
        }

while ( !*p_coords ) () /* Wait here if the system has just been powered up
                           and no coords have been loaded. */

/*      ** Program waits here if no stitchmarking data has been loaded.**   */
/*      ------------------------------------------------------------        */


/* Transform the stitchmarking data. */

/* Prepare for the rotation. */
sin_angle = (double) ( sin(-*(p_orient+0)) );  /* Sine of rotation angle. */
cos_angle = (double) ( cos(-*(p_orient+0)) ); /* Cosine of rotation angle. */


coord_index = 3;  /* Index to the first curve. */

while (type = *(p_coords+coord_index), type) /* While there are still
                                                curves. */
        {
        ++coord_index;

        sd->N = *(p_coords+coord_index); /* Store the number of points in
                                            this curve. */
        num_coords = *(p_coords+coord_index) +2; /* The number of points
                                        to transform (including dummies) */
        num_xcoords = num_coords;
        num_ycoords = num_coords;
```

Listing B.1 (ii) *gsp_stch - overall stitchmarking control program (cont.).*

```
            ++coord_index;
            ii = 0; /* Index for loading x-coordinates into 'spline'. */

            /* Rotate the x-coordinates in this curve through the required
               angle. */
            while (num_xcoords)
                    {
                    --num_xcoords;

                    /* Calculate x_new = x_old*cos_angle + y_old*sin_angle. */
                    x_new = (short) (*(p_coords+coord_index)*cos_angle +
                            *(p_coords+coord_index+num_coords)*sin_angle);

                    /* Translate the coordinates so that they are relative to
                       the centroid of the scanned shape. */
                    sd->x[ii] = ( x_new + *(p_coords+1) );

                    ++coord_index;
                    ++ii;
                    }

            ii = 0; /* Index for loading y-coordinates into 'spline'. */

            /* Rotate the y-coordinates in this curve through the required
               angle. */
            while (num_ycoords)
                    {
                    --num_ycoords;

                    /* Calculate y_new = y_old*cos_angle - x_old*sin_angle. */
                    y_new = (short) (*(p_coords+coord_index)*cos_angle -
                            *(p_coords+coord_index-num_coords)*sin_angle);

                    /* Translate the coordinates so that they are relative to
                       the centroid of the scanned shape. */
                    sd->y[ii] = ( y_new + *(p_coords+2) );

                    ++coord_index;
                    ++ii;
                    }


            /* Draw splines between the coordinate pairs. */
            spline_solve(sd); /* Solve the cubic spline segment. */
            spline_draw(sd); /* Draw the spline in the Image-RAM. */
            }

    init_dzoom();  /* Initialise for 'Pixel Block Transfer' (PIXBLT) of slices
                      of stitchmarking image from Image-RAM to VRAM. */

    source_address = IMAGE_RAM;  /* Start address of the Image-RAM. */

    /* Zoom and transfer the slices. */
    slices = 8;  /* Number of slices to process. */
    while (slices)
    {
            --slices;

            /* Zoom and transfer 1 slice of image. */
            zoomdslice(source_address);

            source_address += SLICE_SIZE;  /* Start address of next slice. */
```

Listing B.1 (iii) *gsp_stch – overall stitchmarking control program (cont.).*

```
        wait_scan(460);  /* Wait for this line before continuing. */
}

/* All slices of the image have been transfered, tidy up for the next
   component. */

/* Clear the VRAM. */
p_vram = (unsigned long *) 0;
while (p_vram < (unsigned long *) END_VRAM)
{
        ++p_vram;
        *p_vram = 0;
}

/* Clear the Image-RAM. */
num_to_clear = 32768;  /* Index to give one above highest address. */
while (num_to_clear)
{
        --num_to_clear;
        *(p_image+num_to_clear) = 0;
}

/* Wait here indefinitely. */
while (TRUE) {}

/* The program waits here until restarted by a NMI after the next set of
   stitchmark data have been loaded. */
}
```

Listing B.1 (iv) *gsp_stch - overall stitchmarking control program (cont.).*

```
/*  delib\gspregs.h             Register definitions. */

/*-----------------------------------------------------------------------
 *
 * Define names of special B-file registers.
 *
 *-----------------------------------------------------------------------
 */
#define SADDR     0     /* register B0  */
#define SPTCH     1     /* register B1  */
#define DADDR     2     /* register B2  */
#define DPTCH     3     /* register B3  */
#define OFFSET    4     /* register B4  */
#define WSTART    5     /* register B5  */
#define WEND      6     /* register B6  */
#define DYDX      7     /* register B7  */
#define COLOR0    8     /* register B8  */
#define COLOR1    9     /* register B9  */

/*-----------------------------------------------------------------------
 *
 * Define base location of TMS34010's on-chip I/O register file.
 *
 *-----------------------------------------------------------------------
 */
#define IO_BASE   0xC0000000    /* address of 1st I/O register */

/*-----------------------------------------------------------------------
 *
 * Define names of I/O registers.
 *
 *-----------------------------------------------------------------------
 */
#define HESYNC    0xC0000000
#define HEBLNK    0xC0000010
#define HSBLNK    0xC0000020
#define HTOTAL    0xC0000030
#define VESYNC    0xC0000040
#define VEBLNK    0xC0000050
#define VSBLNK    0xC0000060
#define VTOTAL    0xC0000070
#define DPYCTL    0xC0000080
#define DPTSTRT   0xC0000090
#define DPYINT    0xC00000A0
#define CONTROL   0xC00000B0
#define HSTDATA   0xC00000C0
#define HSTADRL   0xC00000D0
#define HSTADRH   0xC00000E0
#define HSTCTLL   0xC00000F0
#define HSTCTLH   0xC0000100
#define INTENB    0xC0000110
#define INTPEND   0xC0000120
#define CONVSP    0xC0000130
#define CONVDP    0xC0000140
#define PSIZE     0xC0000150
#define PMASK     0xC0000160
                            /* I/O registers 23-27 are reserved */
#define HCOUNT    0xC00001C0
#define VCOUNT    0xC00001D0
#define DPYADR    0xC00001E0
#define REFCNT    0xC00001F0
```

Listing B.2 *gspregs.h – register definitions.*

```
/*  delib\constant.h  */

/*-------------------------------------------------------------------------
 *
 * System constants
 *
 *-------------------------------------------------------------------------
 */

/* Define logical constants for true and false. */
#define TRUE                    1
#define FALSE                   0

