Loughborough University

This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR/FILING TITLE
TAO, L.G.

ACCESSION/COPY NO.
040101668

VOL. NO.
CLASS MARK

17 NOV 2000
LOAN COPY

FEB 2001

FOR REFERENCE ONLY

0401016684

HIGH-SPEED PATTERN CUTTING USING REAL-TIME COMPUTER VISION TECHNIQUES

by

Li Guo Tao

B.Eng., M.Sc.

A Doctoral Thesis

Submitted in partial fulfilment of the requirements
for the award of
Doctor of Philosophy
of the Loughborough University of Technology

November, 1994

© by Li Guo Tao, 1994

Loughborough University
of [illegible] Library
Date Mar 95
Class
Acc.
No. 040101668

This thesis dedicated in loving memory to my grandmother, Zhang-Fu.

SYNOPSIS

This thesis presents a study of computer vision for guiding cutting tools to perform high-speed pattern cutting on deformable materials. Several new concepts on establishing a computer vision system to guide a CO_2 laser beam to separate lace are presented.

The aim of this study is to determine a cutting path on lace in real-time by using computer vision techniques, which is part of an automatic lace separation project. The purpose of this project is to replace the current lace separation process which uses a mechanical knife or scissors.

The research on computer vision has concentrated on the following aspects:

1. A weighted incremental tracking algorithm based on a reference map is proposed, examined and implemented. This is essential for tracking an arbitrarily defined path across the surface of a patterned deformable material such as lace. Two methods, a weighting function and infinite impulse response filter, are used to cope with lateral distortions of the input image. Three consecutive map lines matching with one image line is introduced to cope with longitudinal distortion. A software and hardware hybrid approach boosts the tracking speed to 1m/s that is 2~4 times faster than the current mechanical method.

2. A modified Hough transform and the weighted incremental tracking algorithm to find the start point for tracking are proposed and investigated to enable the tracking to start from the correct position on the map.

3. In order to maintain consistent working conditions for the vision system, the light source, camera threshold and camera scan rate synchronisation with lace movement are studied.

Two test rigs combining the vision and cutting system have been built and used to cut lace successfully.

ACKNOWLEDGEMENTS

I would like to thank;

Dr. M. R. Jackson, my Supervisor, Professor M. E. Preston, my Director of Research for their support and advice of my study, both of The University of Technology at Loughborough.

Professor T. G. King of the School of Manufacturing and Mechanical Engineering of the University of Birmingham, who was the supervisor of my first two year study, for discussion, advice and support throughout my study.

The UK Science and Engineering Research Council through its ACME Directorate, under grant number GR/F 92084 -'Vision Control of Progressive Operations on Textile Fabrics Leading to High Speed Adaptive Automation'.

Guy Birkin and Co. Ltd. for supplying lace samples and technical support.

Many Department staff members, Mr. P. A. Green, Mr. J. A. Guy, Mr. T. A. Middleton, Mr. D. Powdrill, Mr. S. G. Retter, Mr. V. Roulstone, Mr. K. W. Topley, Mr. A. Wood, for their technical support.

Finally my wife, Wei, and daughter, Ying, for their encouragement, patience and support throughout my study.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMINOLOGY

Breadth refers to a strip of lace with single pattern width. The breadth also repeats itself across the width of the knitted web at regular intervals determined at the design stage.

Centre cutting refers to a separation of a pair of the symmetrical patterns (see figure 1.1), done by cutting the waste between the mirror-image patterns.

Cutting path refers to a path on the image, which is used to guide the laser beam to cut lace.

Finding the start point refers to a search action between an image and a map to make sure that the tracking will start from the correct position on the global area.

Image refers to a digital signal of the lace to be cut.

Line scan is one image acquisition method in which a single line of photoelectronic elements of the camera is exposed by a very narrow linear image area at the same time and then digitised and stored into the memory serially.

Map refers to a reference digital signal of the lace.

Pattern refers to the unrepeated single figure on the lace required by the customer. The design of the pattern makes itself repeatable in the longitudinal direction without any visual discontinuity and has two edges in the lateral direction.

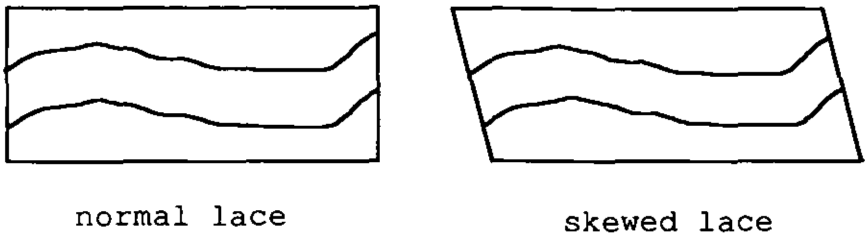
Pixel is the term used to represent a picture element in CCD camera, and digital image.

*Purl*s refer to small projecting loops on the fringe of some lace for decoration purposes (see the area near the arrows of the scalloping 1 and scalloping 2 on the Figure 1.1).

Reference path refers to a predetermined path related to the map.

Rigid lace is made from relatively inelastic yarns and extends little as the tension is increased.

Scalloping refers to the cutting process to separate the edge or fringe of the lace from the waste (see Figure 1.1) without cutting inside of the pattern, which will cause the lace to fray when it is washed, and without leaving too much waste on the finished lace which will impair its quality.



Skew is one of the lace deformations where the line linking two identical points of adjacent breaths of lace is not perpendicular to the longitudinal axis of the lace (see above figure).

Stretch lace, made from elastic yarns, can be easily extended when an external pulling force is applied to it, and will recover if the force is removed.

Tracking refers to following a predetermined (cutting) path on lace by finding proper position on the image progressively and locally.

Waste refers to the backing net between two contiguous breadths. It should be cut off during the lace separation.

Web refers to the bulk lace before being scalloped or centre cut.

Chapter 1:

INTRODUCTION

1.1 BACKGROUND

This thesis describes a high-speed computer vision system, developed to guide a laser cutting system for rapid pattern cutting. The research work for this project has been carried out in the Department of Mechanical Engineering at Loughborough University of Technology (LUT) with industrial collaboration from a textile machine builder and a machine user.

Pattern cutting, in this context, refers to the separation process by which the cutting path on a piece of patterned material is determined taking into account the pattern on the material itself and a predetermined separation scheme. The separation of a breadth of lace from the bulk material is a typical example of the process. It may be either along a single edge or down the centre of a double width to form a product which is defined by a previous manufacturing process. In this thesis lace separation will be the research example of pattern cutting.

Lace is generally a knitted patterned fabric that is often used as decoration for garments, furniture and many other consumer goods. It is primarily used in the manufacture of underwear. Potential ancestors of lace are the cross stitch articles, which existed in Egypt approximate 2000 years ago (Spencer 1983). From around 1560 AD lace, as we know, became an increasingly important feature of fashionable dress in most European countries. Early lace was made by hand with simple tools. In the last quarter of the eighteenth-century the machine-made lace industry was developed (Levey 1983) and lace manufacturing came into the new era of mass production by machines. In the early 1980's computers were introduced into the lace manufacturing process. Computer aided design and computer controlled knitting machines have made the lace industry more dynamic and flexible (Spencer 1983). A customer became able to see a lace sample within a week of selecting a design (Lowe 1984). This would have taken several weeks, or even months, in the pre-computerised era. Currently, machine made lace dominates the market. This is due to its higher productivity, quality, wider range of patterns, and relatively low cost. Figure 1.1 shows a lace sample made by machine, indicating three breadths of lace. Breadth 1 and breadth 3 are the same lace pattern. Breadth 2 is a mirror-image of breadth 1, with a horizontal offset to make the two breadths interlock. Throughout the remainder of this thesis the lace referred to will be machine made.

The manufacture of lace by machine involves many processes. The cycle of lace making starts with the requirements of the customer. Then design, draughting, knitting, setting, dyeing, dressing, separating, inspection and delivery to the customer follow. For each new design, it is necessary to carry out tests to determine the ease of breadth separation and the associated overall final quality. These results must be determined before each pattern is put into large scale production. Sometimes modification of the design will be needed if the separating test fails. In order to maintain high productivity, lace for garments is knitted in continuous lengths, with a design pattern running the full length of the web and repeated at intervals across its width. The full width of the lace is retained throughout the setting, dyeing and dressing processes. The web is then cut into individual breadths or pair of interlocked breadths according to the requirement of the customer. After final inspection they are sent to customer.

In the garment making industry the single breadth or pair of breadths will be further cut into short pieces to suit the requirement of the garments. In this thesis lace separation refers to the operation to cut lace from the whole width of the web into a single breadth.

The world market for separated lace fabric was around 5 billion metres annually and the turnover of lace separation was in the region of £1 billion annually (Preston 1989).

1.2 Basic Problem of the Mechanical Guidance System of Existing Scalloping Machine

There are two different forms of lace separating process, scalloping and centre cutting. Figure 1.1 shows examples of both processes. Currently the centre cutting is mainly done by hand using scissors, while the scalloping process is carried out on some simple but ingenious mechanical machines. A mechanical thickness guiding system and a rotary or band knife are used, but under the intensive supervision of the operators. The guiding system detects the scalloping position and presents the lace to the moving knife for cutting.

In the guiding system a contact head acts as a thickness gate to distinguish the pattern from the backing net. When lace is knitted, the distribution of the thicker and thinner threads will be determined by its design to produce a desired visual pattern and its ability to be scalloped. The visual pattern also can be interpreted into a thickness pattern if the lace is touched. The thickness pattern is important to the scalloping. The scalloping position is usually on one edge of the visual pattern and also is on the transition between thick and thin parts in the thickness pattern. The thickness differences between thicker parts and thinner parts, called thickness *contrast* in the lace industry, is the principle of the mechanical guidance system for detecting and guiding the scalloping position. In the scalloping process, the thin backing net passes through a slot between the contact head and a plate, the thick part of the lace pattern is stopped by the slot. The thicker side of the scalloping edge, or transition, touches precisely one side of the head, and the moving edge of the knife is positioned close to the contact head to carry out the cutting function (see Figure 1.2).

During scalloping the positions of the contact head and the knife do not move relatively to each other, and since most scalloping edges are not straight lines, the lace must move relative to the guidance and cutting station. There is a risk of scalloping at the wrong position if the lace moves naturally. The scalloping machine deliberately moves the lace laterally to keep the knife, and the contact head along the scalloping edge. The lateral movement of lace is determined by the profile of the edge. Often the lace cannot move laterally as scalloping requires because of the handling tension. So an operator uses his or her hand to guide the thick side of the scalloping edge against the contact head (Figure 1.3) to help the mechanical guidance system to keep the scalloping edge in the correct cutting position.

The mechanical guiding system requires that the thickness contrast at the scalloping positions is made deliberately large and kept consistent. Also the width of the backing net between any two adjacent breadths must be wide enough for the head to pass in between.

The lateral movement, and consistency of the thickness contrast of the lace are the key factors in making the mechanical scalloping machine work, but they also set several limitations to this mechanical approach.

- Firstly, the forces from the mechanical guiding system and knife will cause a considerable deformation of a stretch (or elasticated) lace, that can easily be extended by external forces, when it travels from the guiding system to the knife and thus make scalloping more difficult. This requires the scalloping speed to be reduced and the use of more skilled operators. Currently the market for stretch lace has increased dramatically.

- Secondly, the mechanical guiding system needs quite a wide gap between the breadths thus increases the waste and hence production costs.

- Thirdly, the mechanical machine cannot cut both edges of a single breadth of lace simultaneously. This is because the distance between two edges varies from one section of lace to another, which requires a varying distance between two sets of mechanical guiding and cutting systems. In addition the forces of the guiding systems and knives will interact with each other. So it is impossible to put two mechanical systems side by side to produce a single breadth. It is also difficult to arrange two systems in the pipeline (this approach has been tested in a factory but failed).

- The fourth limitation is that the pattern design is restricted with regard to the consistent thickness differential (or contrast) at the cutting position and the geometry of the permissible curves of the edges because of the low response speed of the mechanical system.

- The fifth limitation is that the edge scalloped by a knife will unravel, even when only slight damage is caused by the knife (Jackson 1993).

- Finally, this mechanical method cannot carry out centre cutting. Most of the centre cutting designs have a narrow gap between the breadths. Some of them contain crossed thick threads and some confuse the mechanical guiding system so separation has to be conducted by using scissors. This is a costly process.

In short, lace separation is one of the labour-intensive operations in the lace industry. It can be seen as one of the potential problems preventing further expansion of the lace market.

1.3 AN AUTOMATIC LACE SEPARATION PROJECT

Automation of the separation process is the answer to the problems of mechanical lace scalloping. In order to find how to apply automation to lace cutting, let us analysis different automation approaches.

The basic processes for lace scalloping can be divided into three parts: finding the cutting path, cutting it along that path and handling the lace. One approach to automate scalloping is to add an automatic lace handling device to the existing machine. This would replace the operator and compensate for the interference between the guiding and cutting devices. This poses problems: how to detect the interference, and how to establish the compensation model for the lace? If these problems were solved, this approach would help to scallop the stretched lace. But this approach does not allow the machine to do centre cutting. All these problems are due to the principles of mechanical scalloping; inherent lateral movement of the lace and critical thickness contrast.

The second approach is to use a roller on which a press knife is mounted with the contour for the scalloping or centre cutting (see Figure 1.4). When the lace passes over, it will be separated by the rotating roller. In order to cut lace accurately this approach needs some method to detect the cutting position with respect to the deformation of lace and to compensate the lace variation. The advantage is that the lace does not need to have a forced lateral movement. The problem of this approach is that the cutting force can easily distort the lace.

The third approach is to find a new arrangement in which there are no external detection and cutting forces applied to the lace which cause lateral movement when it is separated. Zero cutting, detection forces and zero lateral movement will minimise the interference between cutting and finding the cut path. This approach led to the automatic lace separation project at LUT.

The aim of the project is to replace traditionally labour-intensive lace separation operations by a real-time computer-vision guided laser cutting system. This will give higher productivity, a greater degree of responsiveness to product change, greater competitive value of the product and more flexibility of the pattern design and material choice. This will lead to just-in-time manufacture with reduced work-in-progress, while retaining or improving the high quality associated with the product and providing an economical rate of production.

The project falls into three parts,

- cutting path determination
- cutting system
- material handling.

Cutting Path Determination

An ideal cutting path determination system should find the cutting path in real time according to its pattern and predetermined position and allow for lace deformation with an industrial acceptable standard. This should not apply any external forces to the lace. A vision system will be a good candidate for this task compared with other approaches, such as contact probes, because no guiding force is applied to the lace. As computer vision systems (or elements of computer vision systems, such as camera, computer) become faster and lower cost they have been increasingly applied to the lace industry.

Cutting System

In order to avoid the problems of the existing mechanical system, a laser based high speed cutting system is proposed. The laser is one cutting tool that does not apply cutting force to the lace except the internal stresses released by the cutting (Jackson 1993). A galvanometer controlled mirror enables the laser beam to move to a required position, which eliminates one problem in the mechanical cutting process: forced lateral movement. This provides the following advantages:

- Zero cutting force, reducing web deformation.
- High speed low inertia flexible cutting point.
- Thermo sealing of the cut edge to prevent fraying.

Material Handling

The material handling system will provide a smooth lace movement with even tension. This is very important for the whole project especially at higher speed.

Visual guidance, laser cutting, and constant tension material handling systems are seen as possible components of a solution to the problem of lace separation. The combination of these has been envisaged at an early stage as Figure 1.5, in which multi edge scalloping would be possible.

How to use a laser effectively to cut lace and how to handle lace with even tension are other research topics in the overall separation project and these have been investigated by other members of the research team. The appropriate method for finding the cutting path on lace utilising a visual approach is the objective of this thesis.

1.4 COMPUTER-VISION TO LOCATE CUTTING PATH FOR LACE SEPARATION

The methods to find the cutting path on the lace can be divided into two types: contact and non-contact. The mechanical thickness detector used on the scalloping machine is one of the contact methods. The primary disadvantage of contact method is the force to be applied to lace. In the most contact methods a probe should usually contact one of the surfaces of the object to be measured and a contact force, combined with a friction force due to the relative movement between the probe and the object, will be applied to the object. In order to relax the requirement for consistency of the thickness contrast and avoid forced lateral movement, a larger area is needed to be monitored rather then just the cutting point. This can be achieved by using a probe array with sub-millimetre resolution to detect the transition position on the edge of the breadth. But the probe array still applies force to lace so that contact methods have been excluded from this research.

Typical non-contact methods are X-ray, ultrasonic and computer vision. X-ray and ultrasonic methods are widely used for non-destructive inspection inside an object. Typical applications are medical inspection or mechanical component defect inspection. Since the lace pattern is visible, computer vision is obviously the best candidate among the non-contact methods. Furthermore computer vision can get information other than the cutting position so it provides the opportunity to find a cutting path without the critical requirement for consistency of the thickness contrast, which will be useful for the centre cutting and could lead to a relaxation of the requirement for the thickness contrast along the scalloped edge.

The areas of investigation of a vision system for cutting path determination for lace separation are:

- Software development and testing for the visual guidance,
- Hardware development and testing for the visual guidance,
- Vision related developments such as optical arrangement, resolution, lighting source, processing speed.

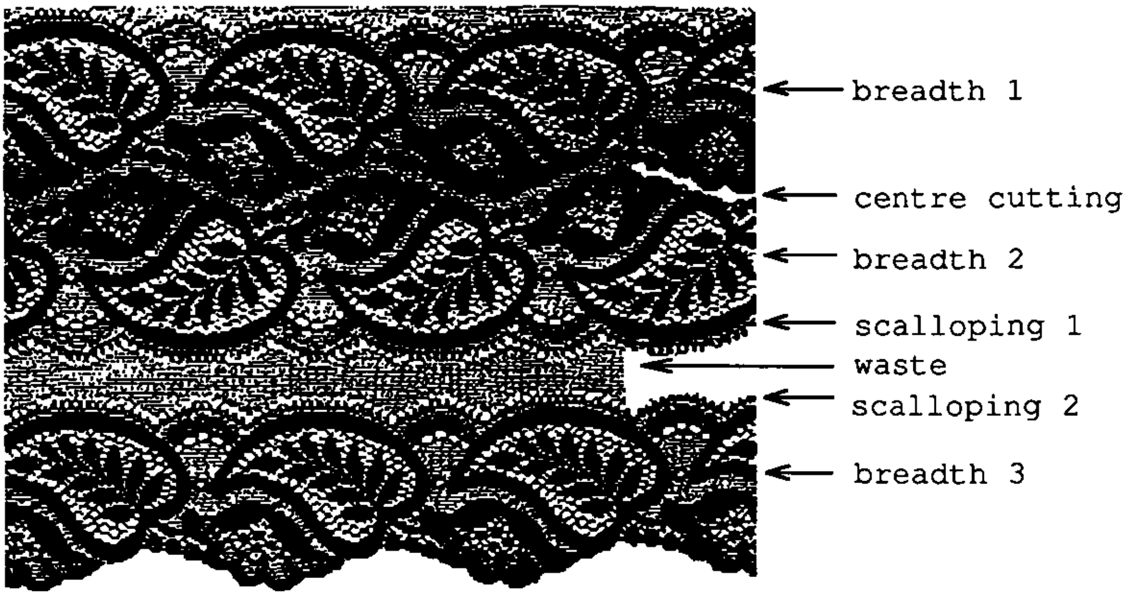


Figure 1.1 Lace separating positions

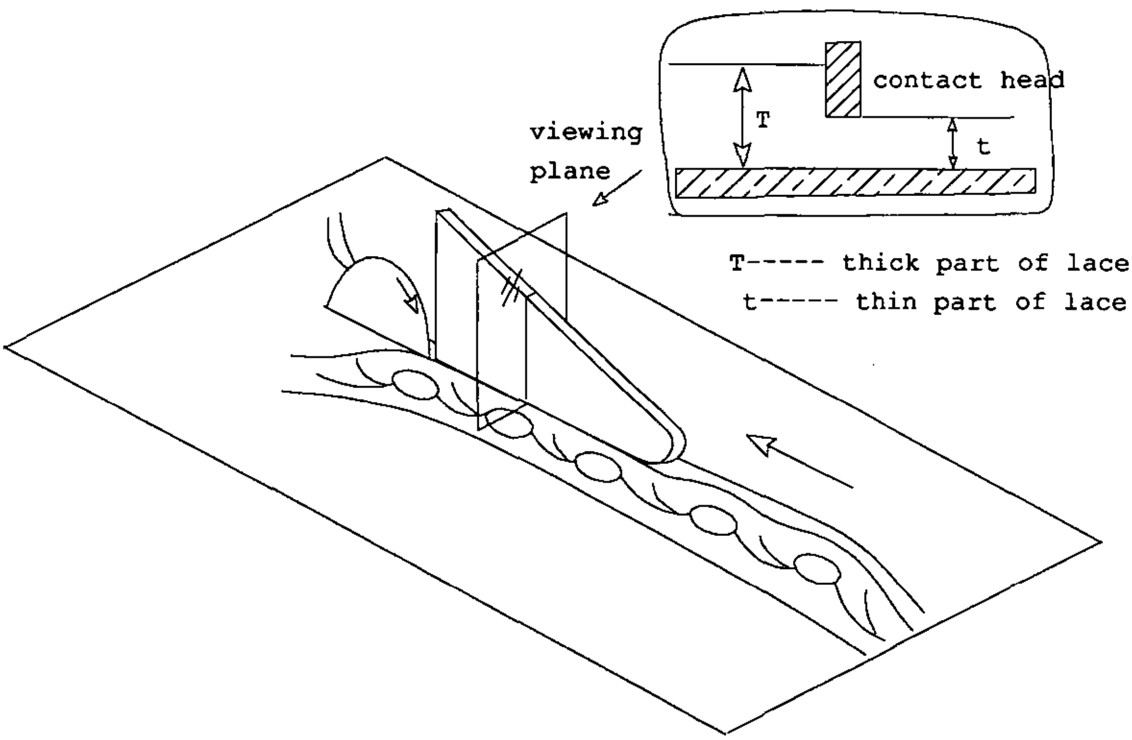


Figure 1.2 Schematic drawing of the mechanical guidance system

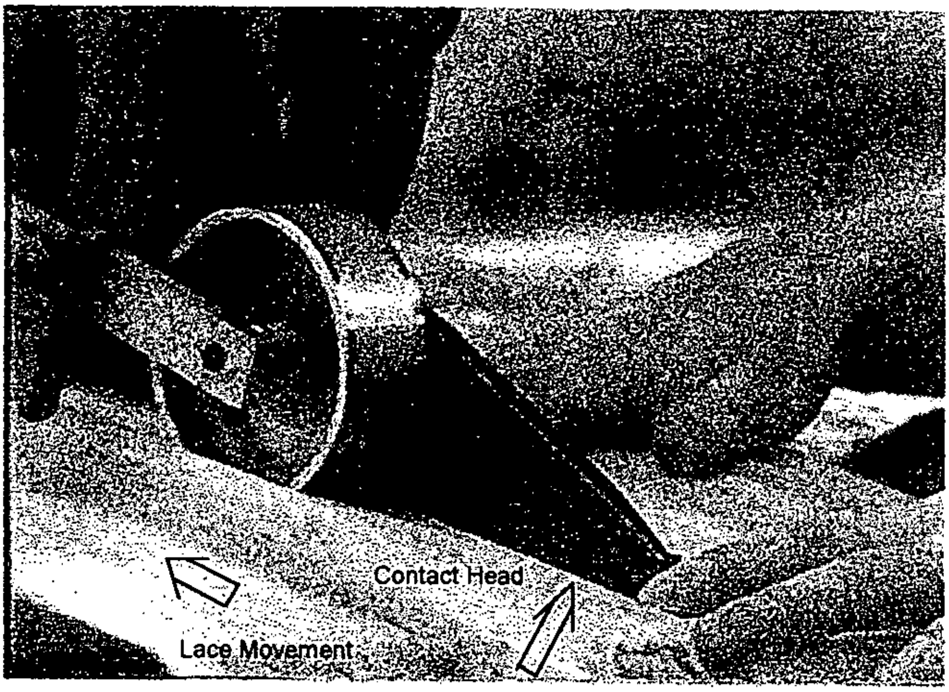


Figure 1.3 Operator assisted machine scalloping

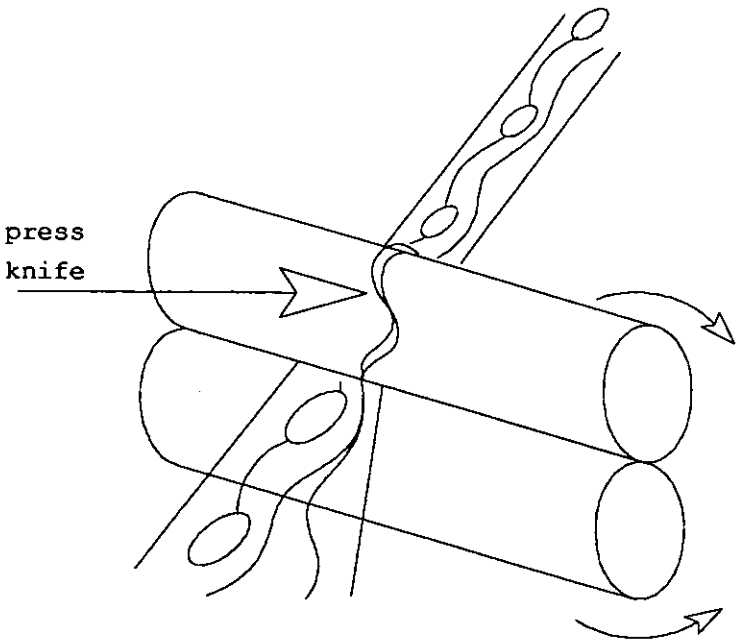


Figure 1.4 Rotary press knife approach

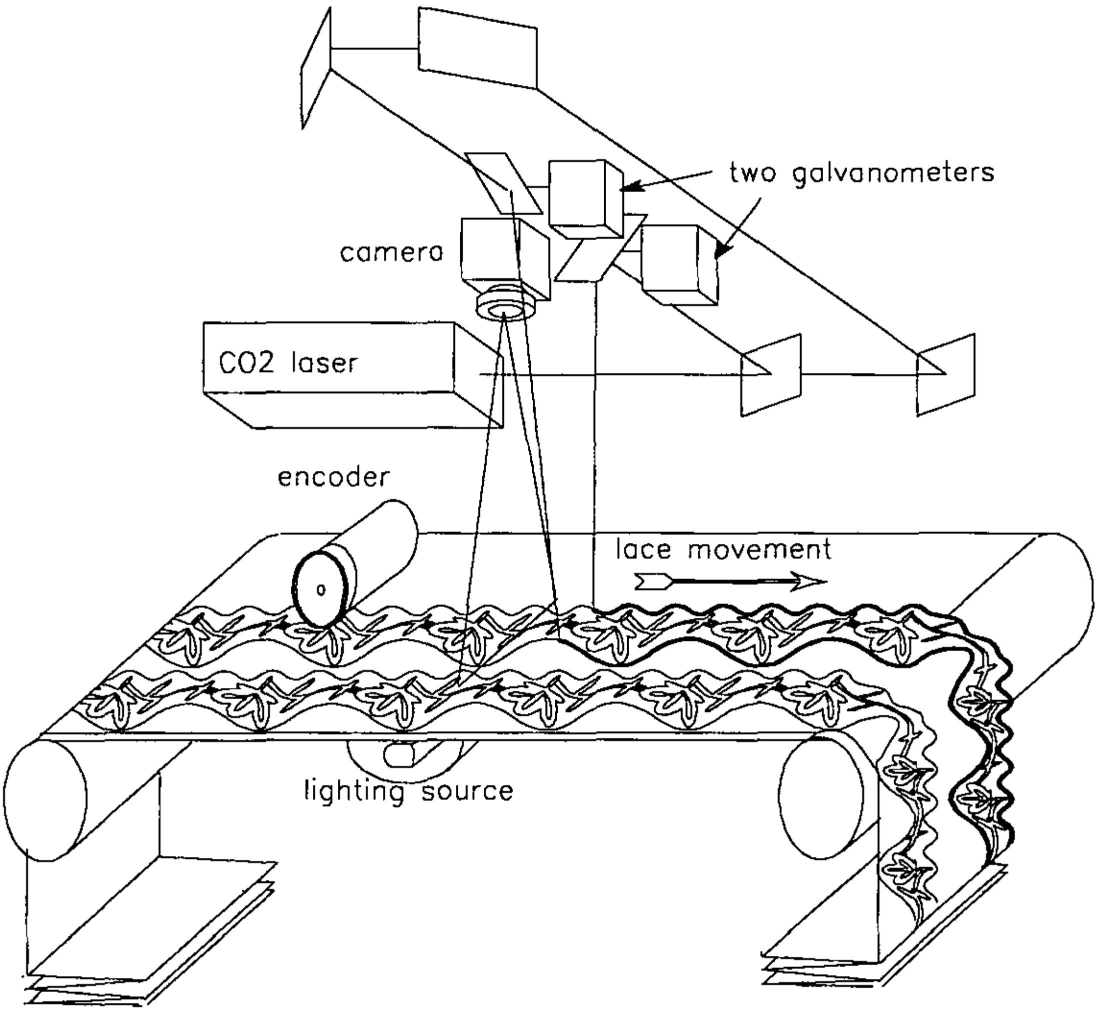


Figure 1.5 Blueprint of new lace separation machine

Chapter 2

REVIEW OF COMPUTER VISION

2.1 OVERVIEW

This chapter reviews the field of computer vision relevant to this thesis. It gives an introduction outlining the current state of computer vision. Then three major areas of research are reviewed. The first is about basic approaches to object location and scene analysis, the second related to applied work in the textile industry, and the last area details its applications in the lace industry.

2.2 COMPUTER VISION

Computer vision is one of the human being's attempts to mimic or improve upon his or her own performance. The performance of computer vision systems has been judged and will be continuously judged by comparing it with the one of the human visual system. So a short description of the human visual system is helpful to an understanding of the current state of typical computer vision systems.

2.2.1 Human Visual System

Over millions of years human beings have evolved five senses - vision, hearing, smell, taste and touch. It is undoubted that vision provides more external information than the other senses and human beings are used to processing this visual information very quickly without intellectual effort. Normally, a human can instantly pick out an object among other objects from a scene using a symbolic description such as the name of the object. This identification process of visual object recognition occurs very quickly, "with such apparently effortless alacrity that it is indeed difficult to realise the complexity of the processes involved and what an unbelievable visual processing ability human beings have" (Humphreys 1987). Yet such is the complexity of the processes involved that, even after more than a century of careful investigation, our knowledge of visual recognition and of how it is implemented in the brain remains all too incomplete. We are still some way from achieving the kind of detailed knowledge that would, for instance, allow a machine to see and recognise objects in the manner that humans do (Humphreys 1987).