/* Define return codes for system error and no error. */
#define OK                      0       /* no error */
#define SYSERR                 -1       /* error */
```

Listing B.3 *constant.h - constant definitions.*

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

 *-------------------------------------------------------------------
 * Header gsphead.h                                        N.R.Tout.
 *-------------------------------------------------------------------

       System constants.

 */


/* SDB addresses. */
#define VRAM_ADDRESS     0x00000000      /* Start of VRAM. */
#define END_VRAM         0x001ffff0      /* End of VRAM. */
#define IMAGE_RAM        0xFFD80000      /* Start of Image-RAM where the low-
                                            resolution image is drawn. */
#define SLICE_SIZE       0x20000        /* Size of a slice of Image-ram. */

#define SDB_COORD_RAM 0xFFE80000  /* Start address on the SDB of the coord
                                     space. */

#define SDB_ORIENT_RAM 0xFFE87d00  /* Start address on the SDB of the space
                                      for the rotation required.


#define NMI_LOW  0xFFFFFEE0  /* NMI vector address */
#define NMI_HIGH 0xFFFFFEF0

#define SRAM_ADR 0x04000000  /* address of shadow ram switch */

/* IBM PC addresses */
#define HOST_CONTROL_HIGH 0xC0000100
#define HOST_CONTROL_LOW  0xC00000F0

#define IHADDH     0xC0007F00  /* Address pointer high word */
#define IHADDL     0xC0007E00  /* Address pointer low word */
#define IHCTRL     0xC0007D00  /* Host control register */
#define IHDATA     0xC0007000  /* Host data register */

/* Host control register bit masks */
#define HLT       0x8000       /* Halt GSP */
#define CF        0x4000       /* cache flush */
#define LBL       0x2000       /* lower byte last */
#define INCW      0x1000       /* Increment on read cycles */
#define INCR      0x0800       /* Increment on write cycles */
#define NMIMODE   0x0200       /* NMI mode (no stack usage) */
#define NMI       0x0100       /* Non maskable int */
#define INTOUT    0x0080       /* MSG Interrupt out */
#define MSGOUT    0x0070       /* Message out */
#define INTIN     0x0008       /* MSG Interrupt in */
#define MSGIN     0x0007       /* Message in */

#define DEFCTL    0xDA00       /* default setup for CONTROL reg */
```

Listing B.4 *gsphead.h – constant definitions.*

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

-------------------------------------------------------------------
  Function init_dimg                                    N.R. Tout.
-------------------------------------------------------------------

        Based on 'init_video' in the Texas Instruments TMS34010 Graphics
Function Library.

        Initializes the video timing and screen refresh registers for use
with the Delphax print engine. The GSP is initialised for generating
images in the Image-RAM (600 dots/line x 1024 lines).
        This function must be called prior to calling any of the graphics
functions.
-------------------------------------------------------------------

/* Names of GSP I/O registers and B-file registers */
#include "gspregs.h"
/* Names of default codes for colors 0 and 1 */
#include "colors.h"
/* Definition of constants for graphics system */
#include "constant.h"

#define IMAGE_ADDRESS    0xFFD80000      /* Start of RAM for the reduced
                                            resolution image. */

/* Declare external functions. */
  extern int poke();
  extern int peek();
  extern void poke_breg();
  extern void sys_error();

/* Declare global variables. */
  extern int screensize;
  extern short xorg, yorg;
  extern int xyorigin;
  extern short penwide, penhigh;
  extern int pensize;
  extern short screenwide, screenhigh;

/* Values to be loaded into first 10 I/O registers during
 *   initialization. */
 int regvals[] =       {0x00a0, /* hesync */
                        0x0000, /* heblnk */
                        0x0096, /* hsblnk */
                        0xffff, /* htotal */

                        0x0100, /* vesync */
                        0x0000, /* veblnk */
                        0x0200, /* vsblnk */
                        0xffff, /* vtotal */

                        0xd010, /* dpyctl*/
                        0xfffc, /* dpystrt*/
                        0xffff, /* dpinit*/
                        0};

void init_dimg()
{
  int address,  /* 32-bit GSP I/O register address */
      i,        /* array index */
      *regval;  /* pointer to I/O register value array */

        regval=regvals;
```

Listing B.5 (i) *init_dimg - function which initialises the GSP for creating images within the Image-RAM.*

```
        screenwide=600;
        screenhigh=1024;

    /* Concatenate screen width and height to form 32-bit int */
        screensize = screenwide + (screenhigh << 16);

    /* Load the first 11 I/O registers with initial values. */
        address = IO_BASE;
        for (i = 0; i <= 10; i++)
        {
        poke(address, regval[i]);
        address += 16;
        }

       /*
        * Configure pixel size at 1 bit.
        */
        poke(PSIZE, 1);

       /*
        * Set x-y offset to establish memory address of pixel at top left
        *   corner of Image-RAM.
        */
        poke_breg (OFFSET, IMAGE_ADDRESS);  /* Specify start of image. */

       /* Set up CONTROL register--
        *   T = 1:  Enable pixel transparency
        *   W = 3:  Enable up window clipping, but not WV interrupt
        *   PBH = PBV = 0:  Array move starts at upper left pixel
        *   PP = 0:  Select replace operation
        *   CD = 0:  Cache-disable off.
        */
        poke(CONTROL, peek(CONTROL) & 0x00DF | 0x00C0);

       /*
        * Set default destination pitch to pitch of Image-RAM to reduce
        *   startup overhead of functions.
        */
        poke_breg (DPTCH, 1024);

       /* Set CONVDP register to correspond to contents of DPTCH. */
        asm("      LMO  B3,B0              ");
        asm("      MOVE B0,@>C0000140      ");  /* load CONVDP */
        asm("      MOVE B0,A8              ");


       /* Default origin coincides with top left corner of screen. */
        xorg = yorg = 0;
        xyorigin = xorg + (yorg << 16);

       /* Set drawing pen to its default size. */
        penwide = 1;  penhigh = 1;
        pensize = penwide + (penhigh << 16);

       /* Initialize register-based global variables for viewport 0. */
        poke_breg(WSTART, 0);
        poke_breg(WEND, screensize - 0x00010001);
        set_color0(WHITE);
        set_color1(BLACK);