What is certainly true is that eyes and brain form the major part of the human visual system and that visual object recognition is done mainly inside this system. The eyes control and provide the focused image on the inside wall or 'retina' of the eye, where the light energy is transferred into coded electrical energy. The brain processes the coded electrical energy to produce symbolic signals such as; what the object is, where and what size it is. The data received by eyes may exceed 10 megabits per second if a computer term is used (Davies 1990). The coded output signal associated with a particular optical pattern is a sequence of frequency-modulated pulses. Many coded signals are sent to the brain simultaneously to be processed (Stubbs 1979). Inside the brain there are some areas associated with vision (Levine 1985) but there is no evidence to show that the rest of the brain has no link to the visual object recognition process. Based on current knowledge the human visual system does not appear uniform in resolution density nor does it perceive light intensity absolutely. Rather it is an adaptive system that compensates for optical anomalies by processing the raw optic nerve signal to produce a uniform density, regular light intensity image (Gibbons 1992). We do not know how these compensations take place. Common sense suggests that some of these compensations have been done subconsciously. If you stand at a road and stare at a passing car, when the car moves far away and becomes smaller and smaller, your eyes will automatically focus on this car.

If one looks at the human visual system and uses computer terms to describe it, one of the ways in which the human visual system gains over the machine vision system is that the brain appears to work like an immense parallel computer with perhaps 10^{10} processing units (Davies 1990), with many inter-connections among them. The parallelism of the brain makes its vision process very fast and more sophisticated than a computer system even though a single unit in the brain works far slower than a single processor in a computer. Clearly, the architecture of data processing in the human visual system is better than any digital one currently realisable and the whole human visual system is not only hard-wired by millions of years of evolution but also programmable by training and active use. There are still many unsolved puzzles about the human visual system from a computer vision point of view. For example, how many processing levels in the human visual system are used to recognise an object and what kind of functions are there for each level. In fact, today's computer technology is far below the human visual system's level. There is no chance of building a computer vision system with all the processing abilities of the human visual system using a personal computer with 10^{10} processing units in the near future and it is therefore not surprising that a computer vision system cannot yet do most of the complex tasks carried out by the human visual system.

2.2.2 Ambiguities in Visual Object Recognition

Even the human visual system is not always perfect for recognising an object from visual information. One example is the *impossible triangle* (Figure 2.1) where the human visual system makes 2D projections of 3D objects thus leading to confusion. Another example is the *vase-face figure* (Figure 2.2) that can be recognised as either two profiles of faces or the silhouette of a vase depending on what you want to see. In the real world, camouflage serves precisely to make it difficult to separate an object such as a chameleon from its background.

All of these illustrate clearly that visual object recognition by computer vision with the same performance as, or better than, the human visual system, must be one of the most difficult problems for researchers to tackle.

2.2.3 Development of Computer Vision Systems

Over the last three decades, many efforts have been put into building machine vision systems to be used in some specific cases such as dangerous working areas, and full inspection for food products. According to the image signal processing methods, machine vision systems can be roughly divided into digital machine vision systems and optical machine vision systems. A basic digital machine vision system includes :

- A photoelectric sensor that transfers an optical image to a coded electrical signal array;

- An analogue to digital converter that provides an interface between a digital image signal processing unit and an image sensor;
- A digital image signal processing unit that can be either a computer or a dedicated hardware circuit board which extracts the desired information from the input image for the purpose of further processing.

If a computer is used as a digital image signal processing unit, image processing software used inside the computer should be included in the basic machine vision system. Recently the optical coherence of the laser beam and the advantages of high-speed signal processing based on the laser-optical development have encouraged researchers to bring laser optical and digital techniques together to build a hybrid machine-vision system (Kamemaru 1992, Mao 1992). The hybrid machine-vision system will be an attractive one in which the high processing speed of the optical elements and easy use of the digital part work together to obtain the best performance. Due to the physical bulk of the optical system, its sensitivity to noise, and the technique being mainly laboratory based, the optical part is more difficult to use than the digital part, hence very few systems have been used in the real world. Here only digital vision systems are discussed so the term 'machine-vision' will generally refer to digital machine vision. In most cases a computer will be used as an image processing unit or a host supplying a human machine interface to talk to a dedicated hardware board. So the term 'computer vision' will be used to refer to machine vision.

Although the digital machine vision is the most mature system among man made vision systems, no one will doubt that it is only an infant when compared to the human visual system. In the next section the problems and advantages associated with the computer vision system will be discussed.

2.2.4 Computer Vision Versus Human Vision

The basic limitations and advantages of computer vision systems, based on current technology should be taken into account when a vision system is going to be used to solve a real world problem.

Main limitations of the computer vision system:

1. The highest number of pixels currently available for an area scan type camera is 4096×4096, and for a line scan camera is 12000 (Fairchild 1990). The number of brighter photo sensors, or cones, in a human eye is about 10 million, and the number of dark photo sensors, or rods, is about 200 million (Gregory 1990). The image quality of most computer vision systems is poor and discrete when compared to the human visual system.
2. A computer has limited processing and reasoning capacity compared with 10^{10} processing units in the brain.
3. There is no clear model that describes how the human visual system works so far. One computer vision system may work well in a special case but no computer vision system can work well in various cases like human visual systems.

Advantages of the computer vision system:

1. Human operatives are subject to fatigue and boredom which of course do not apply to a machine vision system which may be put to productive work for 24 hours a day.
2. The sensory range of wavelengths of a computer vision systems can be bigger than that of the human eyes.
3. Processing speed of a single CPU is higher than a single brain cell so that Computer vision may be able to do simpler jobs faster.

These key points are the basis of research into computer vision. Most of the research effort is directed to achieve vision tasks by using available technology.

2.3 Basic Approaches to Object Location and Scene Analysis

Object location and scene analysis are two basic topics of computer vision. They discuss how to extract the information about the object and its surrounding environment from images and lead to further manipulation of the object.

2.3.1 Object Location

Object location refers to finding a desired object, or objects, in a given image, and describing their position. There are several ways to locate an object.

One way of achieving this is to move a suitable "template" of size $n \times n$ over the whole image, of size $N \times N$, and to find where a match occurs. This is called correlation matching. A match can be defined as a position where there is exact agreement between the template and the local portion of the image but in most real cases a best match will be pursued if the exact agreement cannot be achieved. The template defined as an object contains intensity values of pixels of size $n \times n$, and relating to the features of the defined object and geometric relationships between the features. This approach is most suited to the case in which the image can be acquired under almost identical viewing conditions as the "template" and the geometric relationships between the features do not change. The advantage of this approach is that it can be implemented in hardware to achieve high processing rates. In an automatic wafer inspection system (Yoda 1988) defects on memory devices such as 1Mbit DRAM's and 256 Kbit SRAM's were detected by using auto-correlation, pixel-to-pixel comparison, and morphology. Inside memory devices memory cells are laid out repetitively and some of them are mirrored. A 1024 pixel CCD linear sensor was run at 7 MHz data rate, and the video signal from the CCD was converted to an 8 bit digital signal. The inspecting device was laid on an X-Y moving stage controlled by a computer to keep the position correct when it was being scanned. Real-time correlation circuits matched a memory cell against a previous cell then pixel-to-pixel comparison found the difference between two cells that could be defects in the cells. The inspection speed of the system was 30 times faster than humans. In this case the viewing conditions can be controlled to be kept almost constant during the whole process. Image processing was done by dedicated hardware.

Object location can be achieved by "segmenting" it from its background if either object or background has a high level of uniformity and high level of contrast in some parameters such as brightness, colour or texture. Usually this is called region based segmentation. This method is more suitable in the case when the shape of an object can change but some parameters remain fixed. The segmented portion of an image according to the parameters will be the desired object or part of a desired object for further processing. A vision system for autonomous land vehicle navigation has used colour to segment the road from background in the outdoor case (Turk 1988). A high level reasoning subsystem is used to cope with some occasional segment failures in the vision system caused by changes in the weather on the test track.

If there are several objects with different shapes but the same brightness, colour or texture, the region based segmentation method cannot be used to segment a desired object from others directly. One of the methods used in this case is to locate a desired object by using its edge information. There are several ways to detect the edges of objects in an image (Canny 1986, Lyvers 1988). In order to inspect all leads of the high volume surface mount device (SMD) resistor packages, an automatic inspection station was formed by a mechanical delivery system and a machine vision system (Chapman 1990). The SMDs were back-lit to present a profile of the leads to a camera. A ledge algorithm returned the position, magnitude and direction of all the edges between two predefined positions in an image. A program found the tips of all the leads of a SMD. Then, the tip information was used to judge the quality of a SMD, according to user defined specifications, and to signal the mechanical system to deliver the parts to good or bad shipping tubes. The inspection cycle time was less than 140ms.

Another approach to object location is to use the "Hough transform". The Hough transform was originally invented (Hough 1966) to detect straight lines in an image. In 1972, Duda and Hart developed a Hough transform method to locate circles (Duda 1972). In 1981, Ballond developed the Hough transform to the Generalised Hough transform (GHT) to locate an arbitrary shaped object (Ballard 1981). The basic principle of the Hough transform is to reconstruct the image in a parameter space and every pixel in the image will provide its own vote to the parameter space. Then the parameter space is searched for peaks that correspond to either centres of objects or angle and a distance of a line from an origin. This method has been successfully used to locate the centres of round biscuits (Davies 1984). It is reported that the method is highly robust, in that if part of the object is obscured or has disappeared, the object centre is still located accurately. In order to speed up the processing only the edge pixels on the image are used and it took 2-3 seconds to perform within a 128×128 image on a DEC PDP-11/73. Later a similar method was used to achieve eleven rectangular biscuits per second inspection rate by using bit-slice processors (Edmonds 1991).

2.3.2 Scene Analysis

The ultimate purpose of a vision system is to provide the information that allows an organism or man-made mechanism to interact with its surrounding environment in order to achieve some set of goals. For example walking around in a building to find an office, or moving the computer mouse to point to a desired point on a screen. Scenes contain several objects, and their geometry interrelationships. In scene analysis relative positions among the objects are at least as important as the objects themselves. A typical environmental model for scene analysis usually includes what objects are and what the interrelationships among them are in terms of space and time. There is no general answer as to how to establish an environmental model and the model will vary from case to case. On the basis of the model and an image or sequence of images a machine vision system will be able to answer questions such as where a desired object is and, by combination with some kind of space inference mechanism, to provide direction information for the desired move, for instance, to guide a robot.

Several databases and inference processes formed a terrain model for an autonomous land vehicle (Lawton 1986) to move in the outdoor world. There are three databases, short term memory, long term memory, and generic models. The short term memory is used to record dynamically the incoming image from sensors, middle image results of image processing, such as curves, regions, and surfaces, and hypotheses about objects and terrain in the world. The long term memory stores a priori terrain representation (concerning elevation and terrain type information, as well as knowledge of landmarks) and the hypotheses with enough associated evidence to be worth remembering. The generic models are generic schemes, the inheritance relations of the (model) schema network, and a set of image structure, grouping processes and rules for evaluating image structure interest. The five inference processes are perceptual inference, location inference, schema instantiation, long term memory/short term memory instantiation, and the task interface to deal with image analysis, to resolve ambiguities in the location of the vehicle, to define tasks for the perceptual processes, and to relate the image and knowledge base items.

2.4 COMPUTER VISION SYSTEMS IN THE TEXTILE INDUSTRY

To date, textile manufacturing has not adopted large scale automation. Labour is intensively involved at many stages of the manufacturing process, although there is evidence that there is now changing e.g. lace CAD/CAM up to the knitting stage. Few computer vision systems can be found.

The labour is there because the products are flexible even elastic material and difficult to manipulate and assemble automatically. In the textile industry the labour skills also well developed to deal with the products are very difficult to replace by techniques that have been developed in the other industries such as computer aided manufacturing. The existing computer vision techniques developed to deal with rigid or semi-rigid material face a new challenge from the textile world. With the development of computer vision technology and the rapid price decline of computer systems, more and more computer vision systems will find their way into the textile industry. In this section application of computer vision to the textile industry will be reviewed in inspection and automation.

2.4.1 Inspection

Inspection is the area to which most applications of computer vision systems belong. In (Karkanis 1989) a computer-vision based quality inspection system that can be applied to textile inspection was presented. A local smoothing of small areas and histogram classification were used to identify the defects on the roll of fabric. A monochrome CCD camera was used and the size of the digitised image was 256×256 pixels. The article did not mention the performance of the system. In (Norton-Wayne 1989) progress in a project to use computer vision to measure the dimensions of knitted garments was described. The image format is 512×512 pixels with 8 bits per pixel. The row by row scanning method is used to find essential nodes on a garment then the measurement is based on the node positions. Carpet texture measurement has provided another example of using computer vision (Wood 1989). Appearance retention is a key performance attribute of a carpet, frequently regarded as being more important than long-term durability. A localised intensity variation (LIV) technique is used based on a VAX image processing system (VIPS) to assess the appearance and texture of carpets. The VIPS system generates a 512×512 - pixel (maximum) 8-bit image. The system can have an important role in the objective measurement of carpet appearance. A surface inspection system (ERA 1993) employed CCD line scan cameras, DSP and transputers to detect the existence of anomalies, classify the anomalies and show them as faults. The inspection speed for complex woven fabrics was 1-2 metres per second.

2.4.2 Automation

Automation applications of vision systems in the textile industry are rare. A printed motif on household linen was detected by a computer vision system then cut (Ameziane 1985). Every motif is a group of connected pixels with similar grey level that is different from the grey level of the surrounding background and global grey level thresholding was used to separate motifs from the background. The contour of a motif was encoded by chain code then the centre of the contour was computed for the cutting tool. In (Khoury 1991) a PC-based vision system and a laser were used to cut upholstery fabrics. This was a semi-automatic cutting system. The vision system helps an operator to move a pattern template around the fabric and get the best pattern matching. On 1991 Hanover 11th International Exhibition of Textile Machinery only one machine used computer vision and a laser to cut emblems, embroidery, motifs, printed and woven labels. A 512×512 pixel camera was used to take an image of the label fabric. The cutting speed was 5cm per second. Because of the commercial confidentiality of the product there was no other information except a brochure (Hanover 1991).

2.5 COMPUTER VISION IN THE LACE INDUSTRY

Use of computer vision for lace work operations also falls into two major categories, computer vision inspection tasks, and computer vision for control, where the vision system is used for passing control parameters to an actuator. So far there is one inspection application (Skilton 1988, Norton-Wayne 1991) and two control applications (Russell 1989, Kimoto 1986) on lace. In (Russell 1989) a 256×64 DRAM typed sensor, and an Intel 8051 microprocessor were used to find a cutting position in a single breath of lace, then cut into a shorter length before it was sewn onto a garment. This was a low cost system and the operation time to find the cutting position was below 3 seconds. A CCD camera of 256×256 pixels and a laser were used to track and cut a curve on embroidered lace (Kimoto 1986). The principle of tracking was to find the boundaries of the black pixels along the x direction, then to locate the centre of the black pixels while the lace was fed in y direction (see Figure 2.3). The system was able to cut at 100mm/s. Obviously it could be a problem that cutting is required on a predetermined arbitrary position related to a more complicated pattern than a continuous curve.

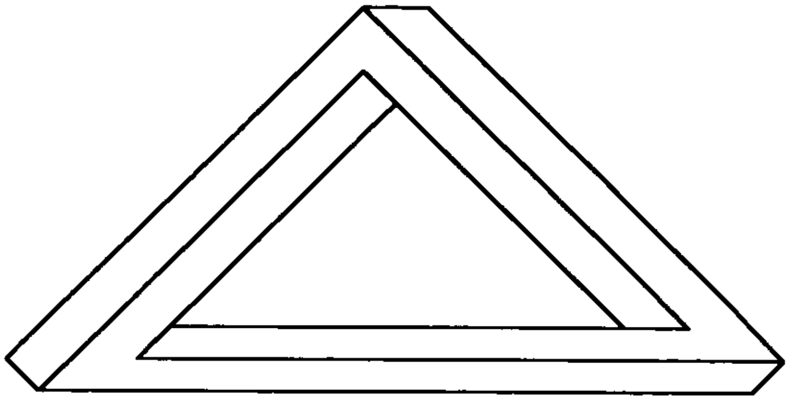


Figure 2.1 The impossible triangle

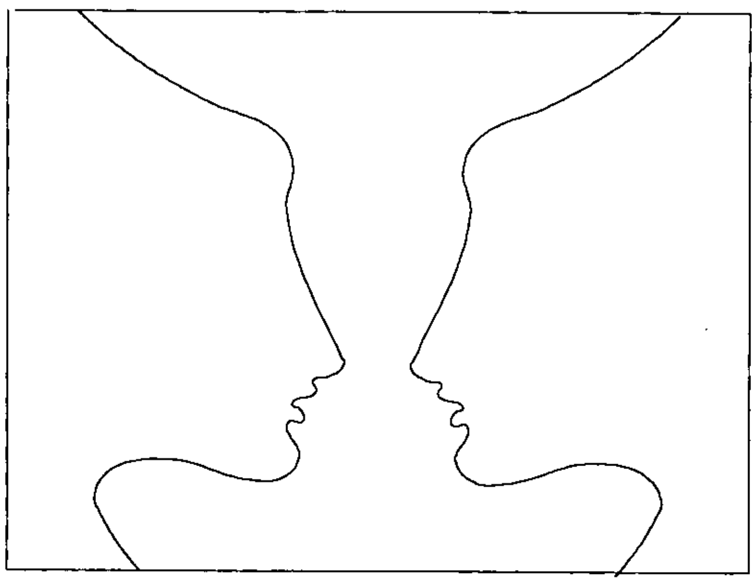


Figure 2.2 The vase-face

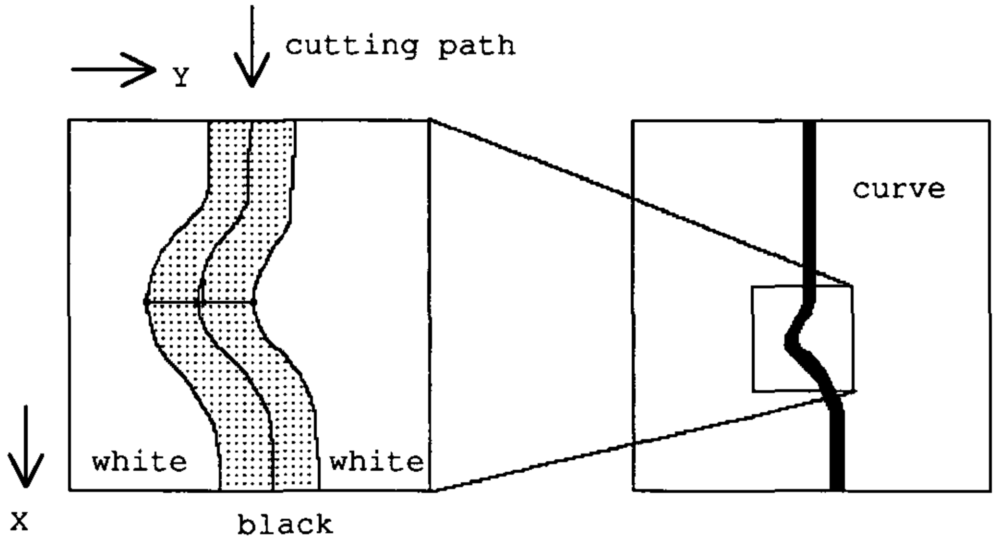


Figure 2.3 Tracking and cutting a lace with a continuous curve

Chapter 3

RESEARCH APPROACH, STRATEGY AND EXPERIMENTAL RIGS

3.1 INTRODUCTION

This chapter outlines the research approach, strategy, and the experimental rigs that have been developed for the task of vision guided lace separation. It covers the philosophy behind the choice of process structures and the design.

3.2 FUNDAMENTAL CHOICES OF THE VISION SYSTEM

The type of camera, vision processing hardware and technique used to track a moving lace web will determine the overall characteristics of a vision system, such as reliability, processing speed, development time. They are fundamental factors of the research approach, and are discussed in this section.

3.2.1 Area Scan Camera versus Line Scan Camera

Most vision systems on the market are based on area scan cameras, similar to TV cameras, producing the image in frames. This leads most commercial software to process image, which is from area scan cameras, frame by frame. In order to distinguish this kind vision system from others, it is called as a frame grabber vision system.

In an area scan camera every pixel has equal exposure time, or integration time, to experience light. This time can be controlled by a built-in shutter or/and a high speed strobe light to avoid blurring when the camera is used to film a moving object. But in a low-cost industrial camera this time equals the frame time. Therefore blurring is one of the problems of applying this camera to a moving object. In order to simplify the discussion, the blurring problem is ignored temporarily in the rest of this section.

Although the frame grabber vision system is an off-the-shelf answer to some applications such as surveillance, it cannot be used to find the cutting path on lace in real time because of the limitation of the frame speed and the unpredictable distortion of lace. The frame speed of most vision systems is 25-30 frames per second at which the display of the image seems continuous for the eyes. Typical values for computer image processing are 512×512 pixels for each frame and eight bits per pixel. If the image uses grey scale and the frame rate is 25 frames/s, at least 512×512×25 bytes data (for a colour image the data will be tripled) should be processed every second to produce the control information for lace separation. Computer capacity for these data is from 40MIPs (Million Instructions Per second) to 200MIPs if the average instruction count in processing each pixel is from 6 to 30. But the major drawback of frame grabber vision systems is that the time interval between the frames is too long to cope with any changes occurring in the lace during this period and so the results of the image processing are inaccurate and can not to be used for lace separation.

Assuming that the lace moves at 1m/s, 25 frames are taken per second and there is no overlap between adjacent frames, every frame covers a 40 mm length of the lace. The time interval between the frames is 40 ms . If the frame grabbing, image processing and cutting are carried out independently and pipe-lined (Figure 3.1), the shortest time interval between image grabbing and cutting is 80 ms and in the following 40 ms period the information for cutting will be drawn from the image without considering any subsequent change in the lace such as release of tension as it is cut. Any change taking less than 40 ms will be ignored. This is a dead reckoning method and might produce incorrect information for lace separation because the last separation information uses the image taken 120 ms ago and the portion of lace corresponding to the image will be changed. Furthermore, any discontinuity between two successive frames that is caused by positional drift or stretch of the lace will introduce new difficulties into the search for the cutting path on the image. One way to cope with change in the lace is to increase the frame rate and reduce the time interval between image grabbing and cutting. This might get more information from the lace change point of view, but more computing power is needed and the computing effort put into the overlapping parts of the image will be wasted. This option cannot be pursued since practically, most frame grabber systems offer little chance for the end user to increase their frame rates.

Due to the flexibility of the lace it is important to monitor its geometric changes all the time. In order to monitor the change closely and use the computing power efficiently, a computer vision system based on a line scan camera has been investigated. A line scan camera is a one dimensional image sensor and the image data can be updated every line. If the line scan rate is 5000 lines per second and other conditions are the same as above, the time interval between the image being taken and cutting will be 0.4 ms. Any change taking longer than 0.4 millisecond can be detected and acted, which is better than any frame grabber system from the close monitoring point of view. Another advantage of the vision system with a line scan camera is that the data rate for image processing is reduced because there is no overlapping data. There is, therefore, a good chance of reducing the requirements for computing capacity. However, typical algorithms based on frame grabbing are unsuitable for processing the image data from a line scan camera. If these algorithms are used without modification all the advantages of the line scan system disappear. Therefore, the development of an image processing algorithm to find a cutting path on lace based on a line-scan vision system was identified as a basic research task of this project.

3.2.2 Choice of Image Processing Hardware and Development of Image Processing Software

There are three different basic pixel formats for images: binary, grey level and colour. Generally speaking, the binary format has the lowest memory requirement and the highest processing speed. The colour format needs the biggest memory requirement and has the slowest processing speed except that if an object can be located by its colour then colour processing may sometimes be the fastest. Usually the desired cutting path on lace has no colour differentiation from other parts of the lace and so a colour segmentation approach cannot be used to find a cutting path. The colour pixel format is, therefore, not discussed in this thesis. A grey level image contains richer image information than a binary image but the processing speed will be slower if similar hardware and software are used or the cost of hardware (even software development) will be higher to keep the same speed.

Due to the binary visual character of lace a back-lit technique (section 3.4.5) can be applied to produce a clear binary image (section 6.3). An initial trial of the reference map (section 3.2.3) concept and its further development (chapter 4) proved that the binary image provided enough information to determine the cutting path. Therefore all subsequent processing is based on binary images.

The development of an image processing algorithm has been carried out in two phases: (1) with a general purpose computer using high-level-language in off-line mode and (2) DSP (Digital Signal Processor) implementation in on-line mode. In the first phase, the C language was used to develop the image processing algorithm. Because of the flexibility of C various possibilities for image processing algorithms were easily explored. After it had been fully tested the developed algorithm was implemented on DSP to run in real-time.

DSP microprocessors are faster devices which enjoy all the advantages of traditional microprocessors: they are easy to use, flexible, and economical. DSP microprocessors are characterised by fast multiply instructions, reduced instruction sets, and specialised instructions to make DSP based algorithms execute fast and efficiently.

The choice of a DSP solution was determined by the balance of requirements between the algorithm execution speed, development time, and development cost. In fact, an image algorithm can be implemented by either DSP, bit-slice elements, digital logic circuits or analogue circuits if there is no restriction on development time. Each solution will achieve different real-time image processing speeds. The DSP implementation is the slowest (Chassaing 1990) among bit-slice elements, and digital logic circuits. Optical implementation is the fastest. The development time and cost of an implementation method are approximately inverse to its execution speed. Most DSPs and their development boards are commercially available and some of them have different versions to be hosted by specific computers. The bit-slice element implementation needs hardware and software system design starting with the implementation code design based on the bit-slice elements. Only then can the algorithm be implemented on this kind of system. The developing time for digital logic circuits and analogue circuits will be even longer because every change in the algorithm will require changing the circuits. The DSP vision system can take advantage of any update of the DSP performance without major software change simply by upgrading the DSP board. Based on these considerations, DSP was chosen to implement the algorithm.

3.2.3 Finding a Cutting Path on Lace by a Reference Map

The lace separating process could be considered similar to that by which an autonomous "robot" might travel along a desired path in a terrain with a lot of landmarks that are formed by the altitude of the land. The "robot" only moves forwards with constant speed. One approach to the "robot" movement is that the path is defined by one of the transition positions of the land altitude or "edges", then the "robot" just follows the edges. In fact, the edges in a visual sense are similar to the transition between the thick and the thin parts, in a tactile sense, on lace. The "robot" is put at the start position along an edge and instructed as to the direction of movement. As soon as the "robot" moves, it only checks the edge and does not take any other land marks within the viewing range into account. If this approach is implemented in a vision system to guide a laser beam to scallop lace it is certainly better than the mechanical lace scalloping method since due to the non contact of the vision system there is no interference between finding the path and following it. But this approach cannot support the centre cutting because of similar reasons as with a mechanical contact head. Another approach to the "robot" movement could be that the "robot" has a map with a desired path. Before moving forwards the "robot" checks the map and its own local environment to find the corresponding point on map to the position where it is. The "robot" only checks the newest environment against a local area of map next to the last located position to find a current location. The map includes all necessary landmarks (in one repeated period since the lace 'landscape' is repetitive) and is arranged in the sequence of the landmarks appearing. This approach will treat thick threads along the path of the centre cutting the same as the other areas without thick treads and can, therefore, support centre cutting. This approach also gives operators a great deal of freedom to cut different profiles on the same lace by giving different guiding maps.

A real autonomous robot is looking at a 3-D world in which its movement will need not only image processing but also spatial reasoning such as scene analysis. The lace image is a 2-D world and the thickness of the lace can be ignored in comparison to the other two dimensions. Another advantage in applying computer vision to lace is that the variation of the work environment, such as lighting, can be separated from the variation of the movement of the lace (this is difficult outdoors when using natural lighting). So the scene analysis for a lace image will be much simpler than for real robotics movement and may not need a complex inference mechanism to achieve failure-free lace cutting.

In order to implement the map-guided approach the research of the computer vision focuses on:

1. Finding the start point;
2. Tracking along a desired path;
3. Maintaining stable working conditions for the vision system.

3.3 RESEARCH STRATEGY

The whole research follows a strategy, from broad to focused, and from low speed of image processing to high speed. At the same time it also tries to achieve the optimum result from the available resource such as computing power.

At the beginning of this project an IBM compatible PC and a commercial area-scan frame grabber system were used to capture an image from a static lace sample (test rig 1, see Figure 3.14). This was carried out in order to determine suitable resolution for the image system, to simulate using a line scan camera, and to evaluate the possibility of developing a tracking algorithm for lace cutting based on a binary image.

An incremental lace pattern tracking algorithm based on the line scan camera vision system has been developed (see chapter 4). The research effort focused on building this vision system includes:

1) An interface board (see Appendix 4) between a line scan camera and a DBV56 DSP board.
2) DSP development software on a host computer (see Figure 3.11).

After the tracking algorithm had been implemented in the DSP, research to find the start point (chapter 5) and maintain stable working conditions (chapter 6) was carried out in parallel. The work carried out to speed up the tracking process is detailed in chapter 7.

The experimental tests for the vision system with a line scan camera have been carried out on three rigs:

1) a frame based on a linear moving table that holds a lace sample and moves backwards and forwards. A computer driven galvanometer mirror points a He-Ne laser beam to the cutting position to mimic the whole procedure of tracking and cutting a desired path on patterned lace (see Figure 3.15);
2) an integration rig (see Figure 3.16) in which a 50w CO_2 laser beam is positioned coaxially with the He-Ne laser beam to cut lace and the lace can travel further and faster than in the previous test rig. This test rig pushes the DSP to its processing speed limit;
3) a prototype machine (Figure 3.17) in which a 240w CO_2 laser and a lace handling sub-system are used to test the higher speed and twin beam vision systems. The test was carried out both at LUT and one of our industrial collaboration.

The evolution of vision system and experimental rigs is shown in figure 3.2. The vision systems and experimental rigs have been developed in two parallel ways. The first vision system and rig formed the first stage of research. The second vision system has been combined with the second rig and the third rig to form the second and third stages of research. The third vision system and the third rig form the fourth research stage. The fourth vision system and the prototype form the last research stage.

3.4 RESEARCH CONSIDERATIONS

In this section some aspects that cannot be presented in an individual chapter later, but which are important to the whole research project are discussed. These are: image resolution, speed, optical arrangement, host computer and its operating system, light source arrangement and making a reference map for tracking.

These different aspects provide an insight into the formulation of the objectives (section 3.5) containing the different stages of the research strategy.

3.4.1 Image Resolution

The spatial resolution of a vision system can be defined in terms of its number of pixels per mm. This is one of the key parameters that determine the performance of a vision system. Higher resolution means richer and more accurate local information contained in the image, but usually needs more computing power to process the image. If the number of pixels in every line from the camera is fixed, higher resolution will mean less global information. Usually the more global information taken into account, the lower risk of getting lost when one travels. If one wants to know exactly where he or she stands, more local information is needed. This is similar to the vision system. The resolution of the vision system should balance between the global and local information.

The resolution of an image from a camera is determined by its optical layout, the geometrical size of the photo detectors on the camera, and the A/D converter sampling rate. The optical layout that can make the image on the focused plane bigger or smaller will be discussed later. The photo detectors in a camera will transfer an optical image continuous in both time and spatial aspects, into an electrical image discrete in both time and spatial aspects. The discreteness in time will affect processing speed and is discussed in section 3.4.2. The spatial discreteness will affect the resolution of the image. Figure 3.3 shows how the fine information is lost when the image is transferred from optical to electronic form. Figure 3.4 shows that a single bar whose image is as large as a single photo detector has two different outputs when it overlays a single photo detector or two adjacent photo detectors. If a threshold method is used to detect the single bar, the detected result will depend on what threshold level is set. If the threshold level is set to see both cases, the width detected on the right output will be as twice big as on the left output. The detection error will be one pixel. The sampling rate of A/D converter will also affect the resolution of image. If the sampling rate is half of the pixel rate of the camera, the resolution of the image after A/D conversion will be only half of the resolution before the conversion. But if the conversion rate rises above the pixel rate, the image detail lost (as Figure 3.3) in the photo detectors will not be recovered. What kind of resolution should be used is dependent on the imaging task. During the first stage of the research, various resolutions of lace sample were investigated to find a range of resolutions with which the finest landmarks (purls) can be sampled on any design without any doubt. Figure 3.5 shows three images with different resolutions taken from the same lace sample. Resolutions higher than 10 pixels per mm are able to image the fine detail, but this kind of detail could slow the image processing down when a cutting path is being searched for. Resolutions lower than 2.3 pixels per mm will lose some information useful for finding a cutting path. Resolutions between 3 and 4 pixels/mm are suitable for finding a cutting path.

3.4.2 Speed

The target system speed of this research project is about lm per second, which is faster than the existing systems in the textile industry. The required processing power of the vision system corresponding to 1 m/s lace movement is proportional to the vision resolution. The highest required longitudinal resolution of the vision system is around 10 pixels/mm so the vision system should be able to process 10,000 lines of image per second. The camera and the interface between the camera and DSP board were chosen and designed to provide 10,000 lines/sec at an early stage of the research. The camera chosen is a Fairchild 2048 pixel device and runs at up to 20MHz pixel rate. If the camera runs at maximum speed, the number of lines of image from the camera is around 10,000 per second. As mentioned before, the camera also divides a continuous optical image into a discrete electronic image in time co-ordinates. All photons on a single photo detector during its integrating period are converted into electrons provided 100% conversion efficiency is achieved. If the object does not move, changing the camera integration period only changes the absolute electron number on every photo detector. It does not change relative electron numbers that represent the geometry relationship between the image elements on the electronic image, except that some photo detectors may become saturated because they experience too many photons.

In figure 3.6 the absolute image levels on each photo detector are reduced while the integration period is reduced, but the distances between two left sides of two bars are the same, a'=a. If the object moves at a constant speed, and perpendicular to the CCD axis, different camera integrating periods will make different lengths of object pass under the camera (see Figure 3.7) during each integrating period, creating a 'blur effect'. The shapes of electron images are different. The shorter the integrating period of camera, the clearer the image.

The next speed related area is an interface for the line scan camera and the DSP. The choice was to buy one or build one. There was only one commercial interface board for this camera hosted on a PC at that time. On this board an output port for image acquisition is connected to the industry standard architecture (ISA) bus. The image from the line scan camera can pass to the host PC or another DSP board hosted on the same computer via the ISA bus. From the speed point of view, unfortunately this board would slow down the vision processing speed because the camera data and signal between PC and an embedded vision system will share the same PC bus. In order to separate the camera data channel from the data channel between a host computer and an embedded vision system, an interface board between the camera and vision system has been designed and built (see Appendix 4). This engineering work was an essential part of the research activity maintaining a constant exposure time for the camera (described in chapter 6).

The principle of the interface board (see Appendix 4) is that every pixel of video output from the line scan camera is converted into 1 bit digital (binary) signal and that the binary image is then sent to DSP for further processing. On the interface board there are two function blocks, analogue block and digital block. A 12 bit (only using the most significant 8 bits, other bits set to zero) D/A converter controlled by DSP sets the threshold level of an analogue comparator. A video signal from the camera is sent to comparator after a video amplifier. A binary image is produced by the comparator. Then the binary image is sent to DSP as 24 bit words which are the width of DSP input port. The interface also has two other connectors, one for a synchronising board (see chapter 6), one for a hardware correlator board (see chapter 7).

The criteria used in the selection of a processing unit was that it must be a development system, must have enough power to carry out the research at stage two and three, and must have separate data channels for both the camera and host computer. Choice of processing unit is discussed in section 3.2.2. The choice of fixed-point DSP or floating-point DSP for use with the vision system is decided after the tracking method has been successfully tested on the PC (see chapter 4). Most operations for tracking are logical and bitwise operations for which most fixed-point DSPs are adequate. The selection process indicated that the DBV56 DSP development system as the most suitable option.

The camera is not the limiting factor on speed (see beginning of this section). It is likely to be the image processing algorithm and or hardware.

The DBV56 board (Data β 1991) contains four Motorola DSP56001 microprocessors which operate at 27MHz, 64K word (24bit) global on board data memory can be accessed by all processors. The DSP global memory can also be accessed by host computer via the VMEbus. Each processor has 64k words (24bit) of private data memory divided into X and Y address spaces, and 32k words (24bit) of program RAM. The host port of each DSP processor can be accessed by the host computer via a short address of the VMEbus. One DSP is used to control the camera, to decide the right threshold level for the A/D converter, and to capture the image data from the line scan camera for further processing. Another DSP is used to implement the incremental tracking algorithm. The remaining DSPs on the board are in idle state. When twin beam cutting is implemented, one more DSP is used for extra tracking.

It was evident from the beginning of the project that the vision system should work at *real-time* speed. Therefore the strategy for algorithm development requires:

(1) a robust method to find the cutting path;

(2) the key part of the algorithm able to be implemented in hardware.

This was achieved by first looking for a robust method to find a cutting path using a general purpose computer and secondly transferring this method to a special purpose computer such as a DSP to implement a real cutting system at a medium speed and then reviewing the whole problem again. If the tracking algorithm implemented on DSP processor was not fast enough, a dedicated hardware or a hybrid implementation of the tracking algorithm to speed up the whole process would be investigated.

3.4.3 Optical Arrangement

Although the Fairchild camera can take a standard 35mm format camera lens, the view angle of the camera will be smaller than on the standard 35mm camera because the width of the CCD chip is 26mm (Fairchild 1990) which is shorter than the 50 mm diagonal length of 35mm film (Figure 3.8). If both CCD chip and 35mm film are on the same focal plane and the same lens is used for the two cameras, the CCD camera needs nearly double the object distance as the 35mm camera to image the same area (Figure 3.9). In other words, the 50mm focus lens is a 'standard' lens for 35mm camera but is a 'telephoto' lens for the Fairchild CCD camera. Resulting in an object-image distance of 200mm object view length with 3 or 4 pixels/mm resolution is approximate 1000mm. This 'telephoto' effect may be a problem for applications in which object-image space is restricted. We can take advantage of the 'telephoto' effect to reduce the wide-angle distortion that mostly exists near the edge of the image, and makes objects on the edge of the film larger than the same sized objects in the middle of the film (Reynolds 1984). However, whilst space is not restricted to any great extent on this vision application, an object-image distance of 1000mm with a 50mm focal length lens was considered too large, a wide angle lens (F=28mm) is used to keep the distance between the lace and lens of the camera in a range around 500 to 700mm. The object-lens distance will be determined by equation 3.1.

$$d = (l_f \times w_o) / (n_p \times w_p) \qquad \text{eq. 3.1}$$

Where d stands for the distance between an object and the camera lens, l_f is the focal length, w_o the viewing width of the object, n_p the number of pixels to be used on the CCD chip, and w_p the width of each pixel.

3.4.4 Host Computer and Its Operating System

Obviously the host computer is not only used for the vision system but also for other control tasks such as mirror control and motor control. A multi-tasking real-time operating system is chosen for this purpose. The DBV56 system needs to interface to a VMEbus. The Syntel LC850 68020 20MHz based computer system is, therefore, an appropriate host computer to control the DSP board. It runs the OS9 real-time multi-user and multi-tasking operating system. Some DSP development software on the Syntel LC850 has been developed. The basic functions of the DSP development software are:

- Send a program or data from host to any DSP
- Get a program or data from any DSP to host
- Monitor the execution of a program on the DSP
- Display the image in the DSP memory
- Modify the data, program, or registers of any DSP.

3.4.5 Light Source Arrangement

There are two possible lighting techniques: front lit and back lit. Which arrangement should be used depends on the optical property of the object to be viewed and what kind of information is needed. For example, the front lighting technique is more likely to be used to view the surface of a solid object (Batchelor 1985). Back lighting will give a silhouette of the object. The thin part of the lace is more like a transparent material and the thick part of the lace is more like a solid material. The back lighting will give a clear contrast between the thin and thick parts of the lace. Although a lot of surface information of the lace will be lost by using the back lighting technique, this data is not a requirement for lace pattern tracking.

3.4.6 Making a Reference Map

At section 3.2.3, a map guided following method is discussed. There are two ways to make such a reference map. The first one is to use the CAD information which describes the lace geometry. This approach will require work to transform the CAD data to the right scale and to make the data look like the image from the camera. The second approach is to use the same camera system to digitise the lace pattern repeat that is used for capturing the lace image during pattern tracking. The advantage of the latter approach is that this method can be used for lace without CAD data. A cutting path can be defined on the map by input to the computer from the operator using the cursor keys or a mouse.

3.5 RESEARCH OBJECTIVES OF EACH STAGE

As mentioned in section 3.3 the whole research into the vision system was carried out via five stages. In order to investigate the problems of the vision system step by step, each research stage focused on different objectives. As the research project proceeded, the work focused more deeply on the problem.

3.5.1 Objectives for the First Stage

For the first research stage, the objectives are to find the correct resolution of the CCD camera for lace, to search for a robust method to find a cutting path on lace and to try different ways to find a starting point when lace is placed flat on a table. This will be achieved by using general purpose computer, area scan camera, frame grabber camera interface, and a test rig. Most commercial image processing software provides a link library to develop user algorithms This cannot be used to develop a line scan based algorithm for DSP code or hardware. Efforts were, therefore, made to simulate a line scan camera and develop a cutting path following algorithm.

3.5.2 Objectives for the Second Stage

In the second stage, the research objectives were to transfer the path following algorithm developed to the DSP assembler, to find a start point by using the following algorithm, and to keep the camera on a constant exposure time. Although the camera, the DSP board, and the computer were bought-in, they were not a complete development system (as mentioned before there was no such development system available on the market at that time). A lot of effort has been put in to establish a development system for the vision system on both software and hardware.

3.5.3 Objectives for the Third Stage

In the third stage, the main research objective was to find out the extent of any interference between cutting and finding the cutting path, by using a CO_2 laser to cut the lace.

3.5.4 Objectives for the Fourth Stage

In this stage, the research was focused on how to make whole system work faster by using hardware correlators.

3.5.5 Objectives for the Fifth Stage

In this stage, the research was focused on how to configure whole system to work as a fast-single-beam version by using hardware correlators and as a slower twin-beam version by using multiple DSPs. The aim was also to investigate any problems encountered when the lace moves faster which were not apparent at the lower speed, and to study the effect of twin path cutting on the performance of the tracking system.

3.6 VISION SYSTEM

In figure 3.2 the development of the vision systems is related to different phases of development of the experimental rigs. It is, therefore, better to describe the development of the vision systems and the experimental rigs in different sections. In this section, the hardware and software for the different vision systems is presented.

3.6.1 Vision System One

The hardware of the first vision system was based on commercial products. The vision system comprised a JVC colour CCD area scan camera, a MicroEye frame grabber board, a monitor and a general purpose personal computer with a 386 processor running at 20MHz. The camera can output 25 frames per second in full colour but the frame grabber board needs nearly 2 minutes to grab and convert a frame of video image to digital image. The monitor is used to help to adjust (focus) the lens of the camera. The bit map, in one of the colour image file formats, was easily converted to three simple monochrome images for further processing. A program written in 'C' was used to develop the tracking algorithm.

3.6.2 Vision System Two

The second vision system is formed by a Fairchild 2048 20MHz line scan camera, a Data β DBV56 DSP board, and a Motorola 68020 20MHz computer. An interface between the camera and DSP board has been built by the author to use one of the DSP processors to control the camera, convert the video data to digital form, and put it into global memory. Another DSP processor implements the tracking algorithm by using the image data from the global memory. The host computer controls the DSP board and a digital to analogue converter that drives a galvanometer to control a laser beam (Figure 3.10). A DSP code development system on the host computer has been built by the author to monitor and debug DSP code and to send data or code to the DSP from a disk file on the host computer or get data or code back from the DSP to disk file (Figure 3.11).

3.6.3 Vision System Three

The hardware arrangement of vision system three is similar to the second vision system except that a hardware correlator board is used to accelerate the tracking (Figure 3.12).

3.6.4 Vision System Four

This system combines twin tracking by the DSPs and single tracking by a hardware correlator board (Figure 3.13).

3.7 EXPERIMENTAL RIGS

The experimental rigs have evolved from merely keeping lace stable to actually cutting lace to suit the different research objectives.

3.7.1 Rig One

A lighting box and a frame form the first rig. The frame can hold a lace sample, with or without stretching and/or skewing, on top of the lighting box (Figure 3.14). Inside the lighting box four 20W fluorescent tubes driven by 240V AC provide a back-lit light source for the camera.

3.7.2 Rig Two

A "U" shaped frame (Figure 3.15) holding lace between the line scan camera and the light source can be moved on a pair of slideways driven manually or by a computer controlled DC motor. A mirror mounted on the shaft of a galvanometer reflects a He-Ne laser beam onto the lace to mimic the cutting process and make visual judgement of the system performance possible. An optical encoder sends a signal of the movement of the lace to the interface board between the camera and DSP board to synchronise the vision system with the movement of the lace. The length of the frame movement is about 600mm.

3.7.3 Rig Three

An integration rig is formed by an "E" shaped frame, a vision system, a cutting system based on a CO_2 laser, and a transportation system (Figure 3.16). A sample of lace is joined to form a belt which is driven by one friction roller downstream of the vision/cutting planes. Lace can travel at 1m per second and the length of track is about 4 m.

3.7.4 Prototype

This rig (Figure 3.17) is formed by a removable lace handling sub-system, 240w CO2 laser, and vision system four. The length of lace loop can be as long as 100m. This uses edge sensors to track the lace within ±5mm for presentation to the vision and cutting planes. The vision and cutting planes are separated by 20mm.