}  /* end of "init_dimg" function */
```

Listing B.5 (ii) *init_dimg – function which initialises the GSP for creating images within the Image-RAM.*

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

-------------------------------------------------------------------
        Function spline_solve()                 R.Yardley / N.R.Tout.
-------------------------------------------------------------------

        Evaluate the Cardinal-spline cubic constants for the current spline.

                ---------------------------------------                 */

struct spline
{
        short N;        /* Number of coord-pairs in this spline (max 18). */
        short x[20];        /* X-coords (up to 18 + 2 dummy points). */
        short y[20];        /* Y-coords (up to 18 + 2 dummy points). */
        double ax[20];
        double bx[20];
        double cx[20];
        double ay[20];
        double by[20];
        double cy[20];
} ;

spline_solve(sd)
struct spline *sd;
{
        short i;
        double p1, p2, p3, p4;

        for (i=1; i < sd->N; i++)
        {
                /* Convert the x-coordinates to floating point. */
                p1 = (double)sd->x[i-1];
                p2 = (double)sd->x[i];
                p3 = (double)sd->x[i+1];
                p4 = (double)sd->x[i+2];

                /* Evaluate the x-coordinate constants. */
                sd->ax[i] = p4/2 - p1/2 + 1.5*p2 - 1.5*p3;
                sd->bx[i] = p1 - 2.5*p2 + 2*p3 - p4/2;
                sd->cx[i] = p3/2 - p1/2;

                /* Convert the y-coordinates to floating point. */
                p1 = (double)sd->y[i-1];
                p2 = (double)sd->y[i];
                p3 = (double)sd->y[i+1];
                p4 = (double)sd->y[i+2];


                /* Evaluate the y-coordinate constants. */
                sd->ay[i] = p4/2 - p1/2 + 1.5*p2 - 1.5*p3;
                sd->by[i] = p1 - 2.5*p2 + 2*p3 - p4/2;
                sd->cy[i] = p3/2 - p1/2;
        }
}
```

Listing B.6 *spline_solve – function which evaluates the constants for the cubic interpolation of the splines of curves.*

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

-------------------------------------------------------------------
        Function spline_draw()                  R.Yardley / N.R.Tout.
-------------------------------------------------------------------

        Evaluates the positions of 10 points within each spline and draws
within the Image-RAM straight lines between them using the library
function 'pen_line'.
        The smoothness of the curves produced can be increased by
increasing the number of points used, at the expense of increased
execution time.

                ---------------------------------------                 */

extern void pen_line();

/* Arrays of pre-calculated constants, their squares, and cubes. */
static double t[10] = { 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
                        0.8, 0.9, 1.0 };
static double ts[10] = { 0.01, 0.04, 0.09, 0.16, 0.25, 0.36,
                         0.49, 0.64, 0.81, 1.0 };
static double tc[10] = { 0.001, 0.008, 0.027, 0.064, 0.125, 0.216,
                         0.343, 0.512, 0.729, 1.0 };

struct spline
{
        short N;      /* Number of coord-pairs in this spline (max 18). */
        short x[20];          /* X-coords (up to 18 + 2 dummy points). */
        short y[20];          /* Y-coords (up to 18 + 2 dummy points). */
        double ax[20];
        double bx[20];
        double cx[20];
        double ay[20];
        double by[20];
        double cy[20];
} ;

spline_draw(sd)
struct spline *sd;
{
        short i, j;
        long xt, yt, xt_previous, yt_previous;


        /* Store first point. */
        xt_previous = (long)sd->x[1];
        yt_previous = (long)sd->y[1];

        /* Calculate for all the splines in this curve. */
        for (i=1; i < sd->N; i++)
        {
                /* Interpolate points within each spline. */
                for(j=0; j<10; j+=1)
                {
                        /* Interpolated x-coordinate. */
                        xt = (long)( sd->ax[i]*tc[j] + sd->bx[i]*ts[j]
                             + sd->cx[i]*t[j] + sd->x[i] + 0.5 );
                        /* Interpolated y-coordinate. */
                        yt = (long)( sd->ay[i]*tc[j] + sd->by[i]*ts[j]
                             + sd->cy[i]*t[j] + sd->y[i] + 0.5 );

                        /* Construct a line between the previous point
                           and the newly interpolated point. */
```

Listing B.7 (i) *spline_draw - function which sets up the stitchmarking as a pattern of 1's within the Image-RAM.*

```
                        pen_line( xt_previous, yt_previous, xt, yt );

                        /* Store the coordiantes of the interpolated
                           point for use next time. */
                        xt_previous = xt;
                        yt_previous = yt;
                    }
                }
            }
```

Listing B.7 (ii) *spline_draw – function which draws the stitchmarking within the Image-RAM.*

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

------------------------------------------------------------------------
        Function init_dzoom                                  N.R.Tout.
------------------------------------------------------------------------

        Initialises the GSP for zooming and transfering a slice of the
stitchmarking image from the Image-RAM to the VRAM.
        This is the normal resolution image, and has 2400 dots/line and
512 lines.
------------------------------------------------------------------------*/

/* Names of GSP I/O registers and B-file registers */
#include "gspregs.h"

#define VRAM_ADDRESS    0x00000000      /* Start of VRAM. */

/* Declare external functions. */
  extern int poke();
  extern int peek();
  extern void poke_breg();
  extern void sys_error();

/* Declare global variables. */
  extern int screensize;
  extern short xorg, yorg;
  extern int xyorigin;
  extern short penwide, penhigh;
  extern int pensize;
  extern short screenwide, screenhigh;

void init_dzoom()
{

        screenwide=600;
        screenhigh=512;

  /* Concatenate screen width and height to form 32-bit int */
        screensize = screenwide + (screenhigh << 16);
  /* Initialize register-based global variables for viewport 0. */
        poke_breg(WEND, screensize - 0x00010001);

        poke (PSIZE, 4);  /* Pixel size for the VRAM. */
      /*
       * Set x-y offset to establish memory address of pixel at top left
       *   corner of screen.
       */
        poke_breg (OFFSET, VRAM_ADDRESS);  /* Specify start of image. */

      /*
       * Set default destination pitch to pitch of screen to reduce
       *   startup overhead of functions.
       */
        poke_breg (DPTCH, 4096);

        poke_breg (SPTCH, 1024);  /* Pitch of source. */

      /* Set CONVDP register to correspond to contents of DPTCH. */
        asm("      LMO  B3,B0              ");
        asm("      MOVE B0,@>C0000140      ");  /* load CONVDP */
        asm("      MOVE B0,A8              ");


}  /* end of "init_dzoom" function */
```

Listing B.8 *init_dzoom – function which initialises the GSP for zooming and transferring a slice of stitchmarking image from the Image-RAM to the VRAM.*

```
/* Copyright 1989 (c) BUSM LTD. / DURHAM UNIVERSITY.

*********************************************************
* c-callable subroutine for zooming a slice by 4 times and transfering
* it to VRAM.                                    N.R.Tout.

* R02 uses PIXBLT B,L for maximum speed and zooms straight from
* Image-RAM to VRAM.
* The B File registers are set up so that PIXBLT expands each bit
* of the Image-RAM to 4 bits in the VRAM.
*********************************************************
FP      .set    A13
STK     .set    A14
        .file   "zmdslice.c"
        .globl  _zoomdslice

* DEFINE CONSTANTS.
*       Copy gspregs.asm, but don't include in listing.
        .nolist
        .copy   gspregs.asm      ; I/O and B-file registers.
        .list

*********************************************************
* FUNCTION DEF : _zoomslice
*********************************************************
_zoomdslice:
        MMTM    SP,A5,A7,A10,A11,FP     ; Save registers to be used.
        MOVE    STK,FP  ; Create new local frame.
        MOVE    *FP(-32),A5,1   ; Location of the source array.

* Transfer each line 4 times.
        MOVI    1*65536+600,DYDX ; Size of array to transfer (rows:
                                   columns).
        MOVI    0,A7             ; (DADDR).
        MOVI    128,A10          ; Lines to transfer.

LOOP1:
        MOVK    4,A11            ; Times for each source line.

LOOP2:
        MOVE    A5,SADDR         ; Source array starting address.
        MOVE    A7,DADDR         ; Destination array starting address.
        PIXBLT  B,L              ; Transfer the array.

        ADDI    1000h,A7         ; To next line of destination.
        DSJS    A11,LOOP2

        ADDI    400h,A5          ; To next line of source.
        DSJS    A10,LOOP1


        MMFM    SP,A5,A7,A10,A11,FP  ; Restore registers.
        RETS    2
        .end
```

Listing B.9 *zoomdslice – function for zooming and transferring a slice of image.*

APPENDIX C

Reconstruction of shape gg23 from its Fourier harmonics

This appendix contains figures showing the reconstruction of shape gg23 from its Fourier harmonics. The Fourier coefficients had been calculated, as detailed in section 9.4.2, for determining the orientation of scanned shapes. As confirmation that the coefficients had been calculated correctly, shape gg23 was reconstructed from the first 20 harmonics. The following figures show :-

Figure C.1 Original outline of shape gg23, sampled at 4° intervals.

Figure C.2 The 1st. harmonic (or fundamental) of the outline of shape gg23.

Figure C.3 The sum of the first 3 harmonics of the outline of shape gg23.

Figure C.4 The sum of the first 10 harmonics of the outline of shape gg23.

Figure C.5 The sum of the first 20 harmonics of the outline of shape gg23.

From the figures it can be seen that after adding together the first few harmonics the resulting outline begins to look similar to the original sampled outline of the shape. This confirms that the calculated Fourier coefficients were correct.