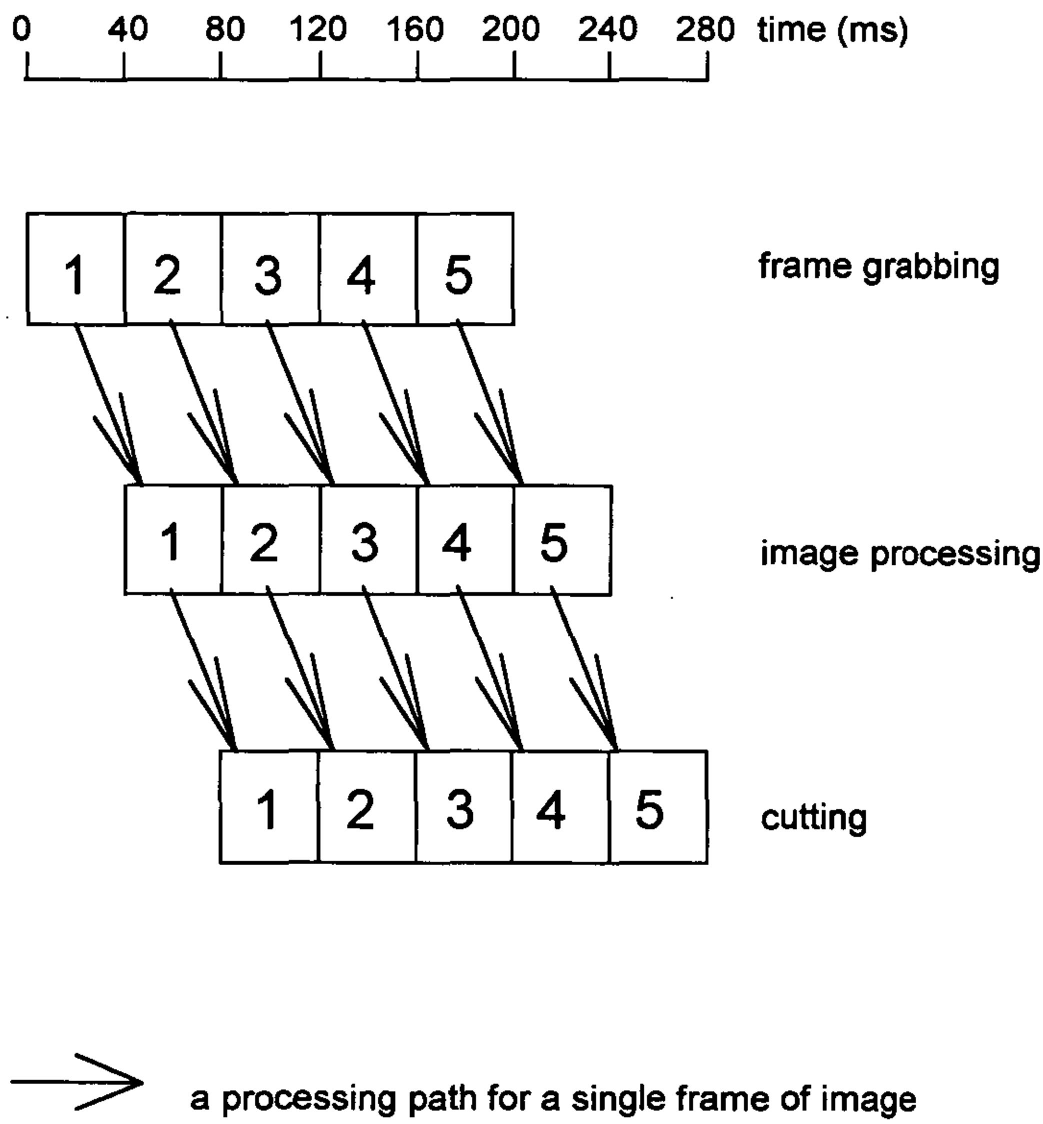


Figure 3.1 Timing for a frame grabber vision system

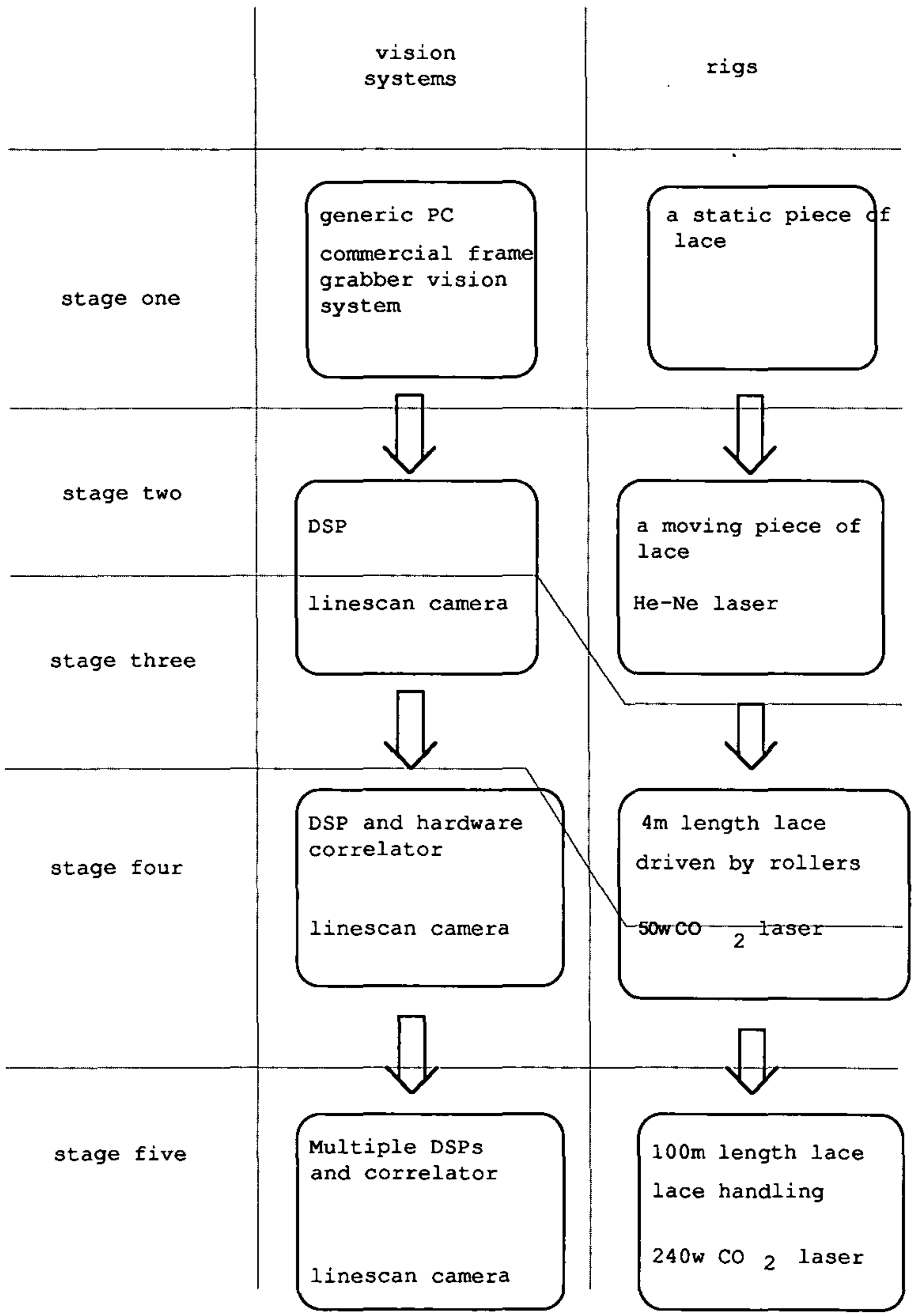


Figure 3.2 Research evolution

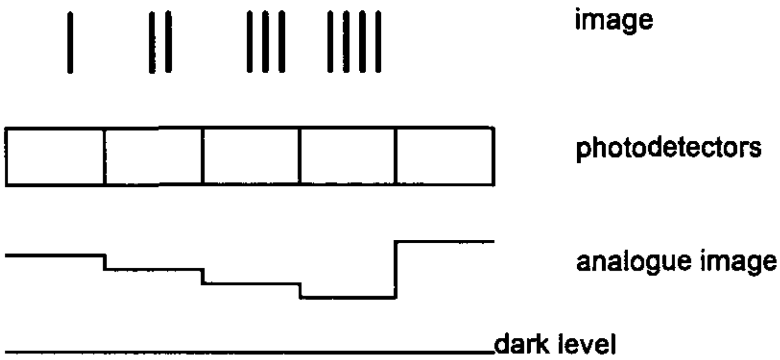


Figure 3.3 Loss of fine information

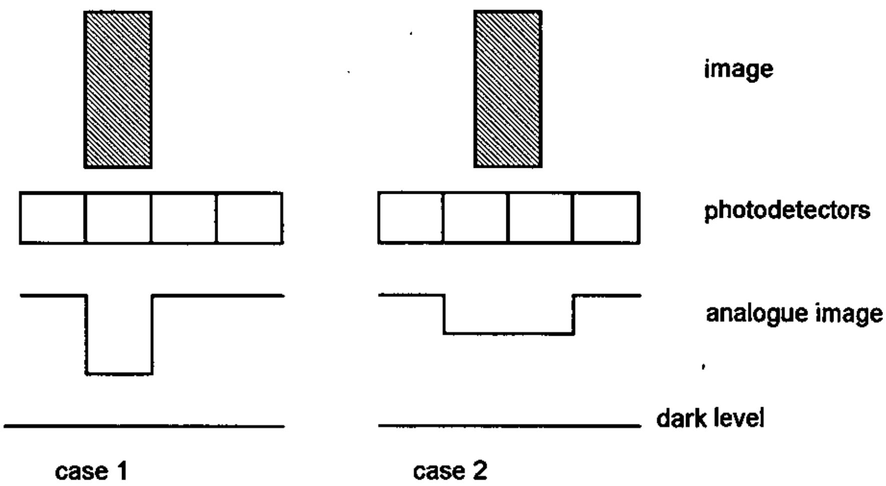


Figure 3.4 Effect of spatial quantisation

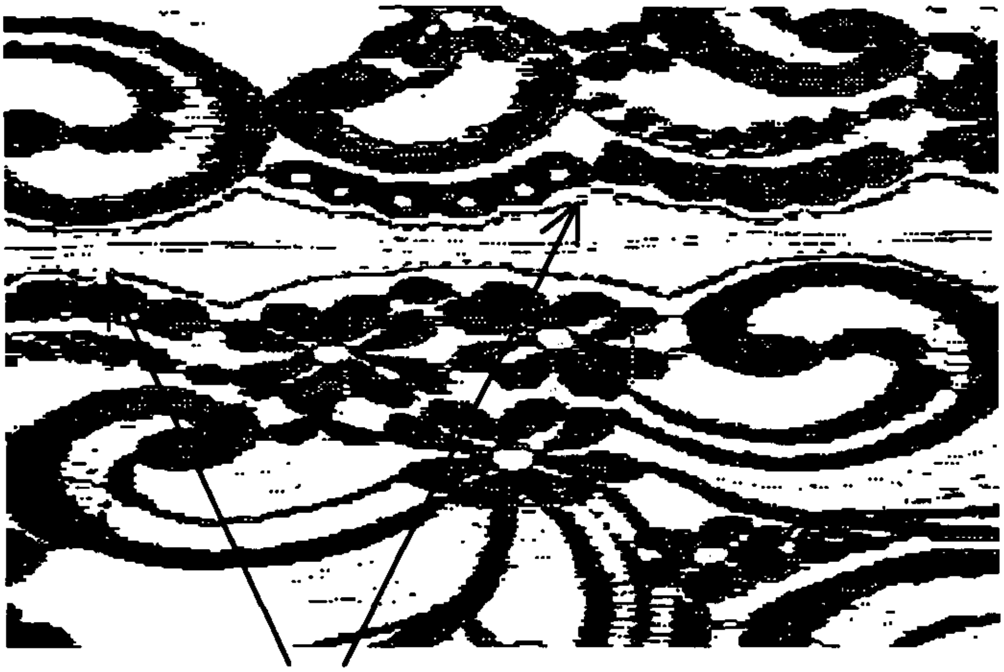


Figure 3.5a Lace imaged at 2.3 pixels per mm

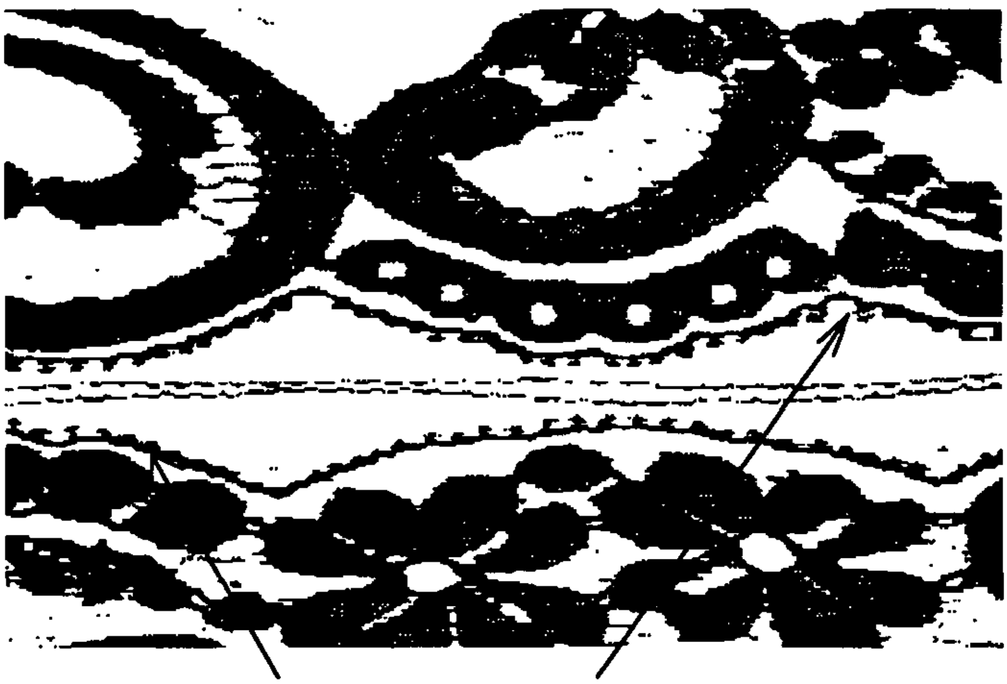


Figure 3.5b Lace imaged at 4 pixels per mm

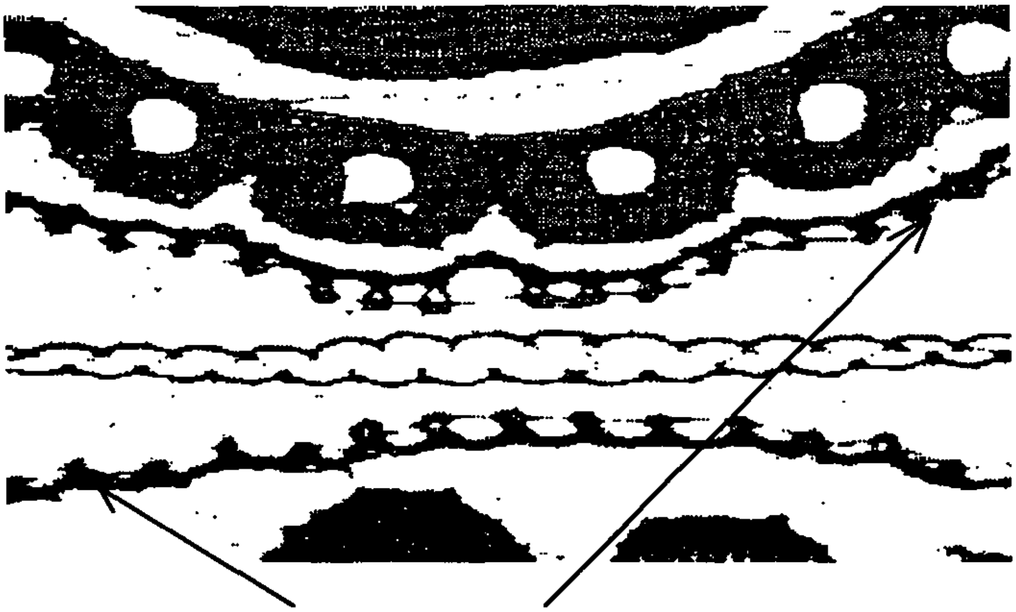


Figure 3.5c Lace imaged at 11 pixels per mm

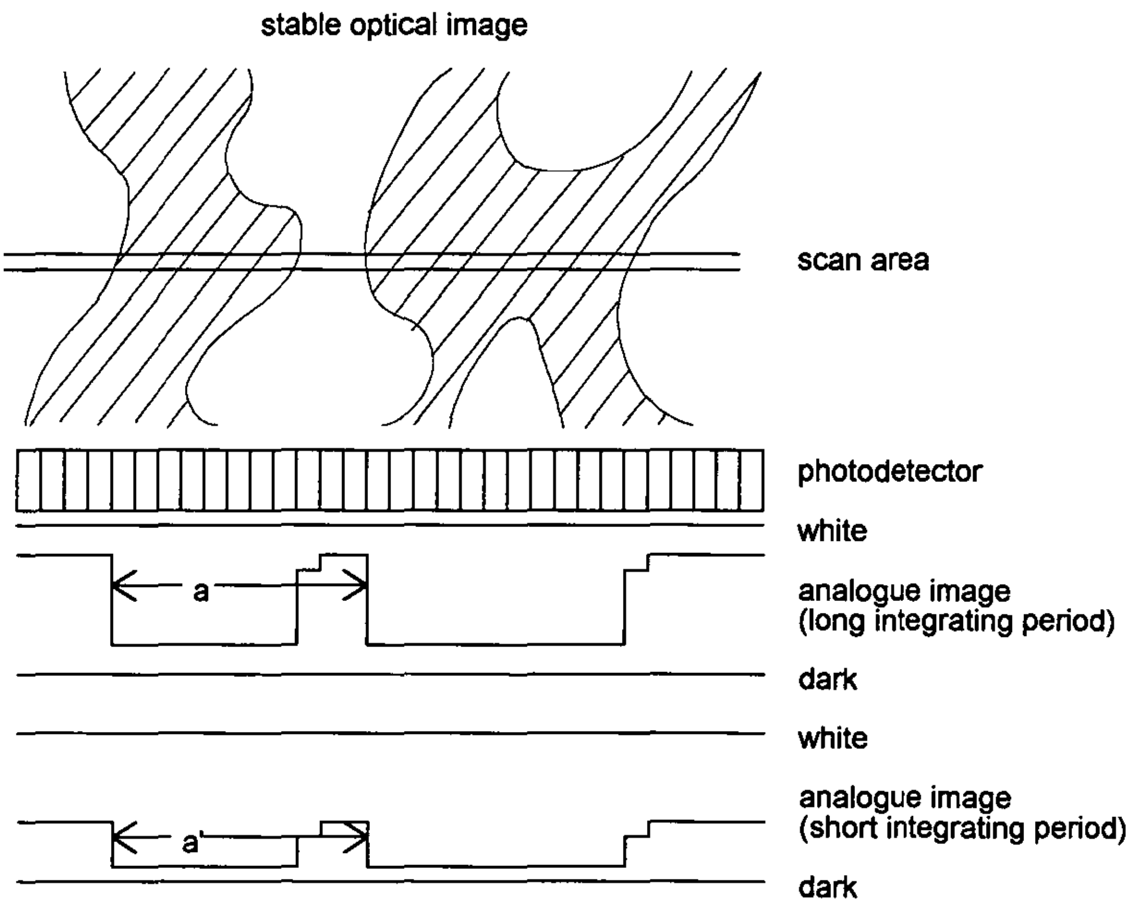


Figure 3.6 Stable optical image with different integrating periods

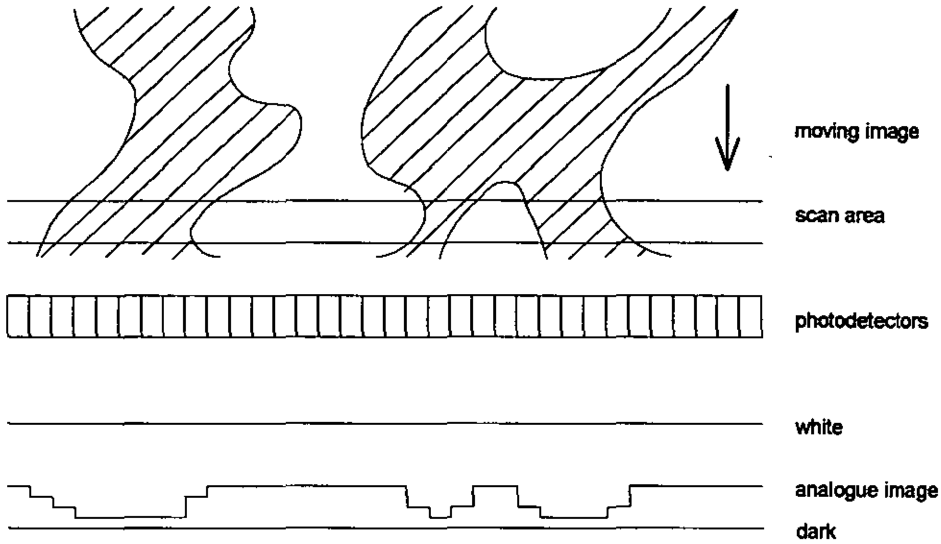


Figure 3.7a Moving optical image with short integrating period

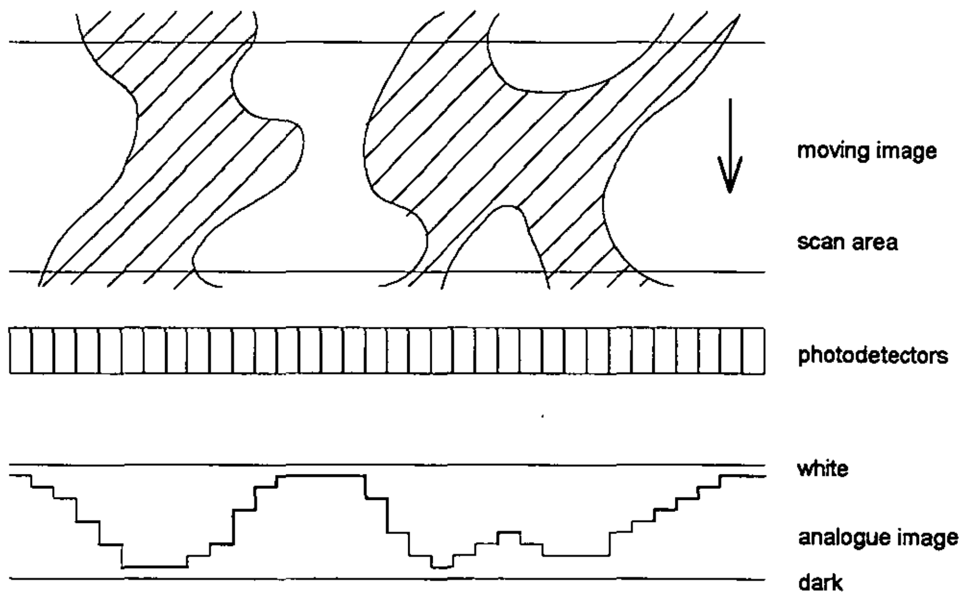


Figure 3.7b Moving optical image with long integrating period

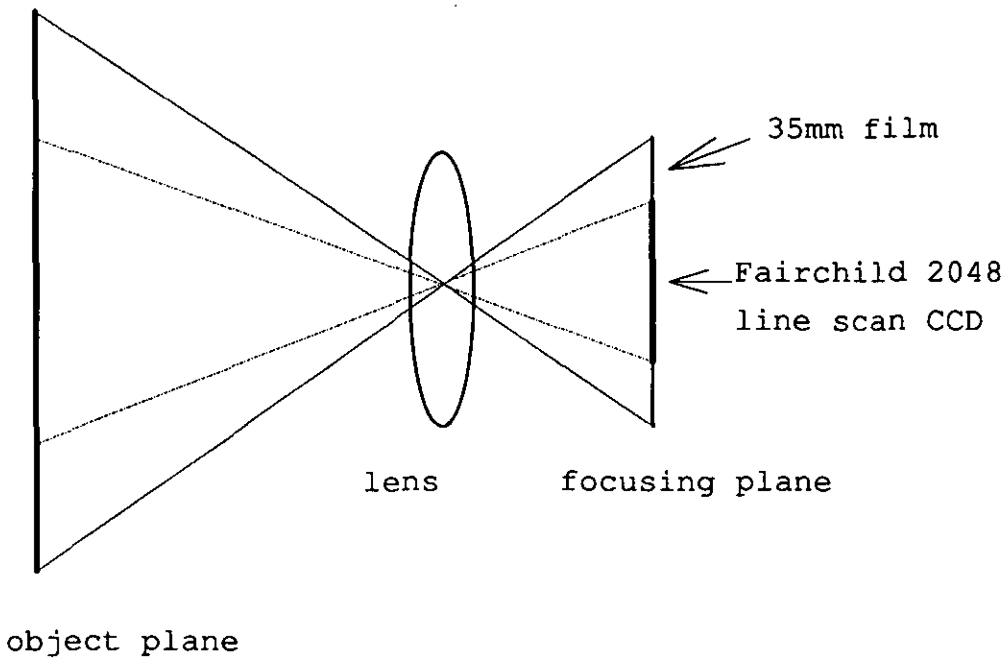


Figure 3.8 Different view angles by using different lengths of films on the same focusing plane

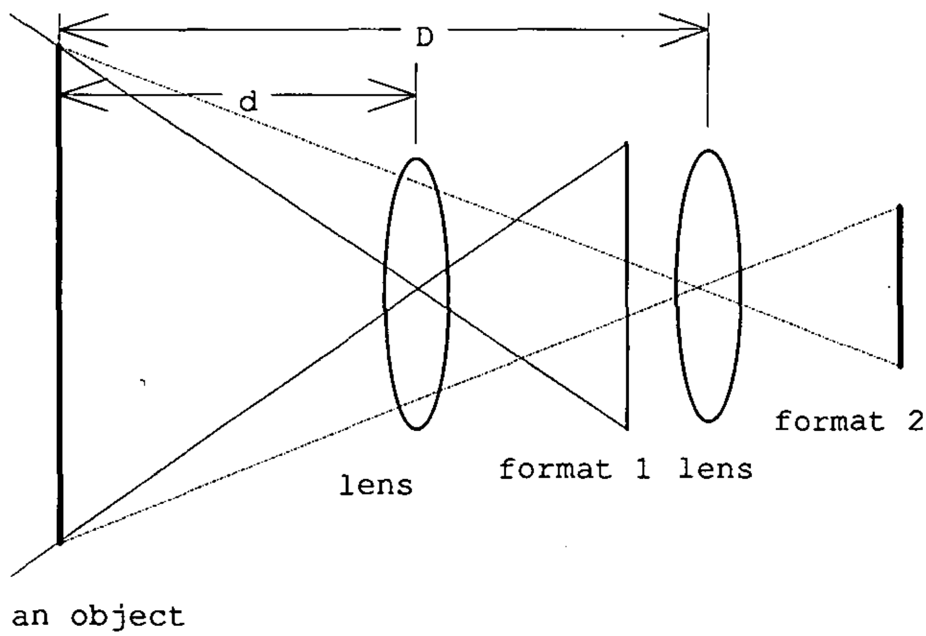


Figure 3.9 Narrower film, longer object-lens distance

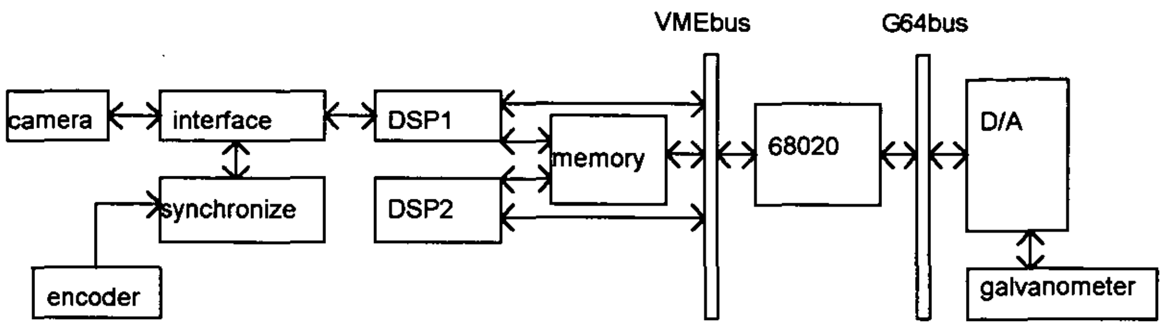


Figure 3.10 Vision system two

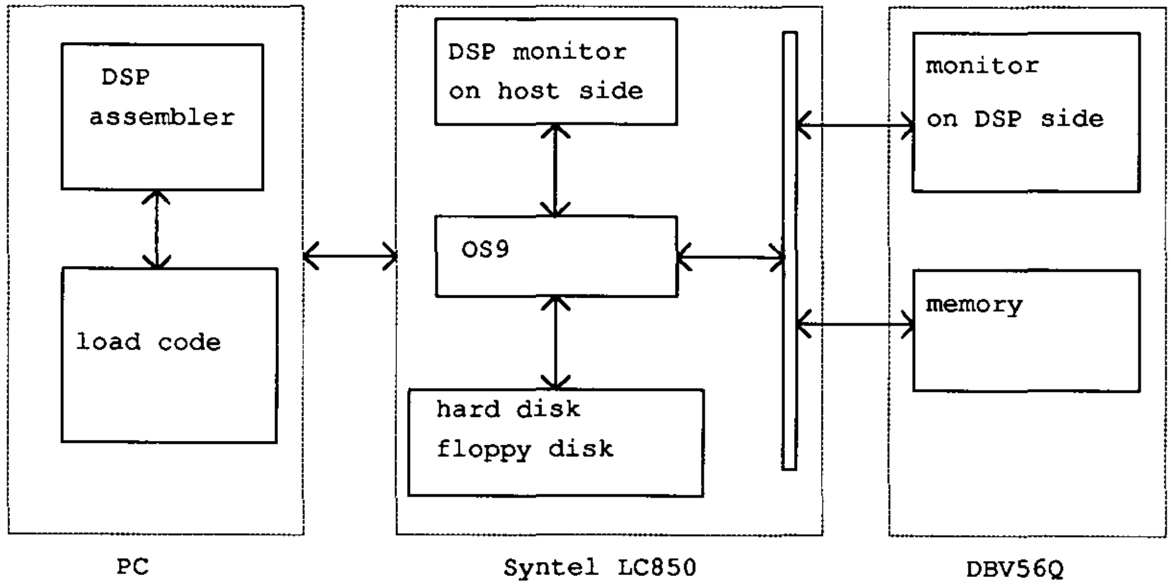


Figure 3.11 DSP code development system

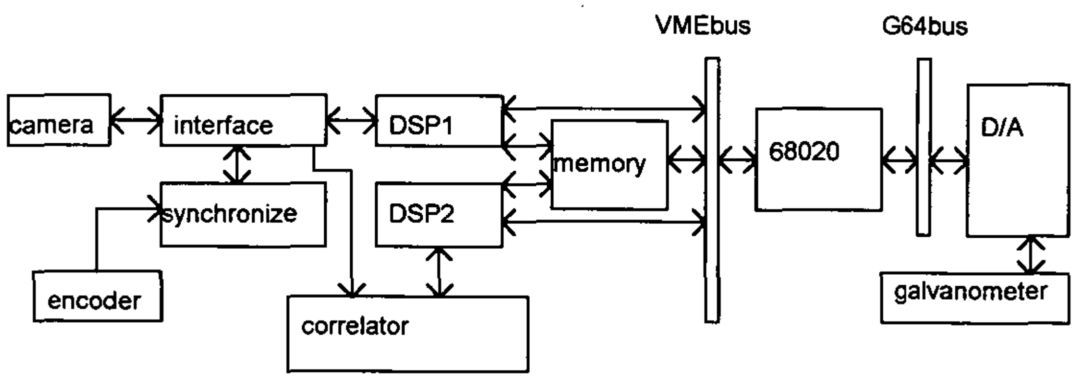


Figure 3.12 Vision system three

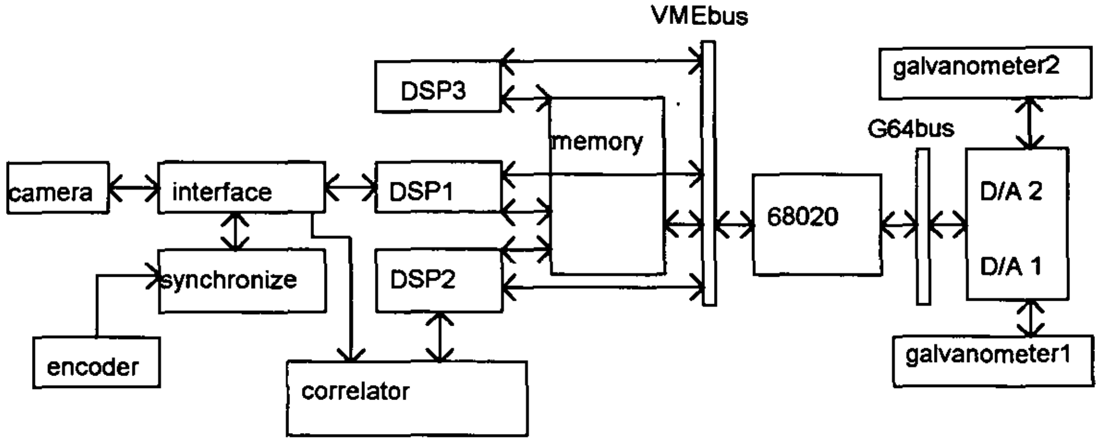


Figure 3.13 Vision system four

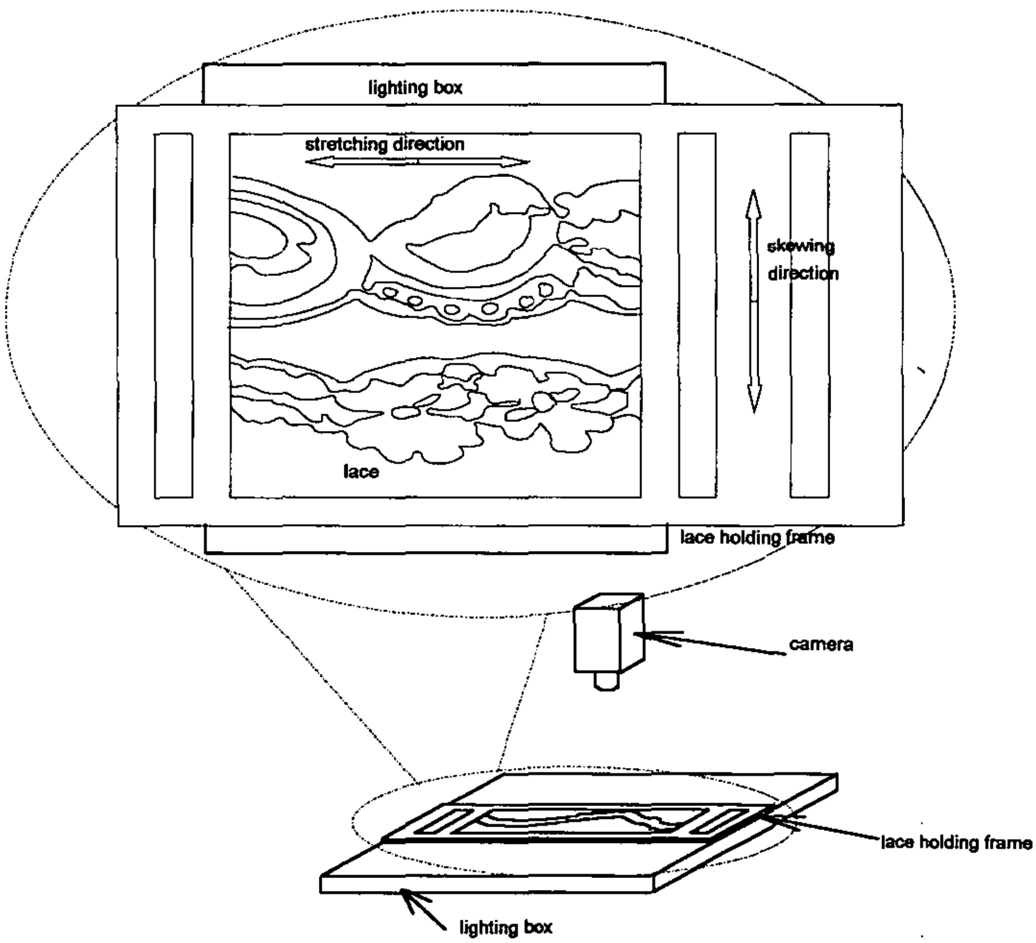


Figure 3.14 Rig one

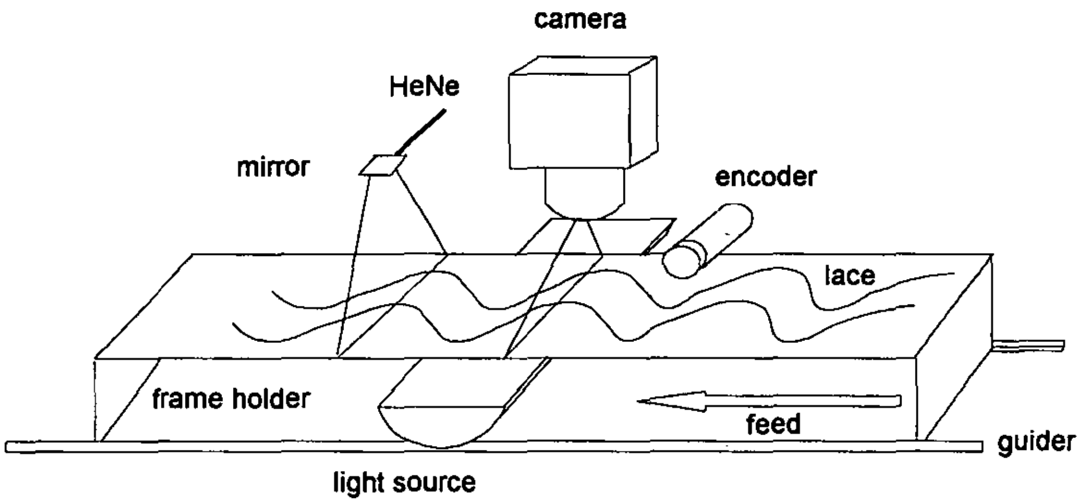


Figure 3.15 Rig two

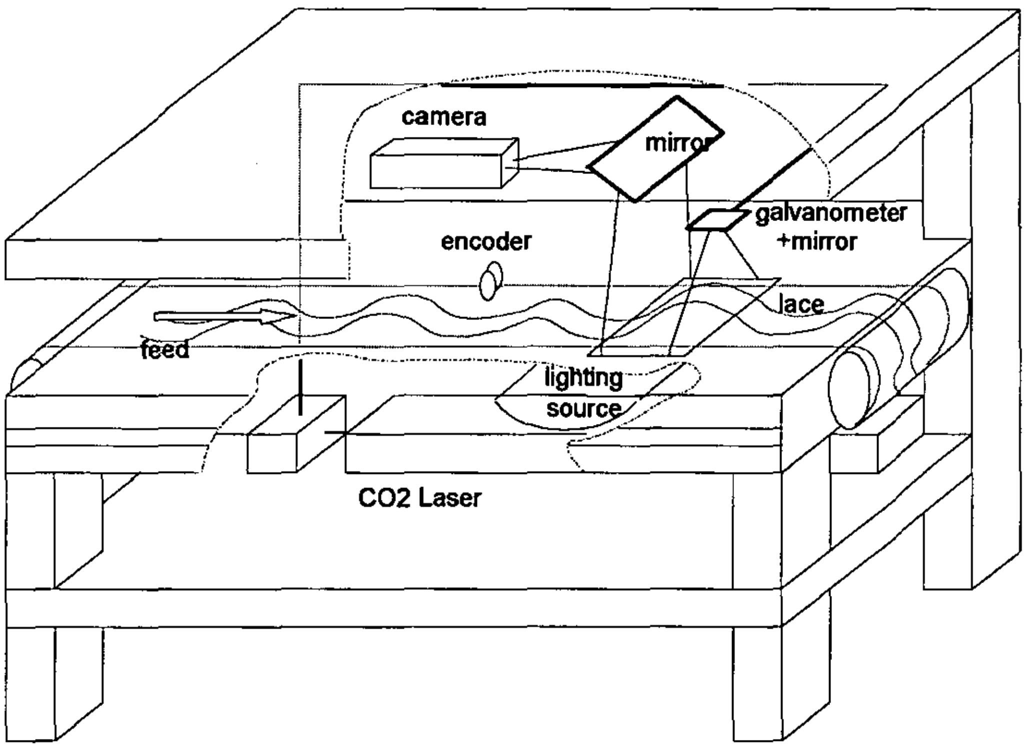


Figure 3.16 Rig three

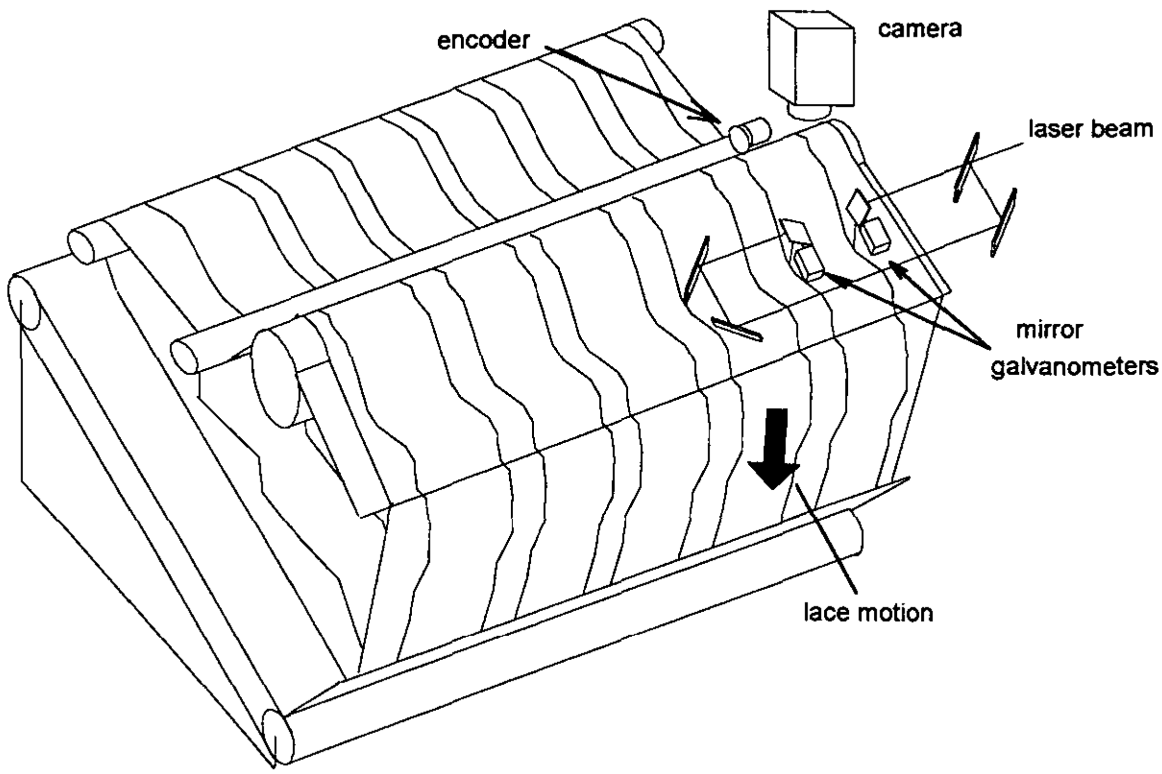


Figure 3.17 Prototype

Chapter 4

AN INCREMENTAL REAL-TIME PATTERN TRACKING ALGORITHM

4.1 INTRODUCTION

The three image processing tasks of the vision system for lace cutting are 1) to find the best matched position between the actual position of the image and the mapped starting point, this is achieved by global searching; 2) to track a predetermined cutting path on the incoming image by local searching; 3) to maintain stable working conditions for the vision system. This chapter will deal with the second task. Chapter 5 will deal with the first task because one approach to finding the start point is to use the algorithm which is described in this chapter. Chapter 6 will deal with the third task.

The algorithm described in this chapter has been developed to allow pattern tracking on a web of lace scanned by a fixed line-scan camera with its line of scan perpendicular to the direction of web motion. It is assumed that the patterning is more or less continuous along the web, so that each line scanned will normally provide some information for tracking.

It is also assumed that the path to be tracked does not double back on itself, a constraint which follows anyway if the lace can only move in a forward direction whilst the mechanism following the tracked path moves only perpendicular to the direction of web motion. However, certain type of path that double back (re-entrant) could be tracked by an extension of this method. In theory a simple double backed path can be replaced by two or three non-double backed paths with extra cutting information.

Since the patterns in this application are essentially binary, the video information is merely thresholded at a preset level. This considerably reduces the amount of data which has to be handled in real-time.

A further important requirement of the system is that it should be robust; i.e. it should be capable of maintaining tracking when presented with patterns which are distorted and possibly flawed. In particular, the ability to cope with patterns which are stretched or compressed across and along the web is essential, since the web material is elastic, varies dimensionally as a consequence of its method of production by around ±5%, and its transport tension past the imaging system cannot be fully controlled. Tests have been performed to investigate the robustness of the approach when used with noisy signals and web patterns which are geometrically distorted from their nominal form, laterally and/or longitudinally, by up to 10%.

In section 4.2 a tracking algorithm based on centre-weighted one-dimensional binary cross-correlation is discussed. In section 4.3 a method to cope with longitudinal distortion is presented and results obtained from test rig 1 (Figure 3.12) and vision system 1 (section 3.6.1) are presented and discussed in section 4.4. Further discussions about DSP implementation is in section 4.5.

4.2 BASIC ALGORITHM

Essentially, this is a centre-weighted one-dimensional binary cross-correlation. In section 4.2.1 the terms to be used to explain the algorithm are listed. The one-dimensional binary cross-correlation is presented in section 4.2.2. The algorithm is explained in section 4.2.3.

4.2.1 Terminology for the Algorithm

x ----------- one axis of co-ordinate of the input image in the opposite direction to the motion of the lace. The unit increment of x is one line of the scanned image. x equals i.

y ------------ the other axis of the input image perpendicular to the motion of the lace. The unit increment of y is one pixel in the line of the input image, x and y form a co-ordinate system in the input image, and the origin of the co-ordinate system is fixed to the input image.

ξ, ψ --------- the axes being used in the map corresponding with those of x,y, in the input image respectively (Figure 4.1). ξ equals j.

q ----------- displacement of one-dimensional cross-correlation search.

$H(i,j,q)$------ the historical matching result, produced by using an infinite impulse response (IIR) filter.

$Ig(x,y)$ ------ the image of the patterned lace to be tracked, being called an input image.

$Lcorr(i,j,q)$ --the weighted cross-correlation between the ith line of the input image and the jth line of the map at position q.

$Ma(\xi, \psi)$ ----- the image of lace with nominally the same pattern as the input image to be used as a reference in the tracking, being called a map.