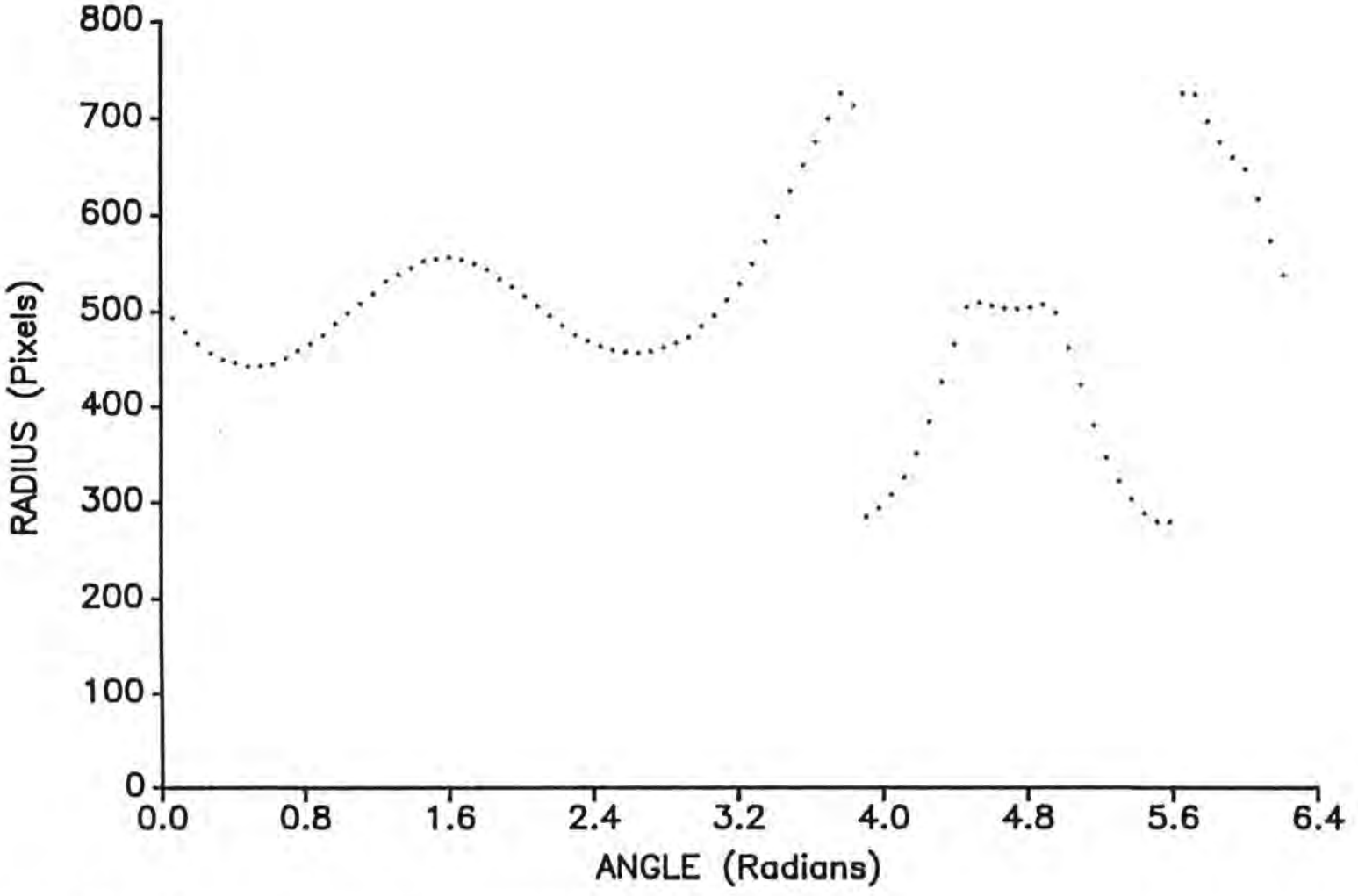


Figure C.1 *Original outline of shape gg23, sampled at 4° intervals.*

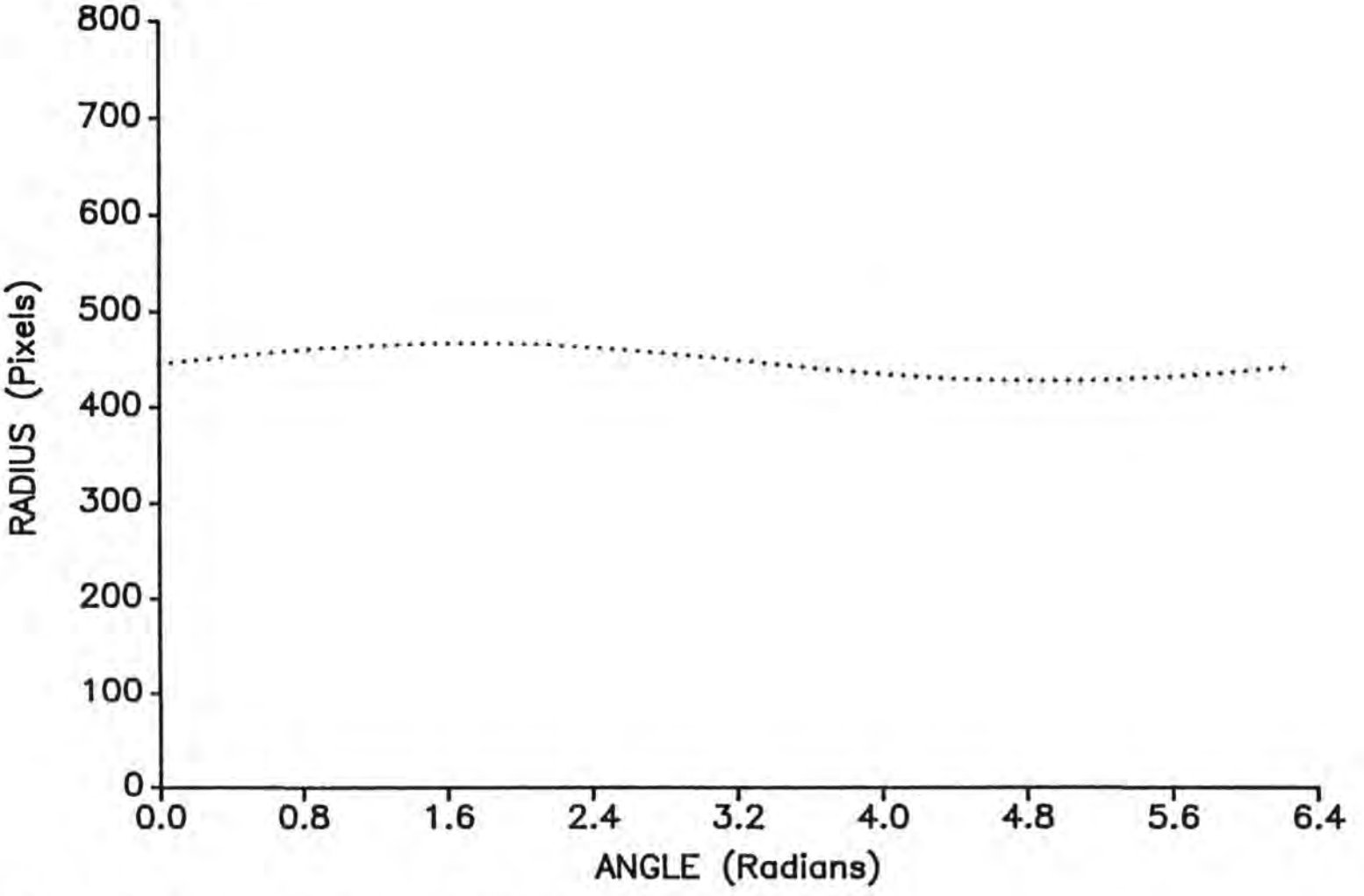


Figure C.2 *The 1st. harmonic (or fundamental) of the outline of shape gg23.*

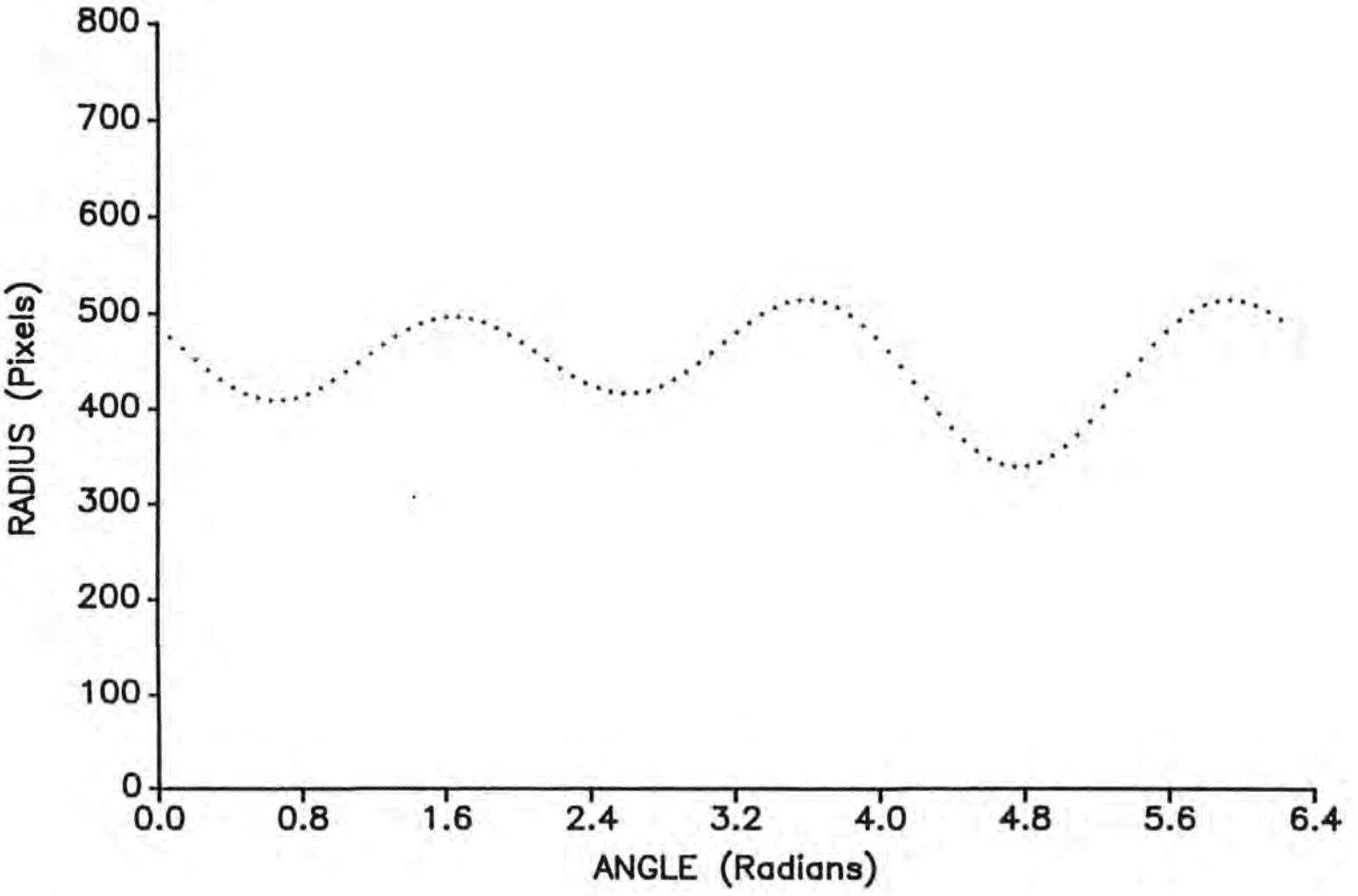


Figure C.3 *The sum of the first 3 harmonics of the outline of shape gg23.*

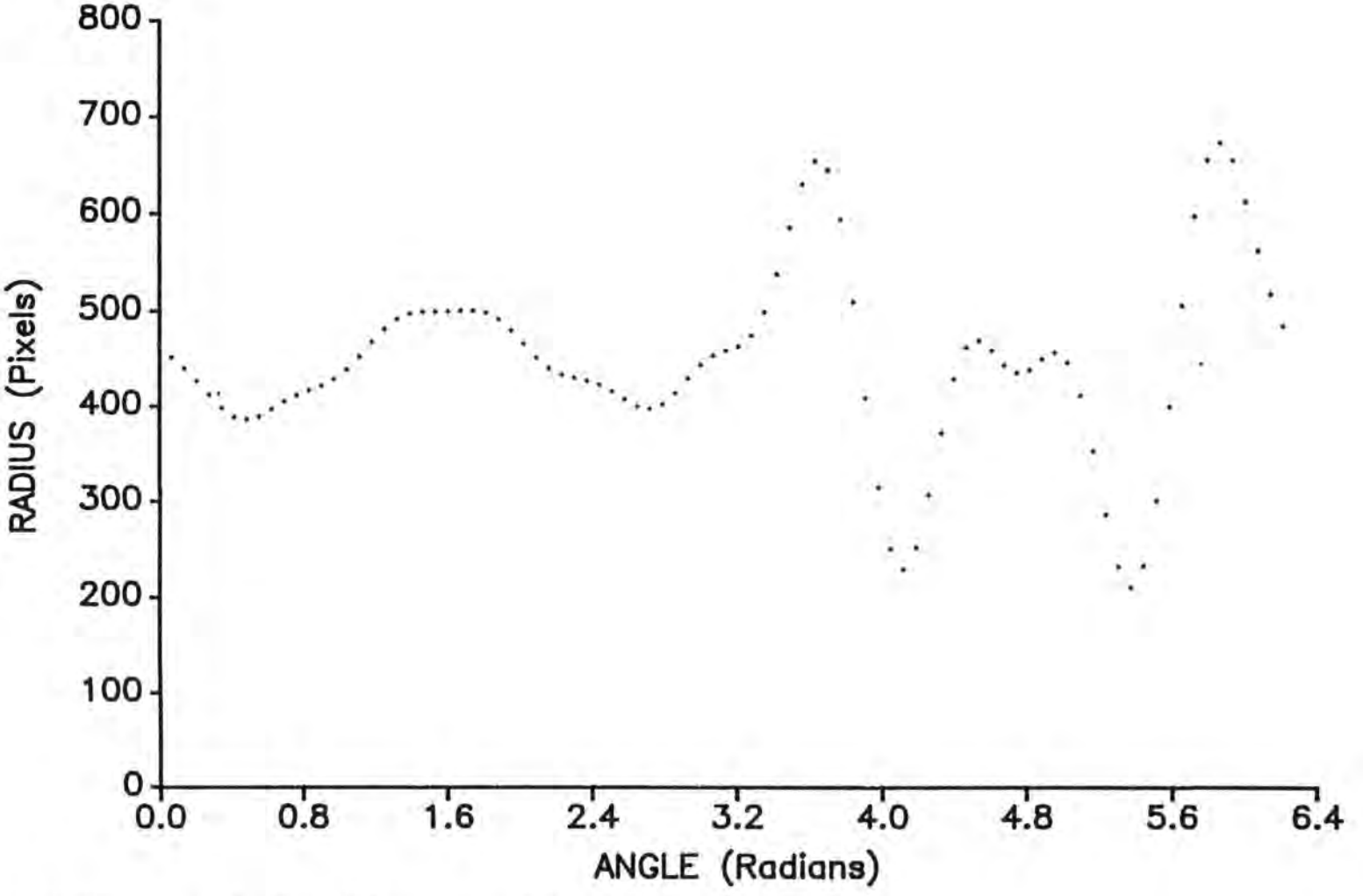


Figure C.4 *The sum of the first 10 harmonics of the outline of shape gg23.*

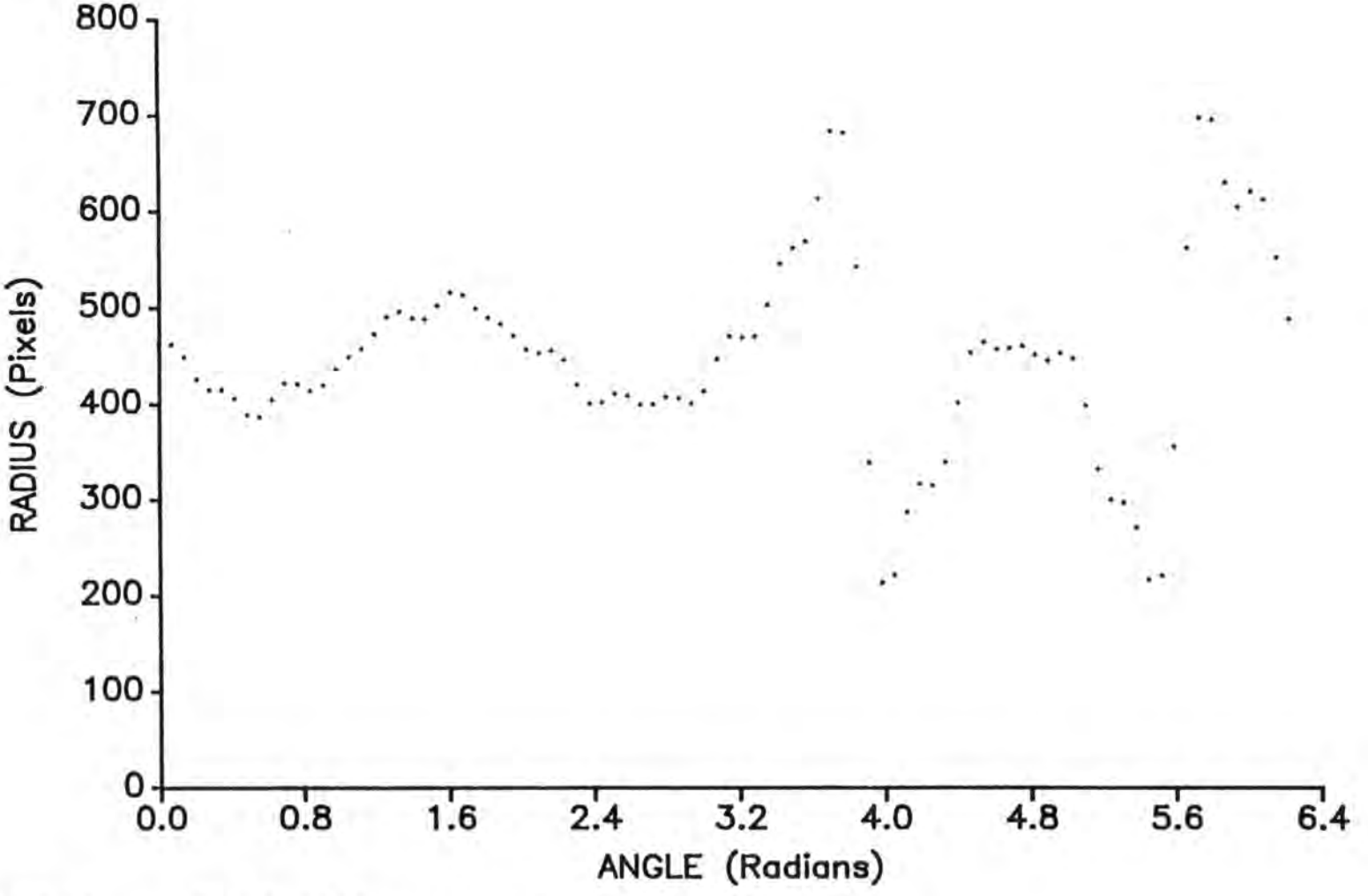


Figure C.5 *The sum of the first 20 harmonics of the outline of shape gg23.*

APPENDIX D

Equations for 3rd. & 4th. order moments of area

Appendix D contains the equations for the variation of the 3rd. and 4th. order moments of area of a shape as the origin of the coordinate system is rotated about the centroid of the shape. From section 9.4.3, the equations for the 3rd. order moments are

$$M_{30} = \sum_{1}^{m}\sum_{1}^{n} x^3 f(x,y)$$

$$M_{21} = \sum_{1}^{m}\sum_{1}^{n} x^2.y f(x,y)$$

$$M_{12} = \sum_{1}^{m}\sum_{1}^{n} x.y^2 f(x,y)$$

$$M_{03} = \sum_{1}^{m}\sum_{1}^{n} y^3 f(x,y)$$

When the coordinate system is rotated by an angle θ about the shape's centroid a pixel with coordinates (x, y) in the xy system will have coordinates (x', y') in the $x'y'$ system. Coordinates in the two systems are related by

$$x' = x\cos\theta + y\sin\theta$$

$$y' = y\cos\theta - x\sin\theta.$$

The equations for the 3rd. order moments now become

$$M'_{30} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta)^3 f(x,y)$$

$$M'_{21} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta)^2.(y\cos\theta - x\sin\theta) f(x,y)$$