$P(i)$ --------- the correct cutting position on line i of the input image.

$Pma(\xi)$ ----- the path to be followed, predefined across the map, being called a reference path.

$Pim(x)$ ------- the path to be found across the input image, being called a found path, or cutting path.

$W(y)$ --------- the weighting function.

$Y(i)$ --------- the predicted point for the cutting path on the line i of the input image.

β ------------ the coefficient of the first order IIR filter.

σ ----------- the tracking error in pixels.

σ^2 ---------- the tracking deviation.

4.2.2 One-Dimensional Binary Cross-Correlation

Using a method similar to that in (Jain 1989), an equation for one dimensional cross-correlation can be derived. The presence of a known signal $Signal_1(y)$ in an input signal $Signal_2(y)$ can be detected by searching for the location of match between $Signal_1(y)$ and $Signal_2(y)$. The matching can be conducted by searching the displacement of $Signal_2(y)$, where the mismatch energy is minimum. For a displacement (q), the mismatch energy, Ω, is defined as

$$\Omega(q) \propto \sum_y [Signal_1(y) - Signal_2(y-q)]^2$$
$$= \sum_y |Signal_1(y)|^2 + \sum_y |Signal_2(y)|^2 - 2\sum_y Signal_1(y)Signal_2(y-q) \qquad \text{eq.4.1}$$

For the mismatch energy to achieve its minimum, it is sufficient to maximise the last term of equation 4.1, the cross correlation

$$Corr(q) = \sum_{y} Signal1(y)Signal2(y-q) \qquad \text{eq.4.2}$$

Although the image and map are two dimensional signals in the human visual system, they can be represented in one dimension inside the computer. For example, if $z=x \times l+y$, where the l is the length of the image in y direction, the two dimensional image $Ig(x,y)$ equals a one dimensional image $Ig(z)$. Actually the image from a line scan camera is formed from one line image after another. If we fix the value of x, namely i, vary the value of y, the z satisfies $i \times l \leq z < (i+1) \times l$. In other words, the $Ig(i,y)$ $(i=0,1,2,...)$ is part of the one dimensional image $Ig(z)$. If $Signal_1(y)$ equals $Ma(j,y)$, and $Signal_2(y)$ equals $Ig(i,y)$, where i and j are fixed, equation 4.2 will be changed to equation 4.3.

$$Corr(i,j,q) = \sum_{y} Ma(j,y)Ig(i,y-q) \qquad \text{eq.4.3}$$

What equation 4.3 points out is that if to search for the maximum cross-correlation between map and image in one direction, and if the i and j in the map and image are known beforehand, search can be carried out in a limited range of positions in one direction to find the maximum instead of searching two directions or the whole range.

When the data is binary equation 4.3 will be equivalent to the equation 4.4.

$$Corr(i,j,q) = \sum_{y} \overline{Ma(j,y) \oplus Ig(i,y-q)} \qquad \text{eq.4.4}$$

In this case, it is sufficient to search for the maximum of the total binary agreement between $Ma(j,y)$ and $Ig(i,y)$.

4.2.3 Incremental Tracking by Using Weighted One Dimensional Binary Cross-Correlation and IIR Filter

Initially, consider that the image and map are scanned with the same rate in x and ξ directions and that the image has the same pattern repeat length in x direction as the map in ξ direction. Suppose before tracking begins a best matched position between the map and the input image around a relatively small area of the reference path has been found by methods described in chapters 5. In this case, ξ for the map is determined according to the input image $Ig(0,y)$; and $Pim(0)$ is found according to the $Pma(\xi)$. This is similar to the 'steady state' case in the middle of the tracking operation. The start point of the path on the input image can be expressed as

$$y = Pim(x) \qquad (x = 0, 1......) \qquad \text{eq.4.5}$$

where y is the pixel number and x the line number of the input image. For the start point of the path x=0.

In general, x equals i-1, where i is the line number of the current line of input image.

Similarly, the start point of the path on the map can be expressed as

$$\psi = Pma(\xi) \qquad \text{eq.4.6}$$

Generally, ξ equals j-1, where j is the line number of the current line of the map.

The first stage in the tracking process is to calculate a weighted cross-correlation between a sub-section of n pixels (the 'matching length') of the ith line of the input image and the jth line of the map;

$$Lcorr(i,j,q) = \sum_{Y(i)-n/2}^{Y(i)+n/2} W(y - Y(i) + n/2)\{\overline{Ma(j, y - Y(i) + Pma(j)) \oplus Ig(i, y - q)}\}$$

eq.4.7

Equation 4.7 is similar to equation 4.4 except a weighting value $W(y\text{-}Y(i)+n/2)$ is applied. Because the segment of the input image used for the cross correlation is nominally centred on the (most recently) found path the use of a centre-weighting function reduces errors due to distortion and mismatch of scale between input image and reference map data.

The centre of the sub-section of the jth line of map for which the weighted cross-correlation is computed is $Pma(j)$. The centre of the matching length of the ith line of the input image is predicted by:

$$Y(i) = Pim(i-1) + Pma(j) - Pma(j-1) \qquad \text{eq.4.8}$$

Figure 4.2 shows the relationships of *Pma(j-1)*, *Pma(j)*, *Pim(i-1)* and *Y(i)*. Assuming that the $\Delta\xi$ equals the Δx and $\Delta\psi$ equals Δy, we can get equation 4.8 according to the linearization and standard linear approximation (Thomas 1988). Assuming the first differentials of *Pma(j-1)* and *Pim(i-1)* are *P'ma(j-1)* and *P'im(i-1)*. The only difference between the linear approximation and equation 4.8 is that the differential form of the *P'im(i-1)* is replaced by the *P'ma(j-1)* because we do know that the *P'ma(j-1)* is similar to the *P'im(i-1)*. The purpose to predict the *Y(i)* is not to determine the *Pim(i)*, the cutting path, on line *i* of the input image but to form a sub-section on line *i* of the input image in which the *Pim(i)* will be searched. The length of the sub-section of the input image is *n+q* and it is centred at *Y(i)*. Figure 4.3 shows how to match map with the image.

Table A - 'linear ramps':
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31 32 32 31 30 29 28 27 26 25 24 23
22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Table B - '4 steps':
1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1

Table C - '2 steps':
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Table D - 'all ones':
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Table 4.1 Look-up tables used for line centre-weighting (shown for 64 pixel matching width)

The matching length, n, used needs to be large enough to ensure that a representative portion of the pattern is used in the cross-correlation. Restricting the cross-correlation to a sub-section of the image line, rather than using the whole line, allows a wide image line (and hence a large amplitude of tracking path excursions) independent of the requirement to minimise the matching length to achieve maximum computational speed. Similarly, the weighted cross-correlation must be computed for sufficient lag (pixel shift) values, q, to ensure that the maximum cross-correlation will be found. This depends on the maximum change of path position from one line to the next. For the lace patterns tested here correlations were computed for 32 lags ($-16<q<=16$ pixels). Weighted cross-correlations for lag values above and below this range are not computed but are set equal to the values at the maximum positive and negative lags respectively (for use by the x-directional matching algorithm described in section 4.3).

Four types of centre-weighting function were evaluated with the algorithm as shown in table 4.1. A linear taper (weighting table A) was used in initial trials and it was found that this could be replaced by a stepped function with 4 steps (table B) to speed computation without significant degradation of tracking performance. Another stepped function with 2 steps (table C) and a function without weighting were tested. The test results are shown in Figure 4.4 where variation of tracking error shown against the lateral scaling error of the image relative to the reference map.

In Figure 4.4 the tracking error is calculated as σ^2 defined as:

$$\sigma^2 = \frac{1}{n}\sum_{i=0}^{n}\{Pim(i) - P(i)\}^2 \qquad \text{eq. 4.9}$$

where $P(i)$ is the correct cutting path on the ith line of the image (Figure 4.5), and n is the number of the lines of the image being tracked.

The bigger σ^2, the greater the tracking error. When the distortion (i.e. lateral scaling mismatch between image and map) is 0%, the tracking errors produced by the four weighting functions are the same. As the distortion increases, the tracking errors of all weighting functions are increased.

Comparing different weighting functions, that in Table D has the biggest error and Table A has the smallest error. The functions in Table B and C give intermediate results.

In the real world, the map and image are acquired separately and some localised mismatches, caused by noise and distortion within the cross-correlation, are inevitable. For example, the map (Figure 4.6) is 12% bigger than image (Figure 4.7b). In Figure 4.7b there is a 'hole' 20 by 35 pixels. The equation 4.7 is directly used to find cutting path. On the area of the 'hole' the cutting path is broken. Figure 4.7a shows the tracking error across the hole.

One approach to overcome the localised mismatches is to apply redundant trackings to the patterned material and to use a sophisticated decision making mechanism. At lest two paths are tracked in parallel. One is the real cutting path, another is the back-up path. If one of paths loses its tracking, the decision based on another path can keep the real cutting path on the correct position. But if both paths lose their way, it needs another back-up tracking path or other method to keep the real cutting path on the correct position. Obviously it needs double or triple computing power.

A first-order infinite impulse response (IIR) filter (Johnson 1989, Proakis 1988) function used to combine the output from the line currently being processed with historical data from previous lines is another approach to tackle localised mismatches;

$$H(i,j,q) = Lcorr(i,j,q) + \beta\{H(i-1,j-1,q)\} \qquad \text{eq. 4.10}$$

where $H(i\text{-}1,j\text{-}1,q)$ represents the previous filter output.

It is necessary to find the range of β to ensure that the $H(i,j,q)$ is convergent. If we only consider the positive range due to the addition of the historical data, the equation 4.10 can be rewritten to equation 4.11. The $H(q)$ stands for $H(i,j,q)$, and the $ln(q)$ for $lcorr(i,j,q)$.

$$H_n(q) = l_n(q) + l_{n-1}(q) \times \beta + l_{n-2}(q) \times \beta^2 + \ldots\ldots$$
$$= \sum_{i=1}^{n} l_i \times \beta^{n-i} \leq l_{max} \sum_{i=1}^{n} \beta^{n-i} = l_{max} \sum_{i=1}^{n} \beta = l_{max} \frac{1-\beta^n}{1-\beta} \quad \text{eq. 4.11}$$

where $l_{max}(q)=Max\{l_i(q) \quad (i=0,1,\ldots\ldots,n)\}$. When n goes to infinity, the condition for $H_n(q)$ to be convergent is $0 \leq \beta < 1$. Then we have equation 4.12 as follows:

$$\lim_{n\to\infty} Hn(q) \leq \lim_{n\to\infty} \{l_{max}(q) \times \frac{1-\beta^n}{1-\beta}\} = l_{max}(q) \times \frac{1}{1-\beta} \quad \text{eq. 4.12}$$

coeff.ß	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
line 1	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
line 2	0.36	0.42	0.49	0.56	0.64	0.72	0.81	0.90
line 3	0.21	0.27	0.34	0.42	0.51	0.61	0.73	0.86
line 4	0.13	0.18	0.24	0.32	0.41	0.52	0.66	0.81
line 5	0.08	0.12	0.17	0.24	0.33	0.44	0.59	0.77
line 6	0.05	0.08	0.12	0.18	0.26	0.38	0.53	0.74
line 7	0.03	0.05	0.08	0.13	0.21	0.32	0.48	0.70
line 8	0.02	0.03	0.06	0.10	0.17	0.27	0.43	0.66
line 9	0.01	0.02	0.04	0.08	0.13	0.23	0.39	0.63
line10	0.01	0.01	0.03	0.06	0.11	0.20	0.35	0.60
line11	0.00	0.01	0.02	0.04	0.09	0.17	0.31	0.57
				……				
line15	0.00	0.00	0.00	0.01	0.04	0.09	0.21	0.46
				……				
line20	0.00	0.00	0.00	0.00	0.01	0.04	0.12	0.36
				……				
line22	0.00	0.00	0.00	0.00	0.01	0.03	0.10	0.32
				……				
line45	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.10

Table 4.2 First-order IIR filter transmissions for a range of β values

The position of the found path at the ith line of the input image is decided by;

$P_{im}(i) = Y(i) + q' - q/2$ eq. 4.13

where $H(i,j,q') = Max\{H(i,j,k)\}\ \ 0 \le k \le q$

Tests were undertaken to show how the value of β affects the tracking behaviour across a 'hole' (i.e. a localised defective area of the input image). The map (Figure 4.6) is 12% wider than the images (Figure 4.8b). In Figure 4.8b there is a hole that is 35 by 20 pixels. In Figure 4.8 the cutting path is continued due to the integrating effect of β. The tracking errors in Figure 4.7a and 4.8a are as follows:

$\sigma = P_{im}(i) - P(i)$ eq. 4.14

The use of the IIR filter allows a degree of two-dimensionality to be achieved in the matching process, rather than a simple line-by-line comparison, but without greatly increasing the computational complexity. The value of the parameter β determines the rate at which the historical data 'fades-away' from the matching process.

Table 4.2 shows the 'fade out' effect for an appropriate range of values of the coefficient β.

4.3 MATCHING ALGORITHM FOR X-DIRECTIONAL DISTORTION

The discussion so far has assumed line by line synchronisation between the map and image in the x (and ξ) direction. This cannot be guaranteed in practice, not only because of sampling quantisation but also due to unavoidable sampling rate differences (in lines/mm of map or image) which may be due to transport velocity or tension related longitudinal scale differences between map and image. These result in the length of the map not matching that of the image. Matching between different lengths of map and image is, therefore, now considered.

Following are the two steps of the strategy of x directional matching:

4.3.1 The Best Matching Selection

Assuming the (j-1)th line of map and the (i-1)th line of image have previously been matched with each other, the matching of the ith line of input image is considered. The ith line of input image is compared with the (j-1)th, jth and (j+1)th lines of the map respectively (Figure 4.9). In other words, the values of matching for three pairs, *Lcorr(i,j-1,q)*, *Lcorr(i,j,q)* and *Lcorr(i,j+1,q)*, are calculated and the best value is chosen among the three pairs. Then this line matching value is fed into the IIR filter using equation 4.10.

The explanations for the line matching results are as follows. If the ith line of image best matches the (j-1)th line of map, this means the image is longer than (or 'ahead of') the map in the longitudinal direction at this point (Figure 4.10a). If the ith line of the image best matches the jth line of map, that means the image is synchronised with the map at this point (Figure 4.9). If the ith line of the image best matches the (j+1)th line of the map, it means the image is shorter than (or 'behind') the map at this point (Figure 4.10b). If the distortion is even all over the patterned material, this approach can only suit three cases; 1) map and image have the same pattern repeat length, the ith line of the image always matches the jth line of the map; 2) the pattern repeat length of the image is the half length of the map. The ith line of the image always matches the *(j+1)*th line the map; 3) the pattern repeat length of the image is infinity so the ith line of the image always matches the *(j-1)*th line of the map.

4.3.2 The Forced Matching Selection

There are various cases in the real world so two forced decisions in tracking are required. Firstly the possibility of requiring a forced decision in tracking may result if matching results are the same for two consecutive match pairs. For example, if the ith line of image can match both the (j-1)th and the jth lines of map or both the jth and the (j+1)th lines of map. In this case the jth line of map is chosen as the matched line to keep the synchronisation between map and image. If the ith line of image can match both the (j-1)th and the (j+1)th lines of map, the (j+1)th line of map is chosen to keep the map moving forwards.

For convenience of description, a 'wait' is defined if the *i*th line of the input image matches the (j-1)th line of the map; a 'step' is defined if *i*th line of the input image matches the *j*th line of the map; and a 'jump' is defined if the *i*th line of the image matches the (j+1)th line of the map. The 'wait' and 'jump' are the exceptional tracking actions.

The second forced decision may need to be taken if an exceptional tracking condition has occurred twice in sequence. If a 'wait' is required twice in sequence after a step, that indicates that the image length from (i-1) to (i+2) is three times longer than the map length from (j-1) to j (Figure 4.11). The real image cannot be three times longer than the map so the algorithm forces a 'step' once without any match condition. After that action, the length of the image at that point is one and half times longer the length of the map (Figure 4.12).

If it 'jumps' twice in sequence after a step, it means that the image length from (i-1) to (i+2) is three-fifths of the length of the map (Figure 4.13). The real lace cannot be such a short length so that the algorithm forces a 'step' once without any match condition. After that action the length of the image at that point is three-quarters of the map length (Figure 4.14). This step of matching also provides for the cases that the match can 'wait' twice after a 'jump' (Figure 4.15) and the match can 'jump' twice after a 'wait' (Figure 4.16).

All these keep the performance of the algorithm correct when the length of the image is within the range from one and half times longer than the length of the map to three-quarters shorter than the map. The working range of the algorithm is over ±10% larger than the real lace variation.

4.4 EXPERIMENTAL TESTS ON PC

In order to test the algorithm, several experiments have been conducted. Initial non real-time testing was performed on an IBM PC-AT compatible computer using stored data obtained from an area-scan CCD camera (vision system 1, chapter 3) and static lace samples (test rig 1, chapter 3).

Scaling and distortion errors in the input image were introduced by manually stretching (or relaxing) the lace fabric laterally and/or longitudinally between capturing the image used as the reference map and the image to be tracked. The path to be tracked was referenced to the map data by manual input using the keyboard. To simulate noise in the images the captured image data was degraded by applying 'white-noise' by modifying a set percentage of individual image pixels selected by a random number generator.

Figure 4.17 is computer output from tracking a lace sample with a fine pattern (pattern p1) with a resolution of 10 pixels/mm. The black pattern is the image of the figuring threads in the lace. The thin black line running from top to bottom close to the centre of the picture is the reference path to be tracked; it represents a cutting path which will separate the lace figuring from its background mesh (not visible here) without damaging the scalloping edge. The grey area to each side of the found path shows the extent of the matching width.

Figure 4.18 is from another lace sample with a large pattern (pattern p2). The resolution is again 10 pixels/mm. The cutting line here is for a 'centre-cut'.

A summary of some of the tests conducted for two lace patterns is given in table 4.3. Distortion of the pattern by stretching in the y direction was applied in 1% steps, while the coefficient β of the IIR filter was varied between the limits shown in table 4.3 in increments of 0.05. All the tracking tests were performed with and without simulated noise. Detailed results for tracking pattern p1 are shown in Figures 4.19-4.20. Figure 4.19 shows the maximum tracking error, over a whole pattern repeat, for differing levels of stretch and filter coefficient. Less than two pixels maximum path error is achieved at up to 10% stretch across a range of values of β. Figure 4.20 shows the maximum tracking error with 2% random noise in the image (but not in the reference map) for the same test series as Figure 4.19. (Average tracking errors of less than 0.2 pixels are achievable under realisable operating conditions.) Figures 4.21 and 4.22 show the similar performance for tracking pattern p2. Performance of tracking with 2% random noise in the image is slightly degraded but is still quite acceptable for the intended application. Space does not permit the inclusion of detailed results for all the tests (see Appendix 2) but in all cases maximum tracking error at 10% stretch was 3 pixels or less, with or without simulated noise, when β values of 0.8-0.9 were used.

A summary of the experiments is given in table 4.3.

The test results show several things. Firstly, as might be expected, the performance of the algorithm is sensitive to the pattern. In general, the matching results on p2 are better than p1. The minimum width of match pixels required for successful matches with p2 is lower than for p1, so less computation time is used for p2. The range of distortion permissible in the y direction in matching p2 is also bigger than that permissible in matching p1. Secondly, for a particular pattern, the lower the image resolution, the lower minimum matching width that can be used. The third observation is that the bigger IIR filter coefficient (β) values generally give better tracking, but, predictably, excessive values prevent the tracking from following sharp deviations in the reference path. For the lace patterns and cutting paths tested, coefficients between 0.75 and 0.85 performed well. Degradation of the image data by superimposing 2% random noise had no significant effect on the matching results of pattern p2, but was significant for p1 with the lower resolution of 2pixels/mm. Finally, increasing the matching width does not necessarily improve the match result on pattern p1.

pattern	p1	p2	p1	p2	p1
resolution (pixels/mm)	10	10	4	4	4
matching width, n pixels	128	96	96	64	128
no noise: coefficient β, y distortion %	 0.4-0.90 1-15	 0.4-0.90 1-15	 0.4-0.90 1-11	 0.4-0.90 1-15	 0.4-0.90 1-11
2 % single-pixel noise: coefficient β , y distortion %	 0.4-0.95 1-15	 0.4-0.95 1-15	 0.7-0.90 1-11	 0.4-0.95 1-15	 0.75-0.90 1-11

Table 4.3 Summary of test conditions for the tracking algorithm.

4.5 IMPLEMENTATION OF PATTERN TRACKING ON DSP

There are two program languages for DSP56001 coding, assembly and C. Normally C programming language is more friendly to programmers than assembly language, but would be slower in operation. Due to the real-time requirement of pattern tracking, assembly language is used to implement the code for DSP56001. Also assembly level programming allows detailed knowledge of the DSP architecture and instruction set to be exploited at the optimisation stage of the algorithm software research (see chapter 7).

In order to explain the DSP implementation clearly and not to involve too much detail of the DSP assembly instruction set, a pseudo code is used in this section.

The executive program of tracking by using DSP56001 can be described as follows:

```
Tracking()
{
        initialisation(i,j,H(i,j,q),Y(i),increment,increment_p);
        do {
            form_segment_image(I_g,i,Y(i));
            one_dimensional_correlation(I_g,M_a(j),L_corr(i,j,q));
            one_dimensional_correlation(I_g,M_a(j+1),L_corr(i,j+1,q));
            one_dimensional_correlation(I_g,M_a(j+2),L_corr(i,j+2,q));
            choice(increment,increment_p,L_corr(i,j+increment,q),L_corr(i,j,q),
              L_corr(i,j+1,q),L_corr(i,j+2,q));
            for(k=0;k<q;k++)
              H(i,j+increment,k)=Lcorr(i,j+increment,k)+β×H(i,j,k);
            tracked_point_decision(Pim(i),Y(i),H(i,j+increment,q));
            i=i+1;
            j=j+increment;
            prediction_new_point(Pim(i),Y(i),Pma(j),Pma(j+1));
            } while(i<tracking_length);
}
```

The procedure 'tracked_point_decision($Pim(i),Y(i),H(i,j+increment,q)$)' is the eq. 4.13 and 'prediction_new_point($Pim(i),Y(i),Pma(j),Pma(j+1)$)' is the eq. 4.8 so no further description. The procedure 'initialisation(i, j, $H(i,j,q)$, $Y(i)$, $increment$, $increment\_p$)' is to set $i=0$, $j=0$, $Y(i)=Pma(j)$, $H(i,j,q)=0$, $increment=1$, and $increment\_p=1$. '$i=0$' means that the tracking starts from the beginning of the image. '$j=0$' is similar to '$i=0$' (in section 5.3 in order to find a start point by using tracking method the value of j is changed). '$Y(i)=Pma(j)$' means that the first predicting cutting position directly uses the first point of reference path without modification (in section 5.3.1.2 an analysis shows the limitation of '$Y(i)=Pma(j)$'). The *increment* and *increment_p* stand for the matching actions for X-direction (see section 4.3) and the *increment_p* is the action one step early than the *increment*. If *increment* or *increment_p* equals zero, a 'wait' occurs; if 1, 'step'; if 2, 'jump'. At the beginning of tracking, no exceptional tracking actions are assumed so the *increment* and *increment_p* equal one.

Section 4.5.1, 4.5.2 and 4.5.3 discuss respectively the procedures as follows:

- form_segment_image(I_g, i, $Y(i)$);
- one_dimensional_correlation(I_g, $M_a(i)$, $L_{corr}(i,j,q)$);
- choice($increment$, $increment\_p$, $L_{corr}(i,j+increment,q)$, $L_{corr}(i,j+2,q)$, $L_{corr}(i,j+1,q)$, $L_{corr}(i,j,q)$).

4.5.1 Forming an Image Segment

The purpose of this procedure is to form a segment in the ith line of the image. The length of the segment is $n+q$ (matching_length + lag) and centre of the segment is at $Y(i)$ (see Figure 4.3). Firstly an image format in the memory is introduced. In order to store and move the image data in memory efficiently, a word (24 bit) in memory stores 24 pixels of image (binary) from the most significant bit (MSB) to the lest significant bit (LSB). Successive 24 bit pixels of image are stored in the next word in ascendant memory address, and so on. The format of the image storage maintains the continuity of image in logical location of memory. For example, an image segment, '1111 0111 1110 0001 1100 0011 1100 0001 1111 1100 0000 0011' (in binary) can be stored in 2 consecutive words, 0xf7e1c3 and oxc1fc03 (in hexadecimal). Alternately the image can be stored in 24 bit word from LSB to MSB and from high address of memory to low address, but this format is the same as the former format in principle and in this thesis the first format is used.

One problem in forming an image segment is how to transfer the beginning of the desired segment to the MSB of a 24bit word (that is convenient for the one-dimensional cross-correlation, section 4.5.2). The beginning point of the desired image segment, $B_{Ig}(i)$, in image coordinate is defined as the equation 4.15.

$$B_{Ig}(i) = Y(i) - (n+q)/2 \qquad \text{eq. 4.15}$$

The value of $B_{Ig}(i)$ can be within the width of the image. Then $B_{Ig}(i)/24$ is a displacement in words from the origin of the ith image and $B_{Ig}(i)\%24$, representing the first bit of the desired image, can be any number between 0 and 23, because the image is stored in 24bit words in successive memory locations. The concept for the procedure is as follows:

```
form_segment_image(Ig, i, Y(i))
{
      BIg(i)=Y(i)-(n+q)/2;
      word=BIg(i)/24;
      bit=BIg(i)%24;
      for(j=0;j<(n+q)/24;j++)
      {
            double_word=((Image(i,word+j)<<24)|(Image(i,word+j+1)));
            double_word>>=(24-bit);
            Ig(j)=double_word&0xffffff;
      }
}
```

4.5.2 One-Dimensional Correlation

This procedure implements equation 4.7. In order to simplify the implementation the Ig and Ma start from the MSBs. Assuming $q<=48$, the procedure is as follows:

one_dimensional_correlation_1(I_g, $M_a(i)$, $L_{corr}(i,j,q)$)

```
{
    for(i=0;i<q;i++) Lcorr(i)=0;
    for(i=0;i<n/24;i++)
    {
        triple_word=(Ig(i)<<48)|(Ig(i+1)<<24)|(Ig(i+2));
        word_map=Ma(i);
        for(j=0;j<q;j++)
        {
            word_image=(triple_word>>48)&0xffffff;
            corr=word_image XOR word_map;
            for(k=0;k<24;k++)
            {
                if((corr&ox800000)==0x800000)
                    Lcorr(j) += weightA(i*24+k);
                corr <<= 1;
            }
            triple_word<<=1;
        }
    }
}
```

There are three loops, i, j, k, in the above procedure. The loop k is a time consuming task. If q=48 and n=96, there are 4608 actions of 'if((corr&ox800000)==0x800000)' in the procedure. One way to speed up is to use a look-up table. The test in table 4.1 shows that if using a 4 step weighting table instead of a linear taper weighting table, the difference in result between the two is small. The 4 step weighting table implies that a group of XOR bits with the same weighting value can be summed up before the weighting value calculation. The sum of how many '1' bits in a group can be done by using the group of XOR bits as a pointer to a look-up table. The elements in the look-up table are the sum of the '1' bits in each of the pointer. For example, using a look-up table to count the '1' bits in any of 4 bit number. The set of 4 bit number is (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) in decimal or (0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111) in binary. Then a look-up table for counting '1' in any 4 bit number is *table*(0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4) in decimal or *table*(0000, 0001, 0001, 0010, 0001, 0010, 0010, 0011, 0001, 0010, 0010, 0011, 0010, 0011, 0011, 0100) in binary. If looking for the number of '1' bits in 12, the sum is *table*(12)=2. The procedure to use the look-up table is as follows:

one_dimensional_correlation_2(I_g, $M_a(i)$, $L_{corr}(i,j,q)$)

```
{
	weightB()=(1,2,3,4,4,3,2,1);
	for(i=0;i<q;i++) Lcorr(i)=0;
	for(i=0;i<n/24;i++)
	{
		triple_word=(Ig(i)<<48)|(Ig(i+1)<<24)|(Ig(i+2));
		word_map=Ma(i);
		for(j=0;j<q;j++)
		{
			word_image=(triple_word>>48)&0xffffff;
			corr=word_image XOR word_map;
			corr1=(corr>>12)|oxfff;
			Lcorr(j) += Look_up(corr1)×weightB(i×2);
			corr1=corr|oxfff;
			Lcorr(j) += Look_up(corr1)×weightB(i×2+1);
			triple_word<<=1;
		}
	}
}
```

With this approach, the execution time of the tracking reduced to one-sixth of the time for the previous tracking concept. Further optimisation of this procedure is discussed in chapter 7. The *corr* variable contains 24 bit XOR result but the *corr1* is 12 bit. The reasons behind this are 1) the biggest number of a group of XOR result bits with the same weighting value is 12 because the 4 step weighting table is used (if n=96, n/8=12); 2) a single DSP56001 has only 3×64k word memory accessing ability (including program memory, see Motorola 1989). A 24 bit address pointer look-up table has 16,777,216 elements, or 16M words, which is far bigger than the memory capacity of DSP56001, although the 24bit table is faster than the 12bit table. A 12bit look-up table only has 4,096, or 4K words, which is small enough to allow DSP56001 to store the map, image and look-up table within its memory at the same time.

4.5.3 Match in Longitudinal Direction

The idea of how to choose 'wait', 'step' and 'jump' is discussed in section 4.3. In order to describe how to make the forced decisions, two variables '*s*' and '*t*' are introduced. The *s* is the sum of three consecutive X-dimension matching actions, that is current, previous and one before the previous. The *t* is the sum of two previous X-dimension matching actions. Before the forced decision the current action results from the best matching of three one-dimensional correlations (see section 4.3.1 and 4.5.2). All possibilities of three consecutive X-dimension matching actions before a force decision are listed in table 4.4. Then the current X-dimension matching action will or will not be changed according to the section 4.3.2. All possibilities of three consecutive matching actions after a forced decision are listed in table 4.5. The values of *c* that have been changed in table 4.5 are highlighted.

c	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
i	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2
p	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
s	0	1	2	1	2	3	2	3	4	1	2	3	2	3	4	3	4	5
t	0	1	2	1	2	3	2	3	4	0	1	2	1	2	3	2	3	4

c	2	2	2	2	2	2	2	2	2
i	0	0	0	1	1	1	2	2	2
p	0	1	2	0	1	2	0	1	2
s	2	3	4	3	4	5	4	5	6
t	0	1	2	1	2	3	2	3	4

Note: c--current increment action;
i--previous increment action;
p--action before i; s=c+i+p; t=i+p.

Table 4.4 Three consecutive actions before forced decision.

Comparing the *c*, *s*, and *t* in table 4.4 and 4.5 the forced decisions on the *c* in table 4.5 are decided by the values of *s* and *t* in table 4.4 as follows:

- if $2 \le s \le 4$, no forced decision is needed;
- when $2 > s$, the forced decision on *c* is determined by $c=2-t$;
- when $4 < s$, the forced decision is determined by $c=4-t$.