$$M'_{12} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta).(y\cos\theta - x\sin\theta)^2 f(x,y)$$

$$M'_{03} = \sum_{1}^{m}\sum_{1}^{n}(y\cos\theta - x\sin\theta)^3 f(x,y).$$

The five fourth order moments are given by

$$M_{40} = \sum_{1}^{m}\sum_{1}^{n} x^4 f(x,y)$$

$$M_{31} = \sum_{1}^{m}\sum_{1}^{n} x^3.y f(x,y)$$

$$M_{22} = \sum_{1}^{m}\sum_{1}^{n} x^2.y^2 f(x,y)$$

$$M_{13} = \sum_{1}^{m}\sum_{1}^{n} x.y^3 f(x,y)$$

$$M_{04} = \sum_{1}^{m}\sum_{1}^{n} y^4 f(x,y)$$

which for the (x', y') system become

$$M'_{40} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta)^4 f(x,y)$$

$$M'_{31} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta)^3.(y\cos\theta - x\sin\theta) f(x,y)$$

$$M'_{22} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta)^2.(y\cos\theta - x\sin\theta)^2 f(x,y)$$

$$M'_{13} = \sum_{1}^{m}\sum_{1}^{n}(x\cos\theta + y\sin\theta).(y\cos\theta - x\sin\theta)^3 f(x,y)$$

$$M'_{04} = \sum_{1}^{m}\sum_{1}^{n}(y\cos\theta - x\sin\theta)^4 f(x,y).$$

These equations can then be expanded and substituted in a search for suitable relationships for defining the orientation of a shape.

APPENDIX E

Results of the two stage method for 'bad' shapes

Section 9.4.4 discussed a two stage method for recognising and determining the orientation of 'bad' shapes. The orientation was determined roughly from the principal axis, for the purpose of recognition, then the shape identity was passed to the usual 'bad' shape method to produce an accurate orientation. The results of one experiment in recognising the 'bad' shapes using this method are shown in table E.1. The shapes are listed in order of GCOEFF ('Goodness coefficient'), i.e. with the very 'bad' shapes at the top. Columns 3 & 4 show the number of radii which do not match between the scanned shape and the best matching shape when using the 'good' method and the 'bad' method, respectively. The best matching shape is always the same as the scanned shape, except where marked 'sub' in column 5, where substitution occurred.

The important information that the table highlights is that for the very 'bad' shapes the 'good' method, using the principal axis, usually gives a best match with many radii which do not match. When the details of the best match are passed on to the 'bad' method it always attains a better match. This is caused by the error in the orientation of the sets of feature radii determined by the 'good' method, which is overcome in the 'bad' method by comparing feature radii sets at 1° intervals. It can be seen that the differences between the two methods becomes negligible as GCOEFF approaches 2.0, at which point the 'good' method gives satisfactory result on its own. Also of importance is the difference in the orientations given by the two methods.

Although the figures do not signify which method provides the most accurate orientation it can be seen that the difference in the orientation given by them is high at low GCOEFF and becomes much lower as GCOEFF approaches 2.0. This again is likely to be have been caused by the inaccuracy in the determination of the orientation by the principal axis, as used by the 'good' method, at low GCOEFF. The three shapes ck73, ck74, and ck76 which have very large differences in orientation by the two methods are similar shapes with outline nearly symmetrical about two axes, as shown in fig. E.1.