If there are two best matchings in the different lines of map, a decision of which one is the matched result should be made. The table 4.6 shows all possibilities of two best matchings. If *best 1* equals *best 2*, the decision of best matching is the *best 1* (see case 1, 5 and 9 in table 4.6). If either *best 1* or *best 2* equals *j+1*, the decision is *j+1* (see case 2, 4, 6, and 8). The rest case 3 and 7 produce *j+2* decision.

c	*2*	*1*	0	*1*	0	0	0	0	0	*2*	1	1	1	1	1	1	1	*0*
i	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2
p	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
s	2	2	2	2	2	3	2	3	4	2	2	3	2	3	4	3	4	4
t	0	1	2	1	2	3	2	3	4	0	1	2	1	2	3	2	3	4

c	2	2	2	2	2	*1*	2	*1*	*0*
i	0	0	0	1	1	1	2	2	2
p	0	1	2	0	1	2	0	1	2
s	2	3	4	3	4	4	4	4	4
t	0	1	2	1	2	3	2	3	4

Note: the value on c row in italics is the result of forced decision.

Table 4.5 Three consecutive actions after forced decision on c.

case	1	2	3	4	5	6	7	8	9
best 1	j	j+1	j+2	j	j+1	j+2	j	j+1	j+2
best 2	j	j	j	j+1	j+1	j+1	j+2	j+2	j+2
decision	j	j+1	j+2	j+1	j+1	j+1	j+2	j+1	j+2

Table 4.6 Combinations of two best matchings.

The implementation is as follows:

choice(*increment, increment_p*, L_{corr}, L_{corr_0}, L_{corr_1}, L_{corr_2})

```
{
    for(i=0;i<q;i++)  Lcorr(i)=0;
    max=Lcorr_0(0);
    max_back=max;
    c=0;
    c_back=c;
    for(i=0;i<q;i++)
    {
        if(Lcorr_0(i)>=max)
        {
            max_back=max;
            max=Lcorr_0(i);
            c_back=c:
            c=0;
        }
        if(Lcorr_1(i)>=max)
        {
            max_back=max;
            max=Lcorr_1(i);
            c_back=c:
            c=1;
        }
        if(Lcorr_2(i)>=max)
        {
            max_back=max;
            max=Lcorr_2(i);
            c_back=c:
            c=2;
        }
    }
    if(max==max_back)
        if(c!=c_back)
```

```
                if((c==1)||(c_back==1)) c=1;
                else c=2;
      s=c+increment+increment_p;
      t=increment+increment_p;
      if(s>4) c=4-t;
      else if(s<2) c=2-t;
      increment_p=increment;
      increment=c;
      if(c==0) for(i=0;i<q;i++) Lcorr(i)=Lcorr_0(i);
      else if(c==1) for(i=0;i<q;i++) Lcorr(i)=Lcorr_1(i);
          else for(i=0;i<q;i++) Lcorr(i)=Lcorr_2(i);
      }
}
```

4.6 CONCLUDING REMARKS

The off-line tests described here have enabled suitable values of the IIR β coefficient and centre-weighting function for line matching to be investigated. The results were sufficiently encouraging for the algorithm to be transferred to DSP based hardware for real-time tracking tests.

For the experimental patterned surfaces tested the algorithm is capable of dealing with degrees of lateral distortion of more than 10% and even greater longitudinal distortions.

The algorithm can cope with an arbitrary geometric relationship between a patterned surface and the path to be followed across it.

The foregoing properties make the algorithm suitable for centre-cutting and scalloping textile lace. It is envisaged that the algorithm can be used in other applications in which a continuous pattern in one direction can be found.

The minimised computational requirements of the algorithm described make it suitable for use in the real-time tracking of high resolution images and also make it possible to be implemented in relatively simple correlation type hardware to achieve higher processing speed (see chapter 7).

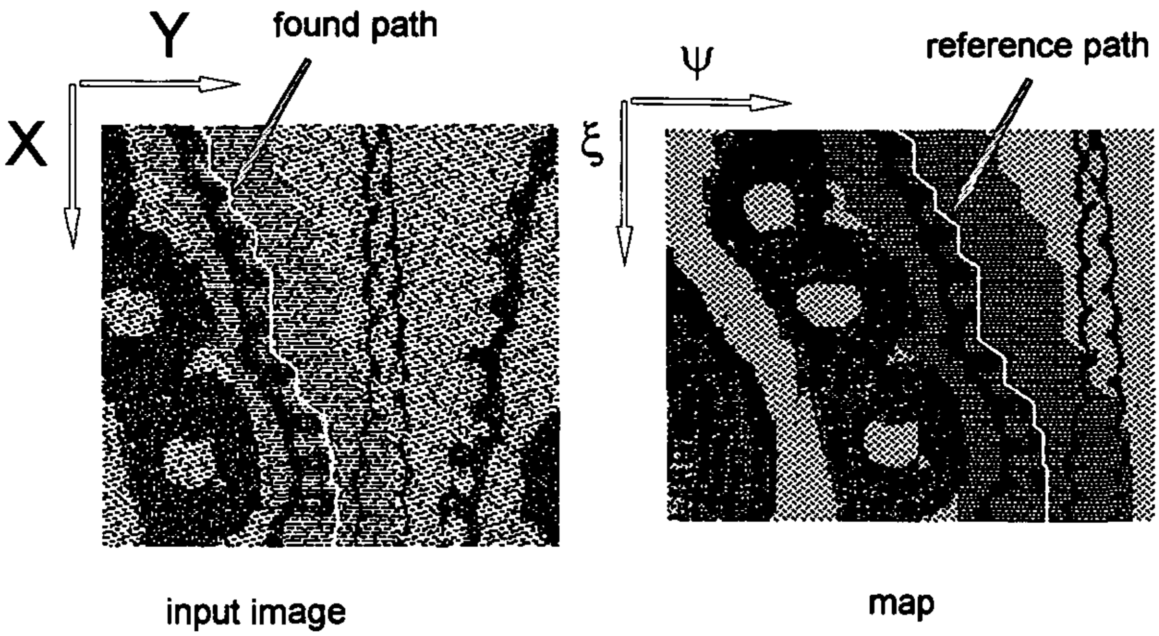


Figure 4.1 Coordinate system of image and map

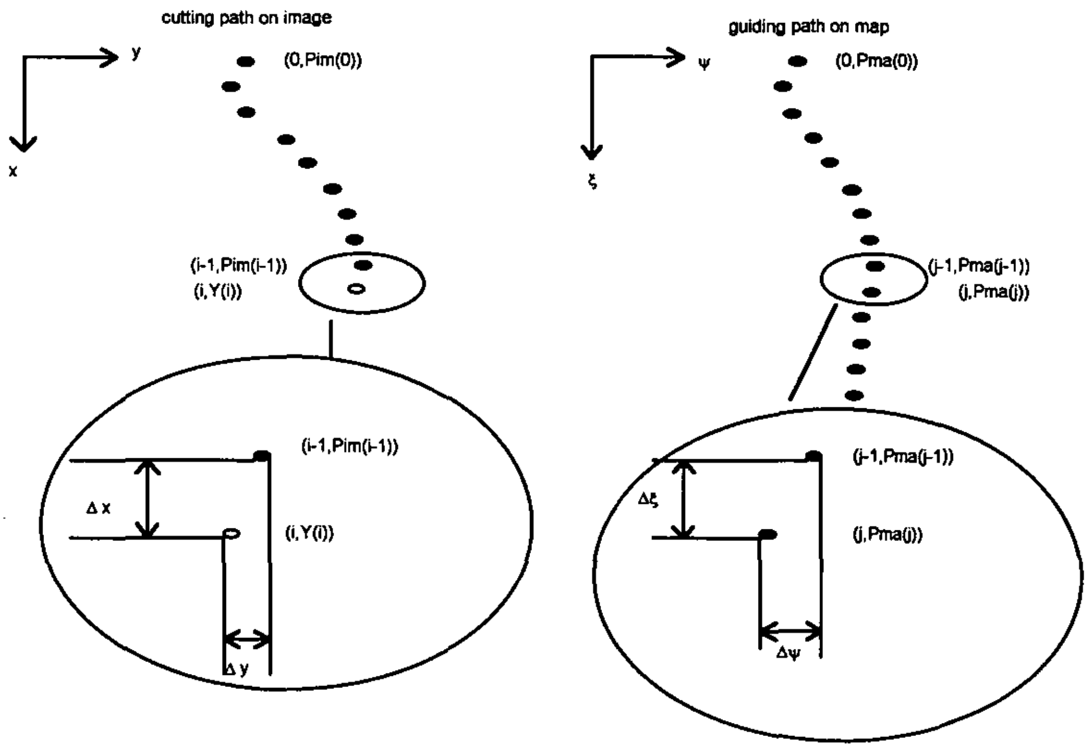


Figure 4.2 Prediction of *Y(i)*

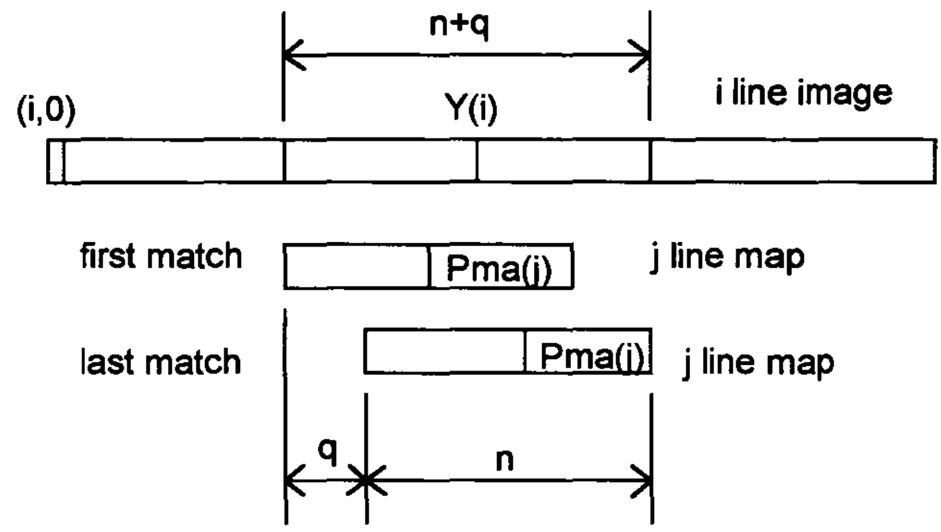


Figure 4.3 One-dimentional cross-correlation

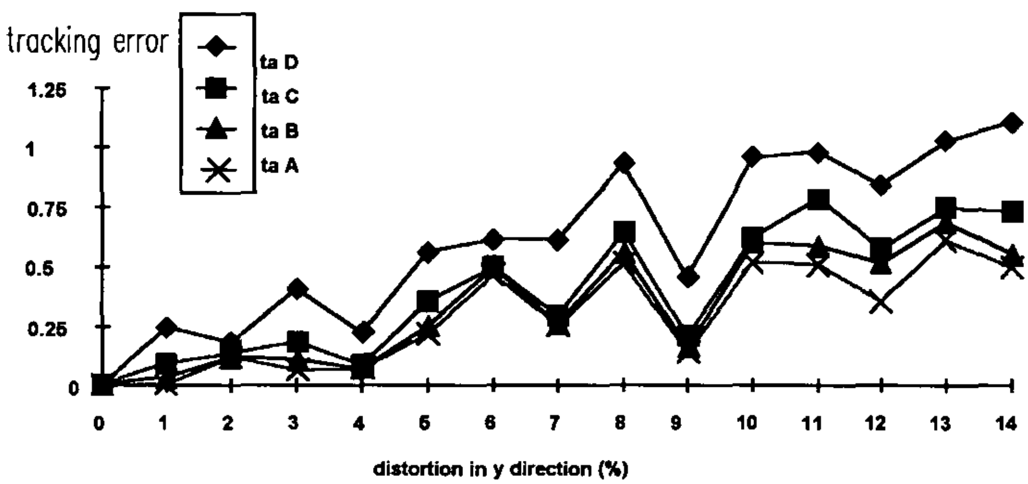


Figure 4.4 Tracking error σ^2 (pixels^2) with different weighting tables

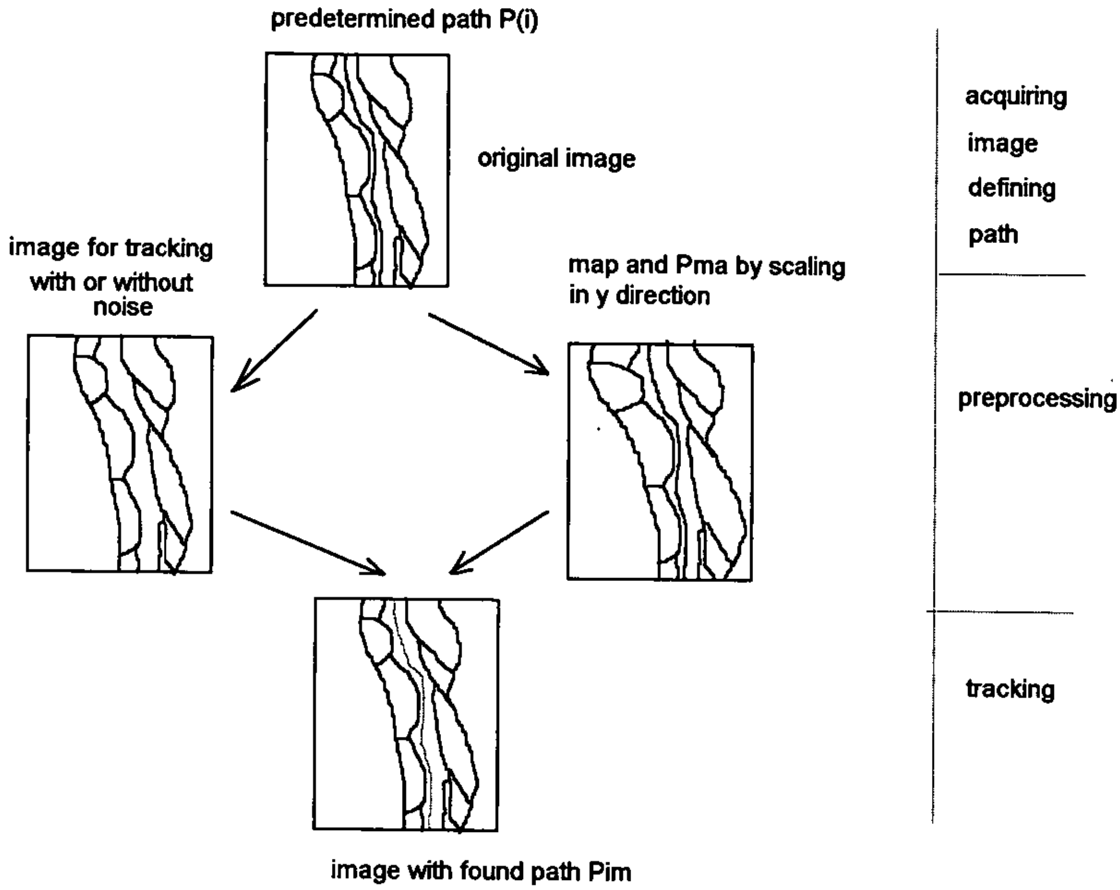


Figure 4.5 Test procedure

Figure 4.6 The map and predetermined path

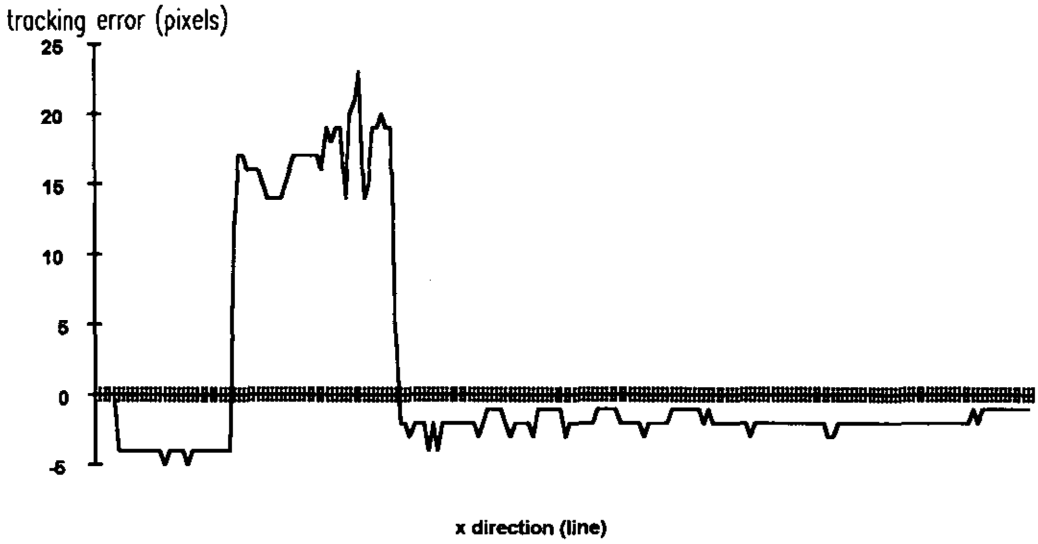


Figure 4.7a Tracking error across a hole with β=0

Figure 4.7b β=0, image and cutting path

Figure 4.8a Tracking error cross a hole with β=0.90

Figure 4.8b β=0.9, image and cutting path

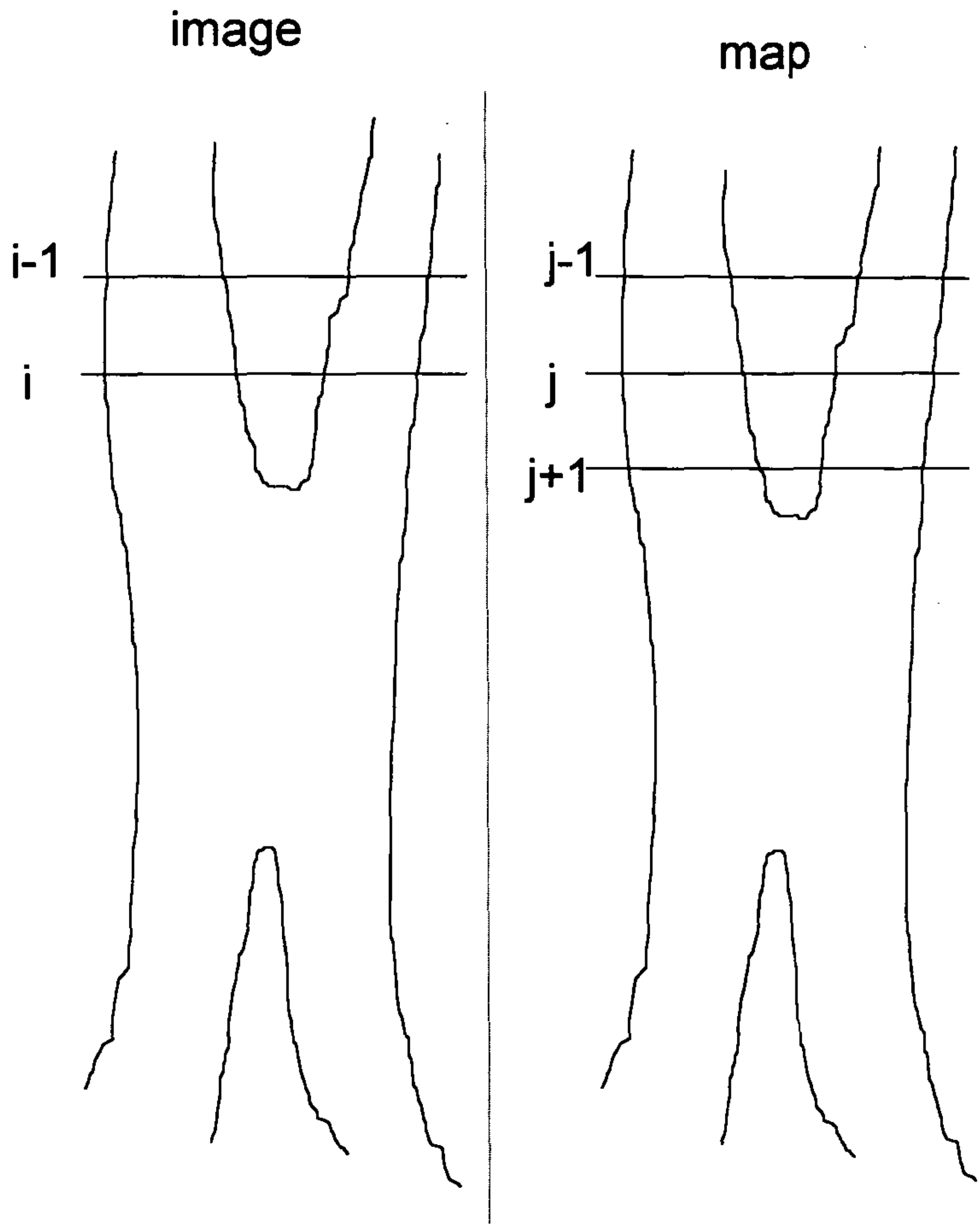
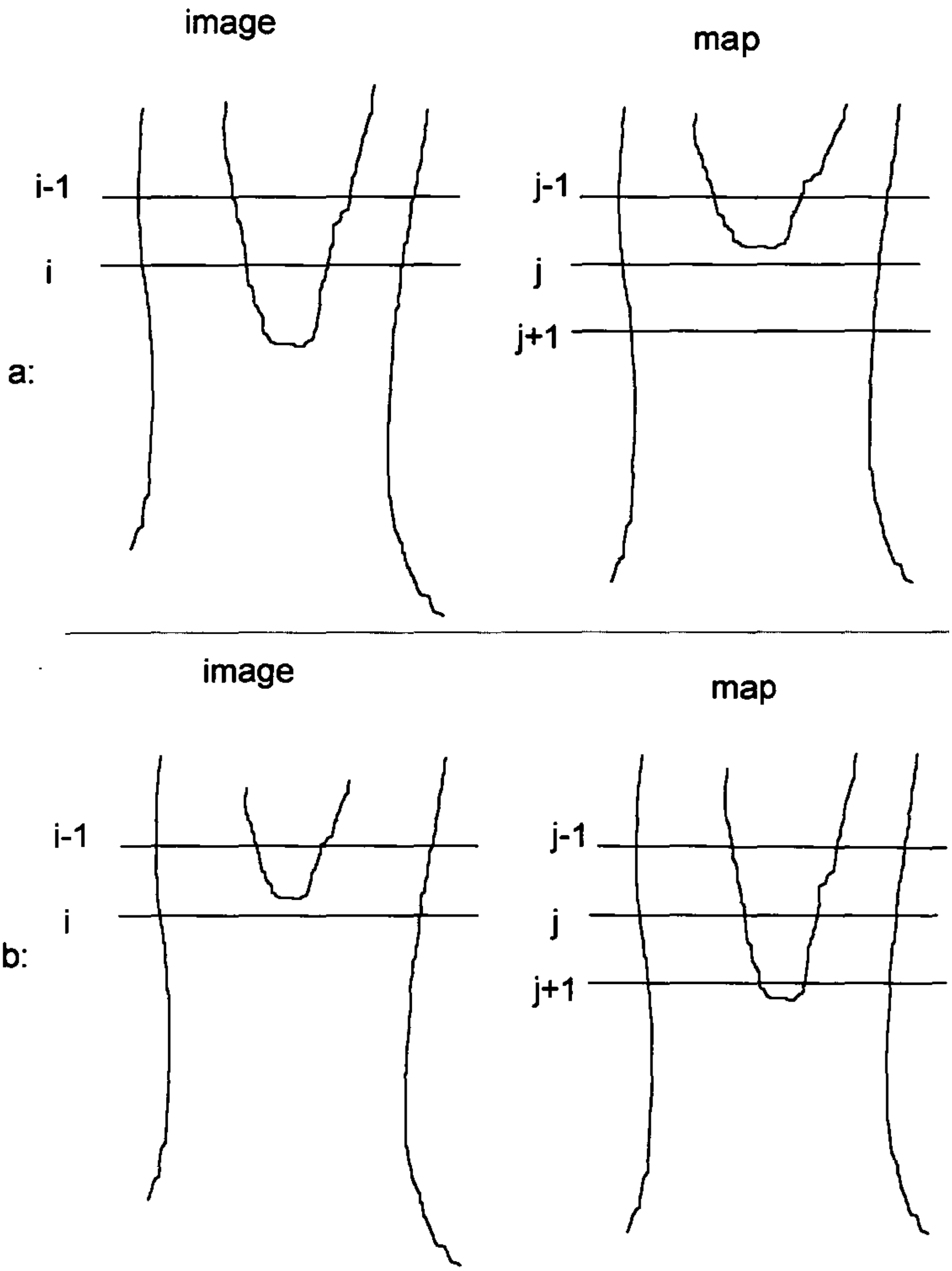


Figure 4.9 One line of incoming image matching with three consecutive lines of the map



a) Image longer than map
b) Image shorter than map
Figure 4.10 Image matching maps with different length

action	image line	matched map line
	i-1	j-1
step	i	j
wait	i+1	j
wait	i+2	j
intervals	3	1

Figure 4.11 Two 'waits' in succession

action	image line	matched map line
	i-1	j-1
step	i	j
wait	i+1	j
forced step	i+2	j+1
intervals	3	2

Figure 4.12 Two 'waits' changing to 'wait - step'

action	image line	matched map line
	i-1	j-1
step	i	j
jump	i+1	j+2
jump	i+2	j+4
intervals	3	5

Figure 4.13 Two 'jumps' in succession

action	image line	matched map line
	i-1	j-1
step		
	i	j
jump		
	i+1	j+2
forced step		
	i+2	j+3
intervals	3	4

Figure 4.14 Two 'jumps' changing to 'jump - step'

action	image line	matched map line
	i-1	j-1
jump		
	i	j+1
wait		
	i+1	j+1
wait		
	i+2	j+1
intervals	3	2

Figure 4.15 Two 'waits' after jump

action	image line	matched map line
	i-1	j-1
wait		
	i	j-1
jump		
	i+1	j+1
jump		
	i+2	j+3
intervals	3	4

Figure 4.16 Two 'jumps' after wait

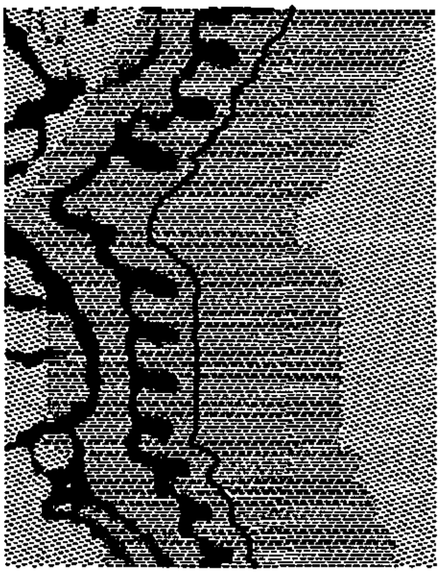


Figure 4.17 Lace test pattern 1 (scalloping)

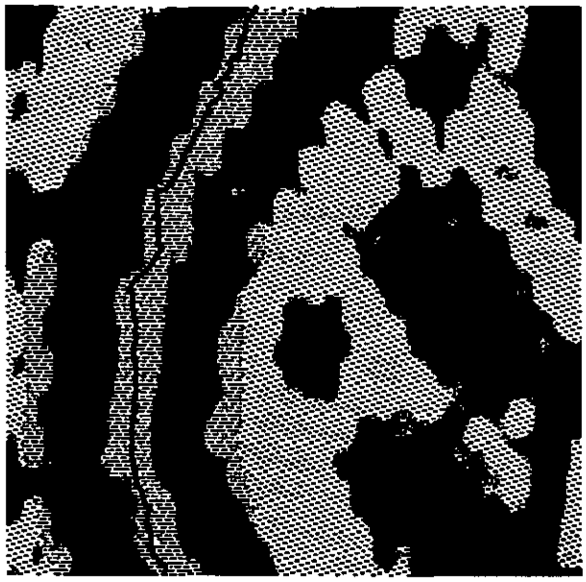


Figure 4.18 Lace test pattern 2 (centre-cutting)

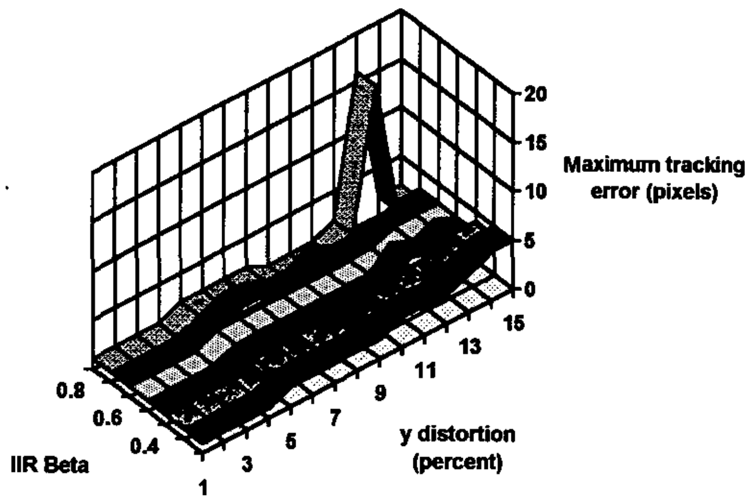


Figure 4.19 Maximum tracking error as a function of pattern y-distortion and IIR filter coefficient β for pattern p1(test conditions as given in column 1 of Table 4.3)

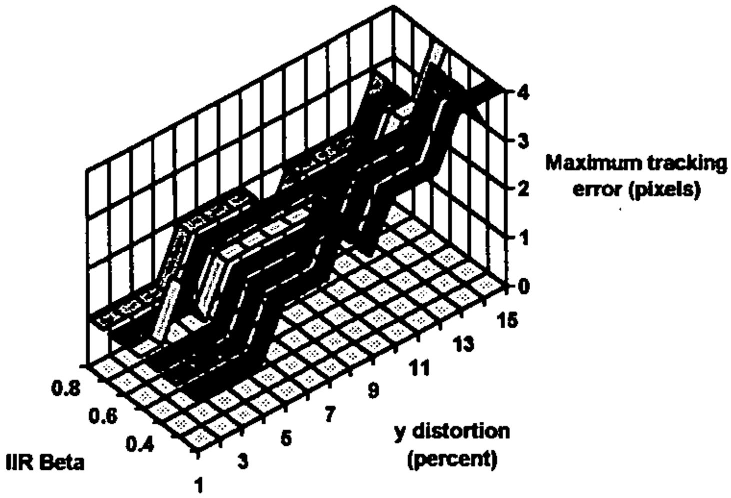


Figure 4.20 Maximum tracking error as a function of pattern y-distortion and IIR filter coefficient β for pattern p1 with 2% noise (test conditions as given in column 1 of Table 4.3)

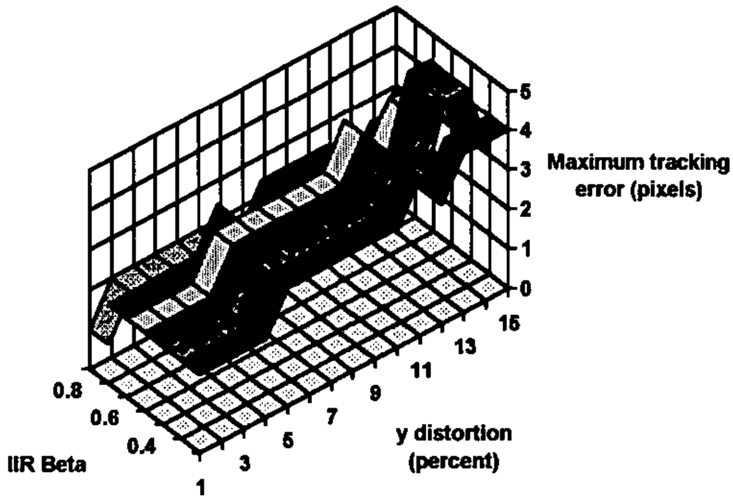


Figure 4.21 Maximum tracking error as a function of pattern y-distortion and IIR filter coefficient β for pattern p2 (test conditions as given in column 2 of Table 4.3)

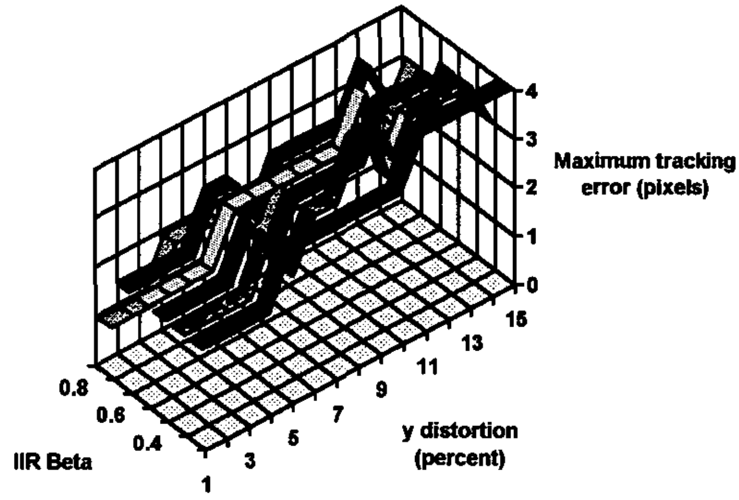


Figure 4.22 Maximum tracking error as a function of pattern y-distortion and IIR filter coefficient β for pattern p2 with 2% noise (test conditions as given in column 2 of Table 4.3)

Chapter 5

METHODS TO FIND THE START POINT

5.1 INTRODUCTION

Finding the start point is basically to find a position on the map corresponding to with the first part of the incoming image, this registering the two images, before the beginning of further tracking. The basic approach to finding the start point is to find desired features on the incoming image.

Feature finding (or segmentation) is one of the basic topics of computer vision. The features to be found on the detected image differ from the model features on the reference image for various reasons such as acquisition errors, different distortions of the material when images are taken. If the differences are much smaller than the dimensions of the features, they can be ignored as is typically the case with interchangeable rigid mechanical components in mass-production. When applied to this kind of subject, feature recognition can be carried out without considering the effect of deformation of the materials. The techniques of feature recognition in this area have been well developed and are now the subject of established academic texts (Davies 1990). There are, however, other situations in which the differences from the reference model are too big to be ignored because the materials on which the features are to be recognised are flexible or elastic. For instance, the appearance of features on knitted fabric varies considerably with knitting tension and level of stretch at the time of imaging, and the human organs on computer tomography charts differ because of the different postures of the patient.

In the situation where the acquired image may differ by rotation, scale and translation (RST) from the reference, solutions have been presented based on invariance kernels. Rubinstein et. al. (Rubinstein 1990) describe a general theoretical approach to distorted or deformed pattern recognition. They show how invariance kernels can be obtained for distortions generated by Lie transformation groups. The most important invariance kernels they report (for rotation and dilation generators) they call the Mellin-Fourier kernel, because it is based on Mellin and Fourier transforms. A practical pattern recognition implementation based on Mellin and Fourier transforms, has been reported by

Hall and Matias (Hall 1993). They attempted to recognise images of viruses, with variable RST, using a template matching algorithm implemented on a powerful array of Transputers (27 T800 devices). Matching of small objects such as keys in their 256x256 pixel images was successfully achieved quickly (around 10 seconds) when the image quality was good, but the noisy images of real viruses could not be distinguished from one another.

A different approach to registering images with varying RST has been presented by Zheng and Chellappa (Zheng 1993) motivated by a requirement for image processing based wind velocity measurement for the Mars '94 project. They have successfully registered aerial images from a balloon mounted camera. Rotation is first estimated by examining illuminant direction information. A small number of feature points are then located, based on a Gabor wavelet model for detecting discontinuities in local curvatures. These are pairwise matched to provide estimates of scale and translation. More accurate estimates of RST are then obtained by hierarchical (coarse to fine) correlation matching.

Distorted feature finding has also been reported by Burr (Burr 1981) who makes use of a 'rubber sheet transform' to find the matched feature pair. The application of this method has been reported by Moshfeghi (Moshfeghi 1991). All edges of the features on an image can be represented by polygonal or polynomial approximations. Similarities between the approximate representations on the images of the reference object and deformed object can then be checked and the features on the image of the deformed object can be detected. This method seems sensitive to noise (Maitre 1989).

The Hough transform offers an alternative approach to feature finding. In principle, the Hough transform is a statistical method of recognizing features. The ordinary Hough transform describes a feature with an analytical model, as in the example of line detection (Hough 1963); or with a list of boundary points that make up a template, as in the generalized version (referred to as GHT) (Ballard 1981). Then, each image point of interest transforms to a parameter space (e.g. line parameters used in such space for line detection) to make an occurrence ('vote') on every possible position according to the model description. A further analysis of the parameter space is employed to find the points in the parameter space which have the highest votes and at which there is therefore the greatest likelihood of locating a desired feature. The positions and the number of features on the image can thus be decided. This method can cope with some occlusions if the vote on the desired position is still higher than the vote on other undesirable areas.

Normally the Hough transform is applied to recognizing a feature on a non-deformable material in which the model of the feature is defined as a set of boundary points or a set of boundary formulae (referred to as a Hough transform with a boundary model). Examples have been reported in biscuit inspection (Davies 1984) and IC inspection (Albanesi 1990). If the Hough transform is directly applied to feature recognition on deformable patterned materials it will fail, because the relationship of the boundary points on the acquired image established by the boundary model does not stay the same for differing degrees of stretch in the material.

In section 5.2 a sub-feature model for Hough transform is proposed and studied. In section 5.3 another method of finding the start point based on the tracking algorithm concept of chapter 4 is investigated.

5.2 MODIFIED HOUGH TRANSFORM FOR FINDING THE START POINT

To cope with feature recognition on deformable patterned materials a sub-feature model is therefore introduced to the Hough transform.

5.2.1 Sub-Feature Model for Hough Transform

Consider an expected feature on an image where the feature is composed of a set of contiguous elements, such as might be the case for a flower on a textile fabric. The feature is divided into several sub-features according to the similarity of texture on a relatively small area. If binary images are considered, a sub-feature will be a small area with the same feature character, black or white.

The sub-feature can be described by a set of pixels within a predetermined area of the feature, *s(i)*. The *i* is the number of sub-feature. If one point on each sub-feature is chosen as a reference for vectors between the sub-features, the vectors and the sets of pixels for every sub-feature can be used to describe the feature. If *i* and *j* are the numbers of the sub-features for a feature, the vector *v(i,j)* represents the geometric relationship between sub-features *i* and *j*. A feature can now be described by sub-features *s(i)* and the vectors *v(i,j)*.

For feature recognition on non-deformable materials the applications of the Hough transform with the sub-feature model gives the same results as the Hough transform with the boundary model. Suppose that there is an image with a feature of interest; the sub-feature model for the feature from the sample image can be made as above, but the reference points for the vectors on the sub-feature are chosen to be on the edge of each feature. The sub-feature model is now equal to the boundary model and the vectors *v(i,j)* can be easily changed into a reference table. However, it is not significantly different if the reference points for vectors are in the middle of the sub-features because the geometric relationship between the sub-features of the feature on the detected image remains the same as, or very similar to that on the sample image. Figure 5.1 is an example of using the sub-feature model Hough transform to detect a feature without any distortion. Figure 5.1a shows the feature which is described by vectors and sub-features, Figure 5.1b is the input image with the feature described in Figure 5.1a and Figure 5.1c is the result after transformation.

The feature on the reference sample is usually different from the same feature on the image detected later because the image for the reference sample and for detection are from different parts of the deformable material or under different conditions. The vectors *v(i,j)* and sets of pixels *s(i)* are used to describe the model of the feature on the sample image. The vectors *v'(i,j)* and sets of pixels *s'(i)* are used to represent the corresponding feature on the detected image. The relationships between *v(i,j)* and *v'(i,j)* can be written as

$$v'(i,j) = v(i,j) + Dv(i,j) \qquad \text{eq.5.1}$$

and the relationship between any *s(i)* and related *s'(i)* will be

$$s'(i) = s(i) + Ds(i) \qquad \text{eq.5.2}$$

The definition of *Dv(i,j)* and *Ds(i)* can be used for either a 'stretched' or 'squeezed' case. The effect of the *Dv(i,j)* and *Ds(i)* on the detection of the feature on the deformable material is discussed here. The effect of the *Dv(i,j)* on the detection of the feature is considered first. Suppose the points on every sub-feature for vectors are in the centres of the sub-features. If *Dv(i,j)* is small and the *v(i,j)* still points to a pixel within the *s(j)*, some pixels of the set of pixels *s(j)* on the detected image can still make a contribution to the vote on the parameter space (Figure 5.2). The feature still can be detected, although the position may change slightly. If *Dv(i,j)* is too big and causes the pixel pointed to by *v(i,j)* to be outside the *s(j)*, the votes in the expected area will decrease. If the votes are the same as or below the levels of other undesired areas, this method will fail. It is envisaged that the effects of *Ds(i)* on the feature finding results would be similar to the effect of the *Dv(i,j)* from the 'vote' point of view. Because usually only boundary pixels are interesting for Hough transformation with the boundary model the parameter space in the GHT form is determined by equation 5.3:

$$A(i,j) = \sum_{k \in \text{all boundaries}} \alpha_k(i,j) \qquad \text{eq.5.3}$$

The $\alpha_K(i,j)$ is the sub-parameter space for pixel k according to the reference table. The parameter space *A(i,j)* will store the final votes for analysis later. This saves a lot of computing time.

When the Hough transform with sub-feature model is used, all image pixels are proposed to be used, including the edge pixels, to find features with a certain degree of deformation or elasticity (see equation 5.4).

$$A(i,j) = \sum_{k \in \text{all sub-features}} \alpha_k(i,j) \qquad \text{eq. 5.4}$$

As proposed in (Merlin 1975), equation 5.4 can be re-written into equation 5.5.

$$A(i,j) = \sum_{all\ directions} image(i-x, j-y) \qquad \text{eq.5.5}$$

The *image(i-x,j-y)* is the image shifted by the search direction *(x,y)*. In this way, the Hough transform is divided into a sequence of movements of the image and addition of the moved image into a parameter space. The result of this sequence change is that more information than just the boundary can be used to determine the point on the parameter space with maximum vote.

5.2.2 Building a Vector Table for a Set of Sub-Features

The procedure to build a vector table is as follows:

1. Choose (manually) the desired feature from a sample.
2. Select a set of sub-features from the chosen feature. All pixels inside each sub-feature have the same image property, 1 or 0 for the binary images considered.
3. The vector table contains the coordinates of the centre and image property for each of the sub-features.

Selecting the sub-feature to be used to represent the chosen feature is a form of discreteness of the image at a larger-than-pixel scale. This needs to be done in such a way that the distinctive characteristics of the feature are suitably sampled but in such a way that distortion of the image tends not to lead to changes in the image property of the sub-features. This suggests that the sub-features should be clusters of pixels of like property, but that they should be chosen throughout the feature to straddle boundaries.

Two methods have been used to select sets of sub-features for a chosen feature. A manual method was first employed for preliminary evaluation of the modified Hough transform approach. It was reasoned that if the approach could not be made to work with intelligent operator judgement in sub-feature selection stage there would be little point in developing an automatic sub-feature selection algorithm. Results using this manual approach were encouraging and so an appropriate automated approach was developed.

In the manual approach the operator chooses points that are in the centres of groups of pixels with the same image property. Figure 5.3 shows an example of a selected 'T-shaped void' feature represented by manually chosen sub-features. Figure 5.3a shows the feature and Figure 5.3b the set of selected sub-features. Table 5.1, gives the x and y pixel co-ordinates (from the nominal feature centroid) and the image property for the sub-features.

43 Sub-feature co-ordinates in pixels relative to centre of feature (x, y, property)
(-48, 7, 0), (-39, -8, 0), (-39, 5, 1), (-39, 15, 0), (-32, 20, 0), (-32, 6, 1), (-32, -16, 0), (-22, -20, 0), (-22, -8, 1), (-22, 6, 1), (-22, 17, 1), (-22, 34, 0), (-22, 43, 0), (-12, 45, 0), (-12, 34, 1), (-12, 23, 1), (-12, 5, 1), (-12, -10, 1), (-12, -24, 0), (0, -32, 0), (0, -17, 1), (0, -5, 1), (0, 7, 1), (0, 20, 1), (0, 32, 1), (0, 45, 0), (9, 34, 0), (9, 12, 1), (9, -9, 1), (9, -21, 1), (9, -32, 0), (21, -32, 0), (21, -22, 1), (21, -13, 1), (21, 1, 0), (33, -6, 0), (33, -20, 1), (33, -31, 0), (45, -28, 0), (45, -21, 1), (45, -11, 0), (53, -11, 0), (53, -22, 1)

Table 5.1 'T-shaped' vector for sub-feature

The automatic sub-feature selection method was implemented as follows:

1. Decide the boundary of the feature, (this was performed manually for these tests).
2. Calculate two lines parallel to the boundary, one inside it and one outside.
3. Select centre of a sub-feature, which is n*n pixels. Ideally all pixels inside a sub-feature should have the same image property, but practically some random noise (such as electronic noise) should be accepted. 90% pixels inside of the sub-feature are required the same.

An illustration of sub-feature selection by this method is given in Figure 5.4 (and Table 5.2). Figure 5.4a shows the feature selected from the sample. Figure 5.4b is the boundary for the selected feature. Figure 5.4c shows the two lines parallel with the boundary. The feature in Figure 5.4a is represented by the set of the sub-features shown in Figure 5.4d. Figure 5.4e is a mirrored set of the sub-features in Figure 5.4d which are used to search for a mirrored feature in the input image. Table 5.2 is the vector table for the set of sub-features in Figure 5.4e.

123 Sub-feature co-ordinates in pixels relative to centre of feature (x, y, property)
(-78, 13, 0), (-77, 22, 0), (-69, 25, 0), (-61, 28, 0), (-52, 28, 0), (-43, 29, 0),
(-34, 31, 0), (-27, 36, 1), (-21, 42, 0), (-4, 47, 1), (5, 46, 0), (14, 45, 0),
(23, 45, 0), (32, 41, 0), (41, 37, 0), (50, 35, 0), (59, 34, 0), (67, 29, 0),
(61, 24, 0), (54, 19, 1), (63, 14, 0), (72, 10, 0), (79, 3, 0), (86, -3, 0), (78, -4, 0),
(76, -13, 0), (69, -15, 0), (62, -16, 0), (56, -12, 0), (51, -8, 0), (45, -25, 0),
(41, -32, 0), (27, -36, 1), (20, -39, 0), (13, -40, 0), (6, -40, 0), (-1, -40, 0),
(-22, -42, 0), (-29, -42, 0), (-36, -41, 0), (-41, -37, 0), (-55, -34, 0), (-60, -30, 0),
(-60, -23, 0), (-60, -16, 0), (-67, -15, 0), (-74, -13, 0), (-81, -14, 0), (-88, -15, 0),
(-85, -8, 0), (-82, -1, 0), (-80, 6, 0), (-92, 33, 1), (-83, 36, 1), (-75, 39, 1),
(-67, 42, 1), (-60, 46, 1), (-51, 47, 1), (-42, 49, 1), (-35, 54, 1), (-26, 56, 1),
(-18, 59, 1), (-10, 62, 1), (-2, 66, 1), (7, 65, 1), (17, 64, 1), (26, 63, 1), (35, 63, 1),
(43, 58, 1), (51, 53, 1), (60, 52, 1), (69, 47, 1), (78, 49, 1), (87, 50, 1), (97, 49, 1),
(106, 48, 1), (109, 39, 1), (111, 30, 1), (104, 29, 1), (97, 24, 1), (90, 21, 1),
(95, 13, 1), (102, 6, 1), (101, -3, 1), (100, -12, 1), (97, -20, 1), (91, -26, 1),
(85, -31, 1), (80, -38, 0), (73, -39, 0), (62, -48, 1), (55, -50, 1), (48, -52, 1),
(41, -53, 1), (33, -54, 1), (26, -56, 1), (19, -57, 1), (12, -61, 1), (5, -64, 1),
(1, -58, 1), (-6, -57, 1), (-13, -59, 1), (-20, -60, 1), (-28, -61, 1), (-35, -60, 1),
(-42, -58, 1), (-48, -55, 1), (-55, -56, 1), (-61, -52, 1), (-66, -48, 1), (-71, -44, 1),
(-74, -38, 1), (-78, -32, 1), (-85, -33, 1), (-93, -34, 1), (-100, -35, 1), (-107, -28, 1),
(-105, -14, 1), (-102, -7, 1), (-99, 0, 1), (-97, 7, 1), (-95, 14, 1), (-93, 21, 1)

Table 5.2 'Flower' vector for sub-feature

5.2.3 Experimental Tests of Modified Hough Transform

The tests were carried out on a lace samples under the test conditions shown in table 5.3.

The lace material was stretched manually in the longitudinal direction (this naturally also caused a degree of contraction in the lateral direction) by using test rig 1 (see section 3.7.1). The reference, Image 1, and image 2 were taken from a normal fabric sample with approximately 10.5 degree of pattern skew and nearly 10% shorter (skew and shortness caused in manufacturing) than the normal sample in its 0% stretch condition. The Hough transform with sub-feature model was then applied to the images. The data used to describe features 'T shaped void' and 'flower' were taken from the reference image and applied to all images.

Reference	0% stretch	no skew
Image1	10% stretch	no skew
Image2	20% stretch	no skew
Image3	0% stretch	skewed
Image4	10% stretch	skewed
Image5	20% stretch	skewed
Image6	30% stretch	skewed

Table 5.3 Test conditions for Hough transform

The results of detecting a feature described as 'T-shaped void' are given in table 5.4.

Image	Reference	Image1	Image2	Image3
First point	x= 96 y=109	x=171 y=115	x=181 y=118	more than one
Second point	x=390 y=110	x=507 y=117	x=540 y=113	more than one
distance between 2 points	294	336	359	-
stretch estimated from the image analysis	0%	14%	22%	-

Table 5.4 Feature detecting results for 'T-shaped' feature

The results of detecting a 'flower' feature are in table 5.5.

The Turbo C programming language was used to implement the algorithm on a PC compatible computer with a 20MHz 80386 processor. The test images were binary image of 640×350 pixels and were acquired with a 'frame grabber' (see section 3.6.1 vision system one). The average computation time for the 'flower' feature was about 7min 30s.

Image	Reference image	Image1	Image2	Image3	Image4	Image5	Image6
first point	x=197 y=133	x=111 y=265	x=107 y=266	x=156 y=244	x=136 y=240	x=131 y=239	x=168 y=235
second points	x=493 y=132	x=448 y=265	x=468 y=264	x=414 y=245	x=413 y=233	x=429 y=234	x=491 y=235
distance between 2 points	296	337	351	258	277	298	323
stretch to Reference	0%	14%	19%	-13%	-6%	0%	9%
stretch to Image 3	-	-	-	0%	7%	16%	25%

Table 5.5 Feature detecting results of 'Flower'

5.2.4 Analysis of Tests

As Table 5.3, Table 5.4 and Table 5.5 show, the experimentally produced distortions of the materials and those found using the detection algorithm agree quite closely. The number of features found is the same as the number judged visually, and the test results using the 'flower' data show that this method can find a feature on an image with 20% stretch even when skewed. The results are sufficiently encouraging to warrant transferring this algorithm to high speed processing hardware.

The differences of distances obtained on Image1-Image2 between 'T-shaped voids' and 'flowers' are mainly due to the statistical principle of the Hough transform and differing sample data. Methods of improving the feature locating accuracy would be an interesting topic for further research.

The tests show that the more data chosen, the wider range of elastic distortion can be dealt with. The 'flower' data can cope with a wider range than the 'T-shaped' data. The small scale and skew of Image3, and fewer data points are the main reasons why the 'T-shaped' data finds false features along with the correct features to match Image3 while the same data can deal with Image1-

Image2 successfully. But the results of matches may still be useful for identifying features that are to be further verified.

5.2.5 Possibility to Implement Hough Transform

The target vision system to implement Hough transform for finding the start point is vision system 2 or vision system 3 (see Figure 3.9 or 3.10) in which a host computer based on Motorola 68020 has 4M byte memory and an embedded multiple Motorola DSP56001 board has 2×32K word (3bytes) private data memory for each DSP and has 2×32K word shared data memory for all DSPs and host computer. There are two possible ways to implement the Hough transform for finding the start point in vision system 2 or vision system 3. One is to be implemented on DSP; another is on host computer.

The major problem of implementation on the DSP is that the data memory on the DSP is not big enough to form a parameter space $A(i,j)$ (see section 5.2.1). The images for the tests on the PC are 640×350 pixels which require 224,000 byte memory as the parameter space. The size of real image must be bigger than the size of the test image on PC, which is restricted by the 'MicroEye' frame grabber board. If the resolution of real image taken by the line scan camera is 0.3mm/pixel and the view size of lace is 300×200mm, the image size is 1000×666 pixels and the size of parameter space will be 666,000 byte which is far larger than the variable memory for DSP56001.

In order to evaluate the implementation on the host computer, the author rewrote a C program combined with 68020 assembly under the OS-9 operating system to test same images tested on PC. The program took about 2min 30sec. to register the desired features which was faster than the program on the PC. The memory requirement of Hough transform is no longer a restriction on this implementation because 68020 has 4G byte memory accessing ability and there is no problem to extend the memory of the host computer up to several Mbyte by adding an extra memory card to host computer. Readers can envisage that for a real cutting machine the beginning a block of image data will transfer from an embedded DSP board to the host computer for finding the start point, then information about the start point will be sent back to a DSP, and the DSP will then start the tracking forwards.

5.3 TRACKING ALGORITHM FOR FINDING THE START POINT

Section 5.2 presented a modified Hough transform method to successfully find corresponding points between the input image and a map image stored in the computer. The results of the Hough transform carried out both on an IBM PC and on a machine based on a Motorola 68020 CPU (which is the host computer for the vision system based on a line scan camera and DSP) to process frame grabbed images were encouraging.

After the tracking algorithm described in chapter 4 had been implemented on one of the Motorola 56001 DSPs; another of the DSPs was used to control a line scan camera and get the image data from the camera. Initially it was assumed that the two different vision tasks, finding the start point and tracking the cutting path, would be carried out on two different CPUs, DSP 56001 and 68020 because of the different memory size requirement. The problem with this arrangement is that the enormous amounts of image data transfer from DSP memory to 68020 memory takes a large amount of time. This correspondingly increases the whole system working time. The second disadvantage is that the DSP for tracking the cutting path can only wait for this information. With this approach the vision system would work, but not efficiently.

The availability of the DSP's suggested an approach in which the start point can be found using DSP's. The concept of using the tracking algorithm to find the start point was thought worth considering. A number of tests were carried out to test this idea and these were in general successful. This method was considered better because a large working space is not needed. The two tasks of tracking the cutting path and finding the start point can be switched very quickly and easily within the DSP because they employ basically the same algorithm.

5.3.1 Principle and Simulation

The preconditions of using the tracking method to find the start point, the confidence level of finding the start point, and simulation of tracking are described to prove how the tracking method can be used to find the start point. The simulation conditions are described at the end of this section.

5.3.1.1 'Finding the Start Point'

In chapter 4 an incremental tracking method has been investigated. One of the assumptions of this method is that before tracking, a corresponding point on the cutting path of the image, the 'start-point', has been found. In reality, the map, a prescanned one pattern length sample, is the same pattern as the image to be processed; the first portion of the image could correspond to one segment within the map. The objective of 'finding the start point' is to determine which position in the map is the most similar to the first portion of the image.

The basic principle in using the tracking algorithm to find a start point is that if the tracking results are measured by their 'mis-tracking' (which will be discussed later), the position on the map, from which the tracking starts and produces the lowest 'mis-tracking', might be the start point. In practice, there are two ways to implement this method. One is that the first line of the input image is used to match against all lines on the map to find a position on the map which has the highest match result for this single line. Then to try to track the cutting path from this position up to a certain number lines of image and calculate the confidence level (this is one of the 'mis-tracking' measurements) for this tracking process. This procedure is then applied to the next match result position to get another confidence level, and so on. After several searches, the position on the map with the highest confidence level is assumed to be the matching position corresponding to the input image. Another implementation is that the first line on the map is assumed to match the first line of the input image and is used as the start point to carry on the tracking for a certain number of lines and the confidence level of this tracking is calculated and stored. Then the second line of the map is used as the start point to carry out the whole process. After the last line of the map has been applied, the map line with maximum confidence level among all results is chosen as the real start point for the real tracking. Obviously the second method needs to search all the map to find the highest confidence level, while the first method (which is two step search) can control the search range for the second step by setting threshold level to the result of the first search. The second method is suitable for use in the DSP because of its simplicity.

5.3.1.2 Preconditions for Using Tracking

If a portion of the map, the most similar to a given portion of image, exists, is it possible to find it by using the tracking algorithm? The answer is 'yes' if the lace to be imaged for finding the start point is put in a position in which the lace with the same pattern has been imaged for making a map originally.

Let us consider that a segment of the first line of the image is the most similar to the segment of the *n*th line of the map around the guiding line. The segment on the image can be described by its centre $cen_{ig}(0)$ and width W_{ig}, and the segment, which is the most similar to the beginning of the image, on the map can be described by its centre $cen_{map}(n)$ and width $Wmap$. The $cen_{map}(n)$ equals to the $Pma(n)$ (equation 4.6). The maximum displacemant q is 2δ. According to the tracking method a segment on the first line of the image, the searching segment, to be used to compare the map will be formed at the centre $P_{ma}(n)$ (that is, also, used to determine the first predicted point $Y(0)$) and width $W_{map}+2\delta$ because no a priori knowledge can be used to predict the centre of this segment. If $W_{ig}=Wmap$ and equation 5.6 is satisfied, the tracking method can certainly find out the centre $cen_{ig}(0)$ of the most desirable segment on the image because the searching segment with centre $cen_{map}(n)$ and width $W_{map}+2\delta$ includes the segment with centre $cen_{ig}(0)$ and width W_{ig} (see Figure 5.7).

$$|cen_{map}(n)-cen_{ig}(0)| \leq \delta \qquad \text{eq. 5.6}$$

In fact, the map and the image might be placed on the same test rig with displacement in x, the lace feeding direction, and in y, the lateral direction. The equation 5.6 tells us that in order to find the displacements in x, y directions between the image and the map by using tracking algorithm, the original position difference of the y direction between the image and map must be smaller than the maximum lag used in the cross-correlation calculation, if the resolution is 4 pixels/mm for example, the original position of the lace for the image can be ±6mm around the original position for the map in y direction. This kind of position accuracy can be easily achieved by aligning the lace to a line on the plate, on which the lace is placed, along the x direction. If we increase the value of δ, it reduces the requirement of the accuracy of the setup position but will increase the computation time.

5.3.1.3 Confidence Level

A confidence level is used to judge the success of finding the start point. The confidence level, which is determined by equation 5.7 and 5.8, is used here to represent the similarity between a portion of the map and the first portion of the image. It can be actually calculated as a weighted cross-correlation which is produced during the tracking procedure (see chapter 4) as

$$confidence\_level(i,j) = Max\{Lcorr(i,j,q)\} \quad q \in (0, 2\delta) \qquad \text{eq. 5.7}$$

Here i is the number of the input image line and j is the number of the map line. The whole confidence level for a certain number of tracking steps should be

$$whole\_confidence\_level(l) = \sum_{i=0}^{n} confidence\_level(i,j) \qquad \text{eq. 5.8}$$

Where n is the number of lines of the image to be tracked. The sum operation is used here because of the equal contribution of each confidence level to whole confidence level(l). Then the start point on the map can be decided

$$start\_point = l' \qquad Max\{whole\_confidence\_level(l)\} \quad l = l' \qquad \text{eq. 5.9}$$

$$\rho = \sum_{i=0}^{n} (Y(i) - Pim(i))^2 \qquad \text{eq. 5.10}$$

Equation 5.10 provides one way to judge the geometry difference ρ between the guiding line on the map and the cutting line on the image. The $Y(i)$ is the predicted cutting point for the ith line of the image and determined by the pervious tracking and the difference between the current guiding line position and previous guiding line position (equation 4.8). The *Pim(i)* is the current cutting position determined by the result of the current tracking action. The difference between *Y(i)* and *Pim(i)* reflects the difference between increment of the cutting line and increment of the guiding line at the current position.

The confidence level of the tracking function will first be judged by equation 5.9, and if there is difficulty, such as there are two positions on the map with the same highest confidence level, then equation 5.10 will be used to choose one position with lower geometry difference ρ between the guiding line on the map and the cutting line on the image.

With the confidence level and the geometry error the performance assessment of the tracking function is similar to an area cross-correlation. In the area cross-correlation the geometry relationship between rows or columns of a template is fixed by the format of the template itself and the geometry relationship in an image is also fixed. When the cross-correlation is used the template is moved around the searching area as a whole and the cross-correlation value is enough to judge the similarity of the image because the geometry relationships in both the template and the image do not change during the search. In tracking action a map, as a template in the cross-correlation, has only fixed geometry relationships between pixels within one line to form a slice of landmarks. The geometry relationships between lines or slices of landmarks are determined by a guiding line. The found path, the cutting curve, in the image determined by the tracking function represents a geometry relationship of the landmarks in the image. The difference between the cutting curve and the guiding line is the geometry error. If the tracking is correct, the geometry errors are quite small (see chapter 4). In finding the start point, the tracking starts from every possible position in the map. When starting from a correct position in the map, the confidence level is high and geometry error is low. If the tracking starts from an incorrect position, the confidence level might be high because the tracking tends to follow a path with the highest historical value (equation 4.13), but the geometry error must be large because the found path has a different shape from the guiding line. If there are several peaks with the same or similar confidence level, the position with the lowest geometry error is the most likely correct position for the start point.