Scanned shape	GCOEFF ('Goodness coefficient')	Radii not matching		Difference in the orientation given by the two methods
		'Good' method	'Bad' method	
dur18-r	0.010	11	1	3.54
dur16-r	0.013	11	1	4.53
dur18	0.013	9	1	3.78
dur16	0.014	7	0	-2.26
dur17	0.014	11	2	-3.33
ck74	0.015	0	0	87.14
ck76	0.015	0	0	181.64
dur17-r	0.015	12	1	-3.83
ck89	0.016	0	0	0.19
ck90	0.016	13	0	9.99
ck90-r	0.016	12	1	3.50
ck91-r	0.016	11	0	2.97
ck89-r	0.017	6	1	-1.32
dur19	0.017	4	2	1.49
dur19-r	0.017	7	0	2.26
ck7	0.021	5	0	0.82
ck91	0.021	11	1	-3.34
ck7-r	0.022	10	1	3.50
dur126	0.022	4	0	1.12
dur11	0.023	5	1	-1.38
dur11-r	0.025	9	1	-2.24
dur10-r	0.027	10	0	-4.23
dur12-r	0.027	3	3	0.46
dur10	0.028	8	1	3.23
dur12	0.030	8	2	2.35
dur13-r	0.033	12	0	-5.10
dur13	0.034	9	0	-3.16
ck73	0.035	4	0	181.00
dur14-r	0.036	11	2	2.56
dur15	0.036	7	0	1.69
dur14	0.037	2	0	0.81
dur15-r	0.037	7	1	-1.56

Table E.1 (i) *Recognition of 'bad' shapes by the two-stage method.*

Scanned shape	GCOEFF ('Goodness coefficient')	Radii not matching 'Good' method	'Bad' method	Difference in the orientation given by the two methods
dr10	0.039	2	1	-0.58
dr8-r	0.040	4	0	-0.91
dr9	0.040	9	0	1.79
dr9-r	0.040	3	0	-0.65
dr8	0.041	12	0	-3.14
dr11-r	0.041	2	2	0.25
dr10-r	0.042	3	1	-0.59
dr11	0.042	10	2	-2.41
dr2-r	0.052	3	0	0.83
dr6-r	0.052	0	0	0.18
dr5-r	0.053	2	0	0.66
dr7-r	0.053	5	2	-0.79
dr2	0.054	1	1	-0.51
dr6	0.054	0	0	-0.33
dr1	0.055	4	0	-0.92
dr7	0.055	6	0	1.11
shoe3-1	0.055	3	1	-0.61
dr5	0.056	6	0	1.24
shoe3-2	0.056	3	0	-1.80
dr1-r	0.057	7	1	1.17
dr3	0.057	14	8	18.21
dr3-r	0.057	1	1	sub
dur111	0.057	12	1	3.48
dur109	0.058	6	0	1.14
dur108	0.080	7	7	sub
dur110	0.089	6	5	sub
dur112	0.090	10	1	2.04
dur107	0.092	4	1	-0.89
dur102	0.100	14	5	-4.71
dur106	0.112	13	4	sub
shoe1-1	0.113	0	0	0.09

Table E.1 (ii) *Recognition of 'bad' shapes by the two-stage method (cont.).*

Scanned shape	GCOEFF ('Goodness coefficient')	Radii not matching 'Good' method	'Bad' method	Difference in the orientation given by the two methods
dur103	0.129	1	1	-0.36
dur101	0.136	1	1	0.16
dur125	0.138	3	1	0.69
dur105	0.139	4	4	-1.09
gg30	0.139	0	0	-0.07
dur104	0.140	5	1	0.84
nb1	0.141	0	0	-1.00
shoe2-2	0.142	1	1	0.76
gg23	0.143	1	1	-0.66
gg25	0.144	0	0	-0.10
ck6	0.145	5	3	0.61
ck6-r	0.145	14	11	sub
gg24	0.145	0	0	-0.28
dur113	0.146	0	0	-0.03
gg26	0.146	1	1	0.11
gg33	0.146	0	0	-0.30
gg28	0.148	0	0	-0.20
gg27	0.150	4	0	-1.22
gg32	0.150	0	0	-0.08
ck4	0.151	1	1	0.03
gg36	0.151	0	0	0.10
gg34	0.154	1	0	-0.51
ck4-r	0.155	2	2	-0.18
ck5	0.156	3	3	0.48
ck5-r	0.158	4	2	-0.51
gg35	0.161	0	0	0.00
ck2	0.162	3	3	-0.42
ck2-r	0.162	1	1	-0.23
gg37	0.162	0	0	-0.12
gg39	0.162	1	1	0.21
gg38	0.163	2	2	-0.53
gg42	0.164	0	0	0.05

Table E.1 (iii) *Recognition of 'bad' shapes by the two-stage method (cont.).*

Scanned shape	GCOEFF ('Goodness coefficient')	Radii not matching 'Good' method	'Bad' method	Difference in the orientation given by the two methods
gg44	0.169	0	0	-0.19
ck3	0.170	1	1	-0.05
ck3-r	0.170	1	1	0.09
gg41	0.171	3	0	0.77
ck1	0.175	0	0	0.15
gg68	0.175	3	3	-0.09
ck1-r	0.176	12	11	sub
gg43	0.177	0	0	0.09
gg69	0.179	4	4	-0.48
gg70	0.183	4	4	-0.34
gg67	0.184	4	3	-0.76
gg71	0.184	4	4	-0.49
gg72	0.185	2	2	0.10
gg74	0.186	4	3	-0.45
gg76	0.188	2	2	-0.03
gg75	0.190	5	5	0.36
gg77	0.190	4	4	0.63
gg73	0.191	3	3	-0.40
gg80	0.192	2	2	-0.26
gg78	0.193	4	4	-0.13
gg79	0.196	3	3	0.24
gg81	0.197	3	3	0.27

Key

sub - Substitution occurred.

Table E.1 (iv) *Recognition of 'bad' shapes by the two-stage method (cont.).*

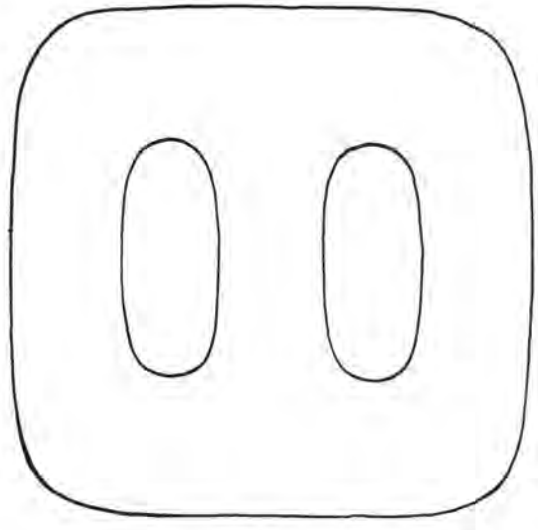


Figure E.1 *Shape ck76, actual size.*