5.3.1.4 Simulation

The simulation of finding the start point by the tracking algorithm was carried out on a PC using Turbo C. The simulation was used to make sure there is one and only one position on the map to match the input image and decide the minimum number of lines necessary on the input image which ensure correct start point to be found with minimum search time. The pattern in Figure 5.8 was used to simulate finding the start point by the tracking algorithm. Figure 5.9 to Figure 5.16 show the confidence levels and errors by varying the number of lines of image to be tracked from 20 to 90. When 20 lines of image are used to be tracked (in Figure 5.9), there are several peaks with similar heights including the correct matched position in the confidence level curve so that there is a danger of picking an incorrect one. As the number of lines of the image to be tracked increases, the height of the confidence level at the correct matching position becomes higher than others at the non-matched positions. When the number of lines of the image is 40 or 50, the confidence level at the correct matched position is 16% to 18% higher than the nearest confidence levels at the non-correct positions. Observation of the tracking error illustrates that the correct matched position has a low error but not the lowest error through the whole map range. This suggest that the confidence level is more relevant to the correct match position, and that if more lines of the image are used in searching for the start point, there is less risk of picking up a incorrect matched position when using the maximum confidence level approach.

The result of the simulation suggests that there is one and only one position on the map to match the input image if a proper number of lines of input image are considered as a group. Forty lines of input image are enough to find the start point on the test patterns. The PC with 50 MHz 486 processor spent about 4 hours for each pattern's simulation.

5.3.2 Modification of Tracking Algorithm to Suit the Requirement of Finding the Start Point

The only modification required of the tracking algorithm is to calculate the confidence level, which is not a normal requirement of the tracking algorithm but is one of the intermediate results. The *whole_confidence_level(l)* can be found by modifying one of the calculations in the former tracking algorithm (equation 4.11 of chapter 4)

$$H(i,j,q) = Lcorr(i,j,q) + \beta \times H(i-1, j-1, q)$$
$$0 \le \beta < 1 \text{ if normal tracking} \qquad \text{eq.5.11}$$
$$\beta = 1 \text{ if start point tracking}$$

Then eq.5.8 is part of the eq.5.11 if *H(i,j,q)* equals the *whole_confidence_level(l)*. Equation 5.11 can serve both normal tracking and start point tracking.

The algorithm is

```
main
{
  if every thing ready
   {
    do
    {  /* finding the start point */
     β=1
     N=40
     for(start=0;start<last_map;start++)
      {
       tracking
      }
     decide start_point
    }
   do
    {  /* normal tracking   */
     β=coefficient
```

```
      length_input_image=N
      start=start_point
      tracking
     }
    }
  else wait
}
```

5.3.3 Experimental Tests of Tracking for Finding the Start Point

Test rig 2 (Figure 3.13) and vision system 2 were used to test the algorithm. Tests were carried out on two different laces. One is a rather dark lace and another a lighter lace with centre cutting. The DSP took about 1 minute to find the start point on the map and the first line of the input image can start from anywhere. Then the DSP changes to the task of tracking the cutting path. The judgement of the test results is carried out visually by displaying the map, predefined path, input image and cutting path.

5.4 CONCLUDING REMARKS

It has been demonstrated that the modified Hough transform method proposed can find corresponding features between a searched image and a reference image to determine the start point on the deformable patterned materials considered. This method is computationally highly parallel, the basic operation being one of moving the input image according to the searching vectors, but the parameter space is large. It is foreseen that this method can be implemented advantageously using DSP or specific graphic processors to find a start point for the pattern cutting if they can access a wide range of memory, or the parameter space can be reduced.

A realistic way to implement the Hough transform is to use Motorola 68020 because of large memory available. It took 2min 30sec faster than the Hough transform method carried out on the PC. A possible way to implement a pattern cutting system is to find the start point on the 68020 and determine cutting path on DSP.

Another more efficient way to find start point is to employ the tracking function on DSP. The method works. It is faster than the Hough transform method. The tasks of tracking a cutting path and finding a start point can switch from one to another quickly and easily because the two tasks use the same algorithm inside the same DSP. The method also saves a lot of time by avoiding image transfer from DSP to 68020 memory which is required if the Hough transformation is used. It is envisaged that on the real lace cutting machine the switch between two tasks may be necessary and a basic operation to cope with some unexpected problems, such as loss of tracking due to major damage to the lace (e.g. big holes).

The confidence for finding the start point may be useful to cope with fault handling. It is evident that when tracking is lost the confidence level reduces. The confidence could be used as an index to monitor the tracking status.

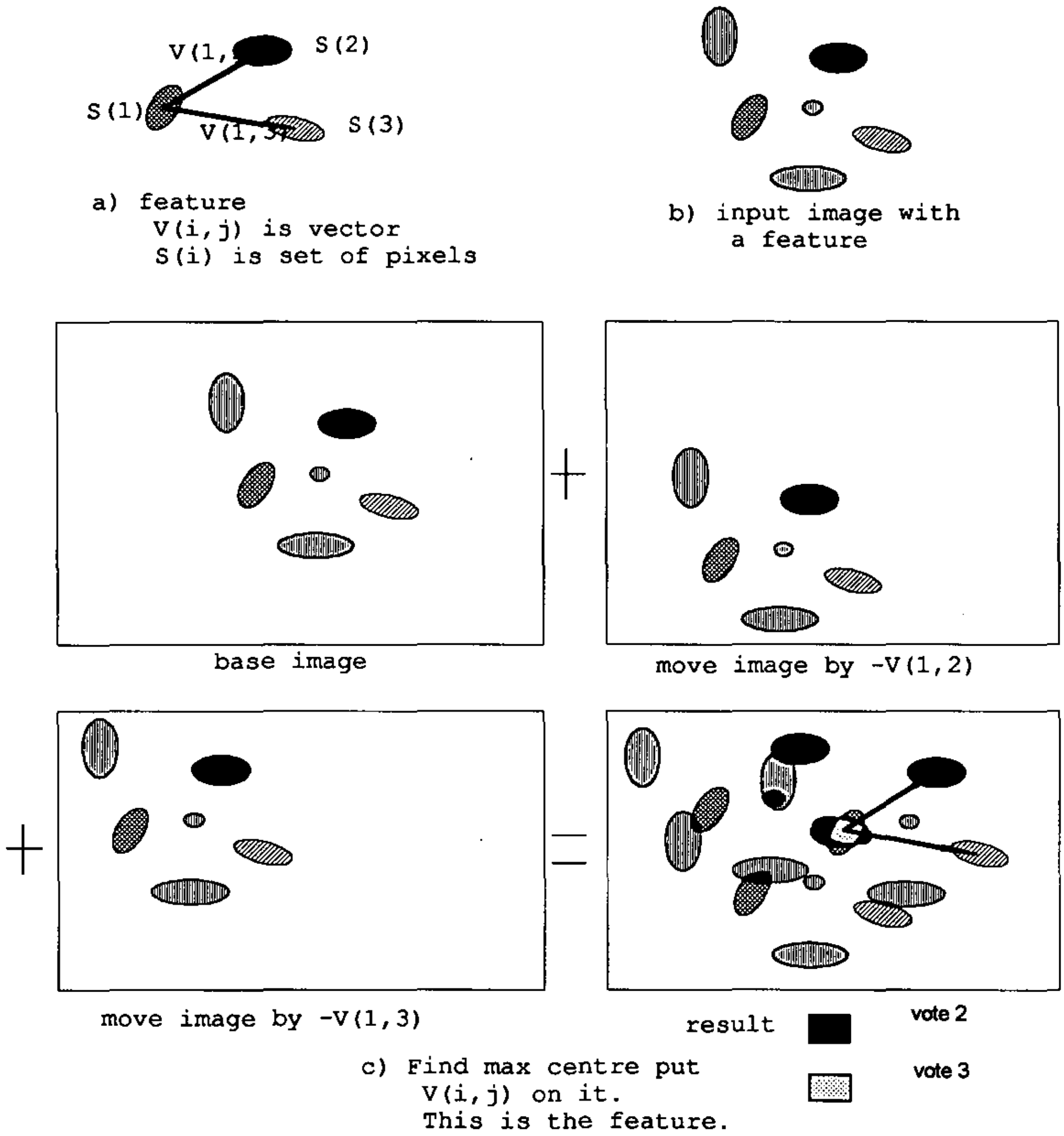


Figure 5.1 Finding a feature by sub-features and vectors

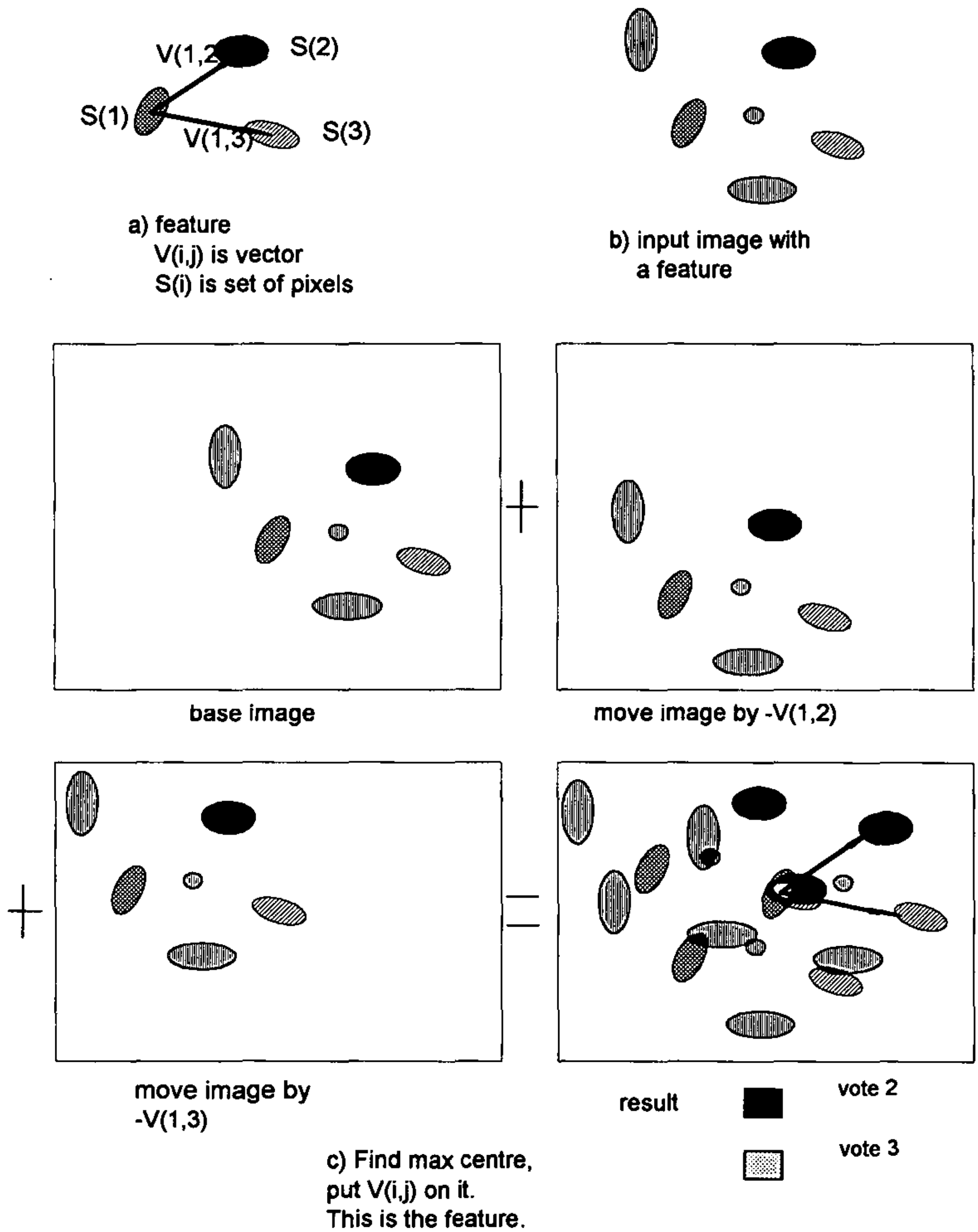


Figure 5.2 Finding a feature in a stretched image by sub-features

Figure 5.3a Selected feature for 'T-shaped' feature

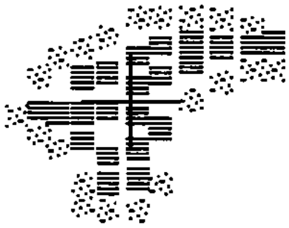


Figure 5.3b Set of sub feature for Figure 5.3a

Figure 5.4a Selected feature for 'flower' feature

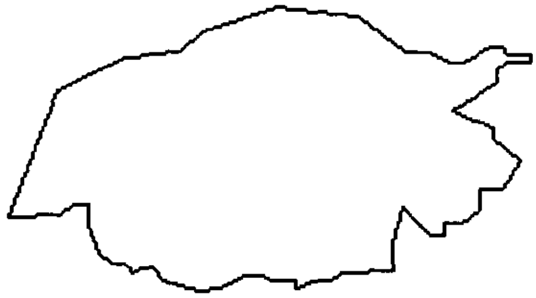


Figure 5.4b Boundary of the feature in Figure 5.4a

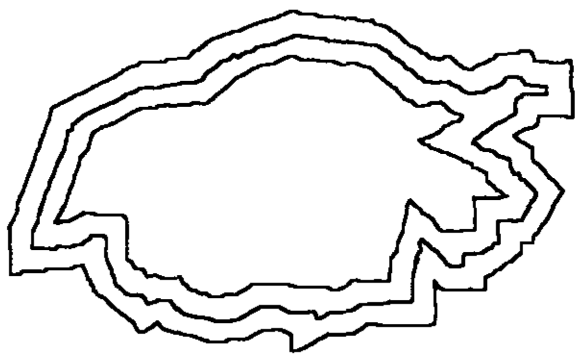


Figure 5.4c Two lines parallel to the boundary of Figure 5.4b

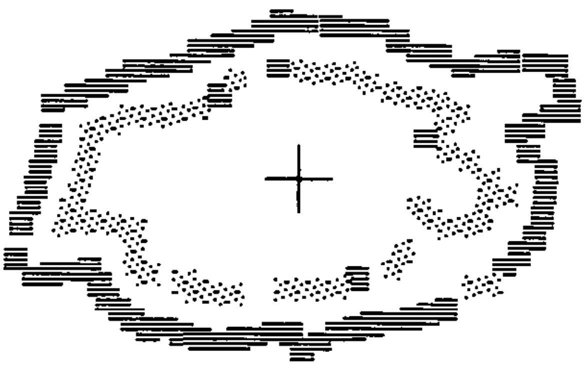


Figure 5.4d Sub-feature of Figure 5.4a

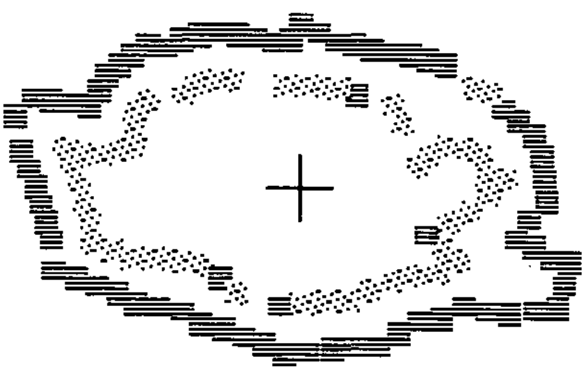


Figure 5.4e Mirrored sub-feature of Figure 5.4a

Figure 5.5 Search result by using a sub-feature of Figure 5.3b

Figure 5.6 Search result by using a sub-feature of Figure 5.4e

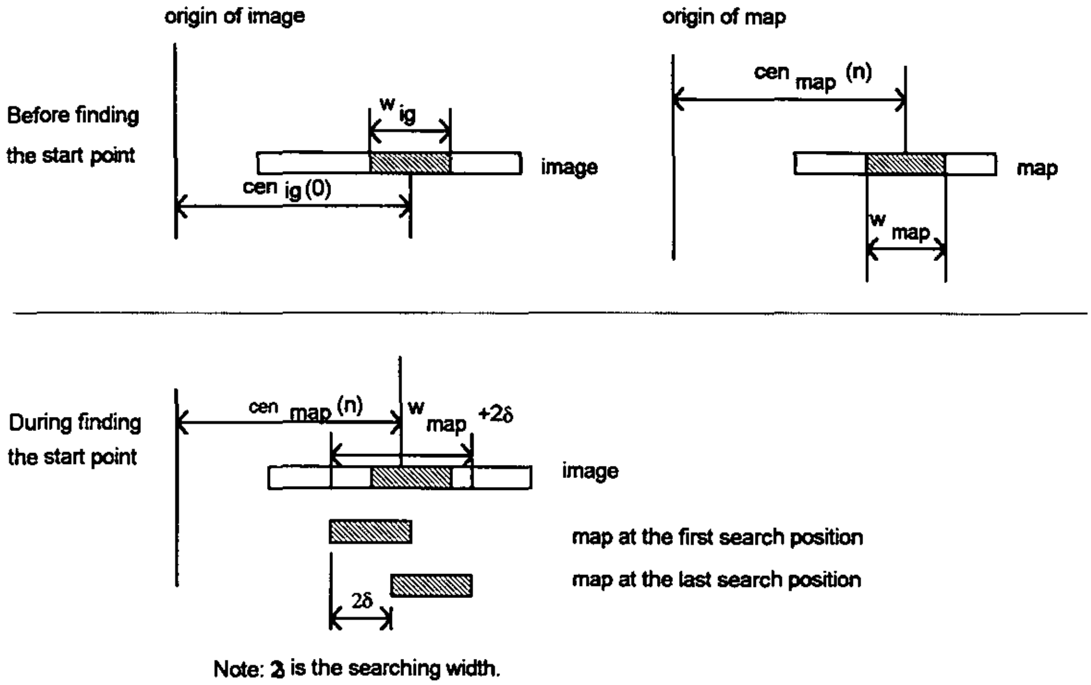


Figure 5.7 Search width and centre for finding the start point

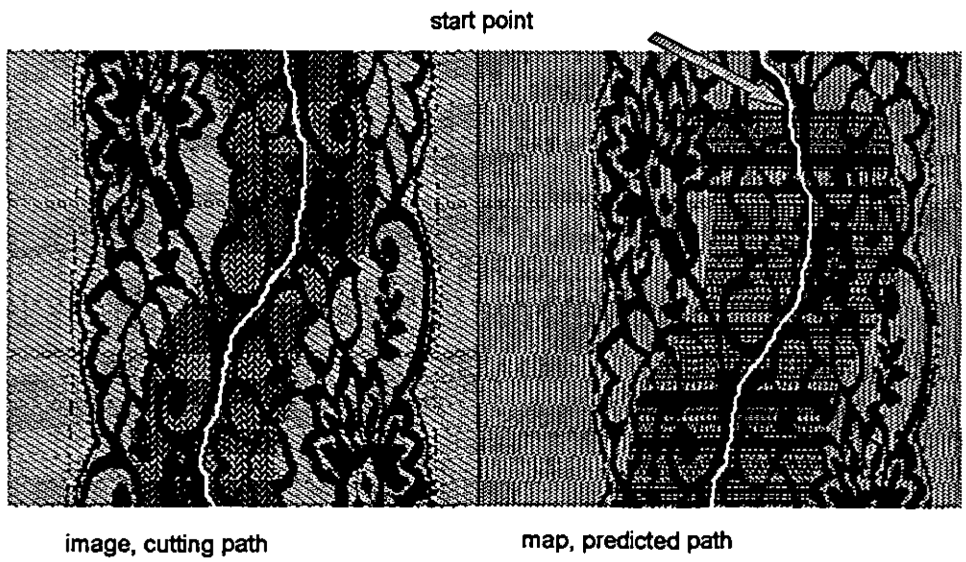
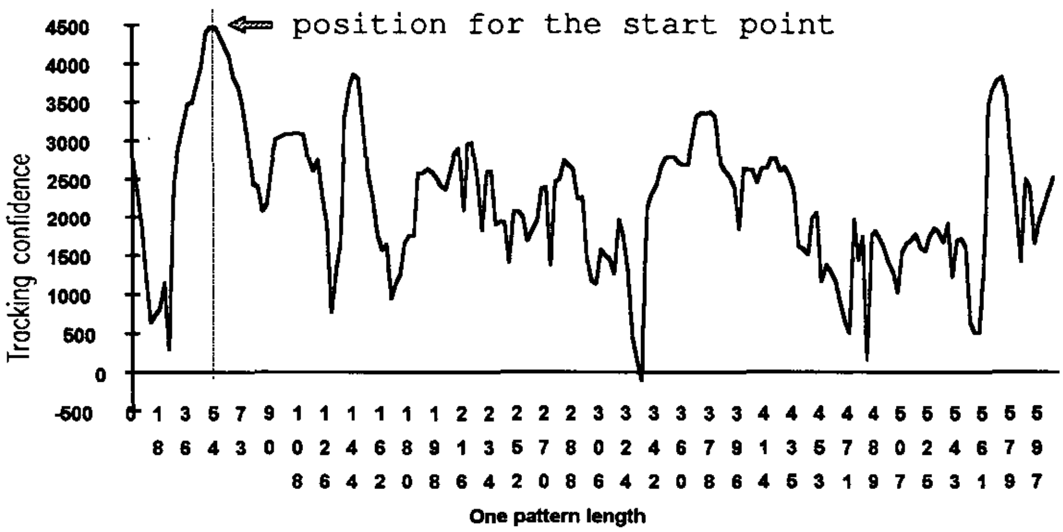
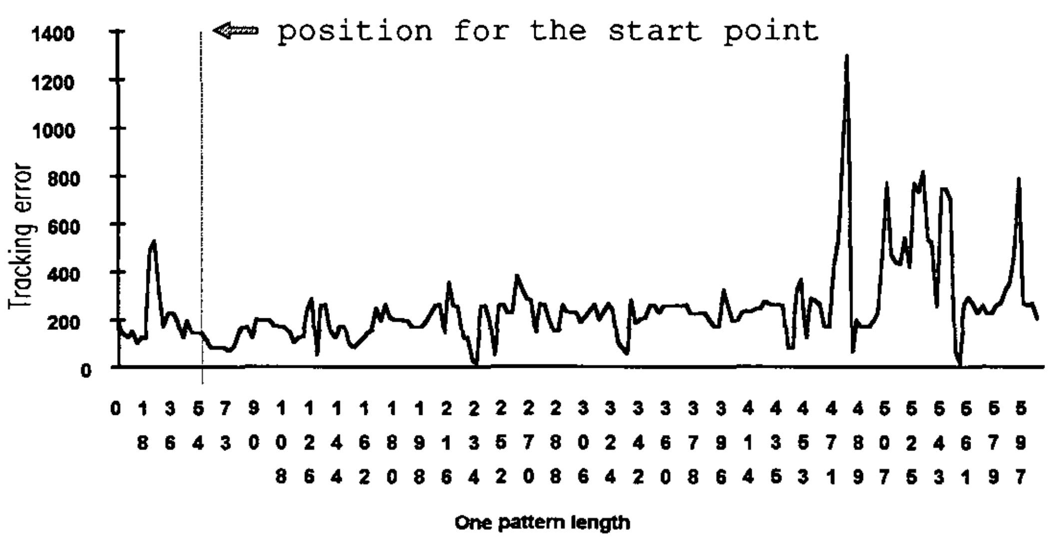


Figure 5.8 Finding the start point and tracking

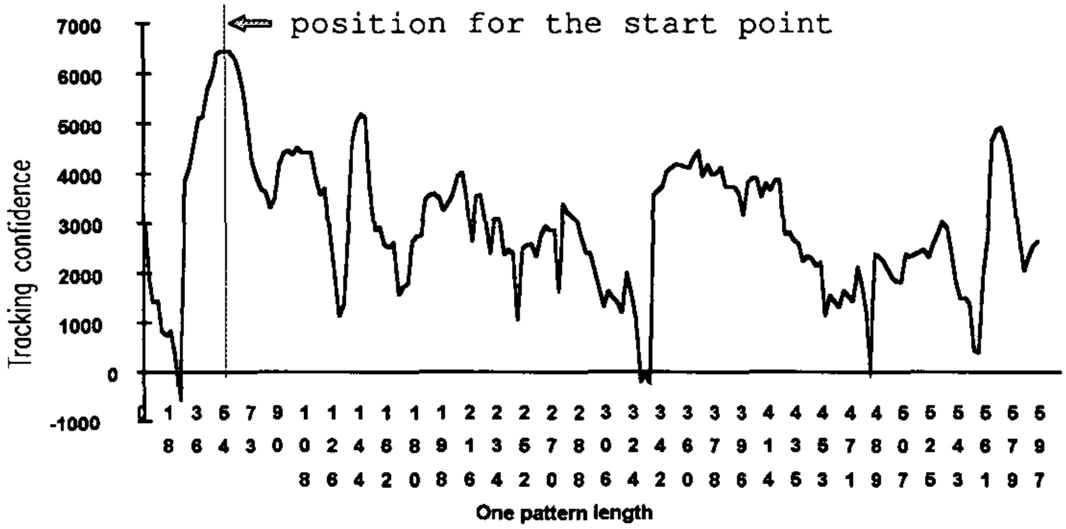


a: tracking confidence

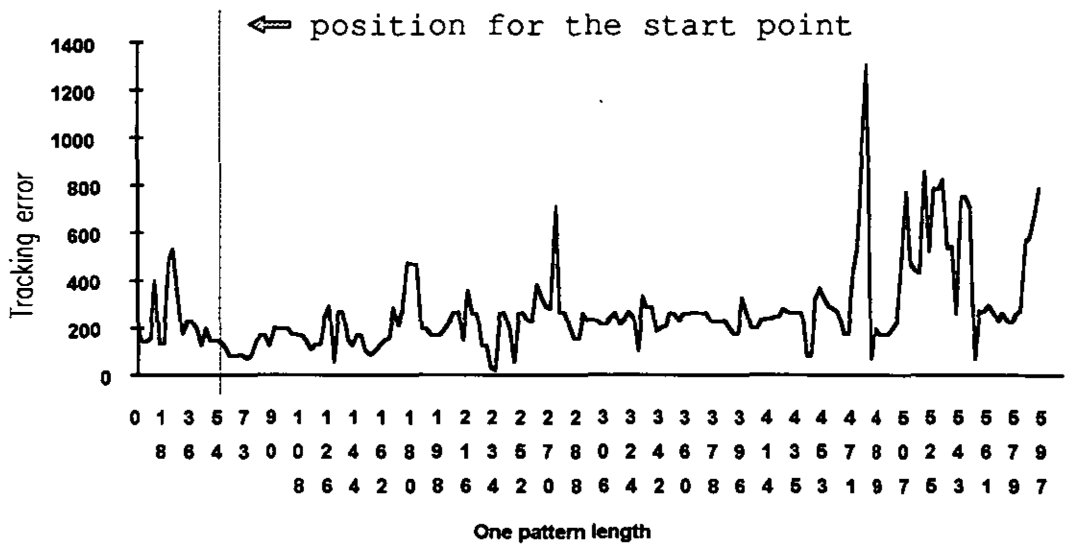


b: tracking error

Figure 5.9 Find the start point by 20 line tracking all over the map

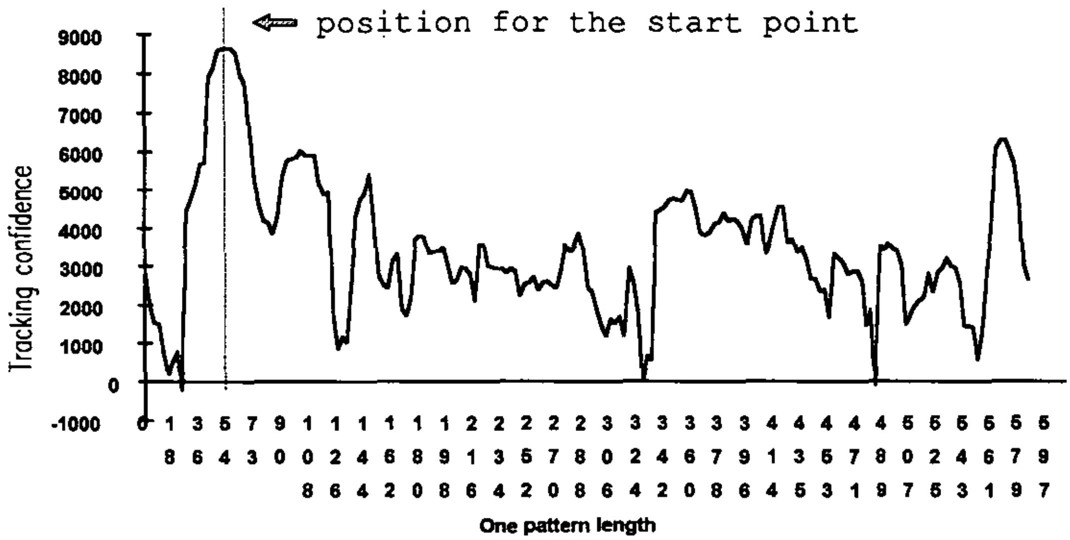


a: tracking confidence

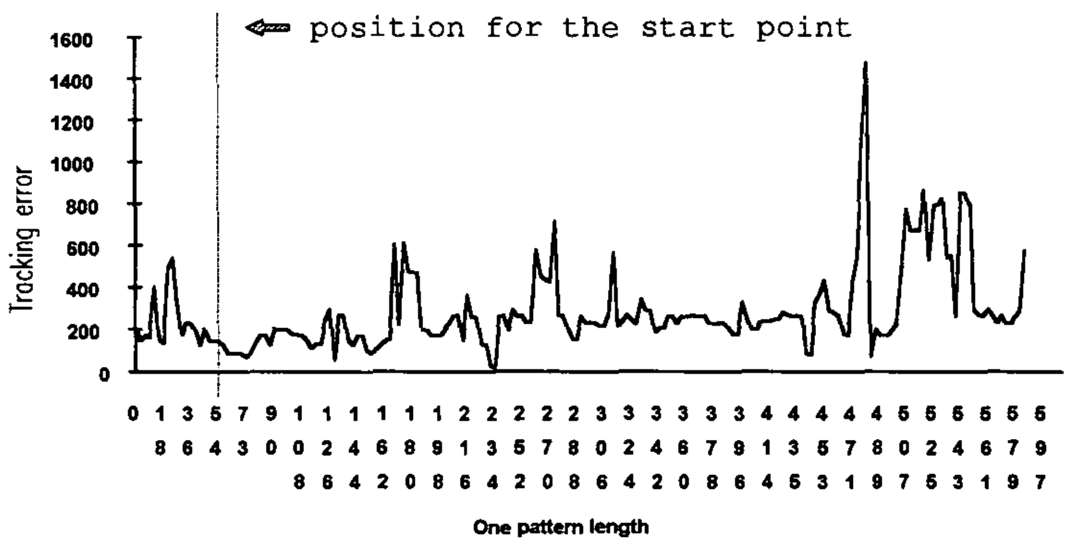


b: tracking error

Figure 5.10 Find the start point by 30 line tracking all over the map

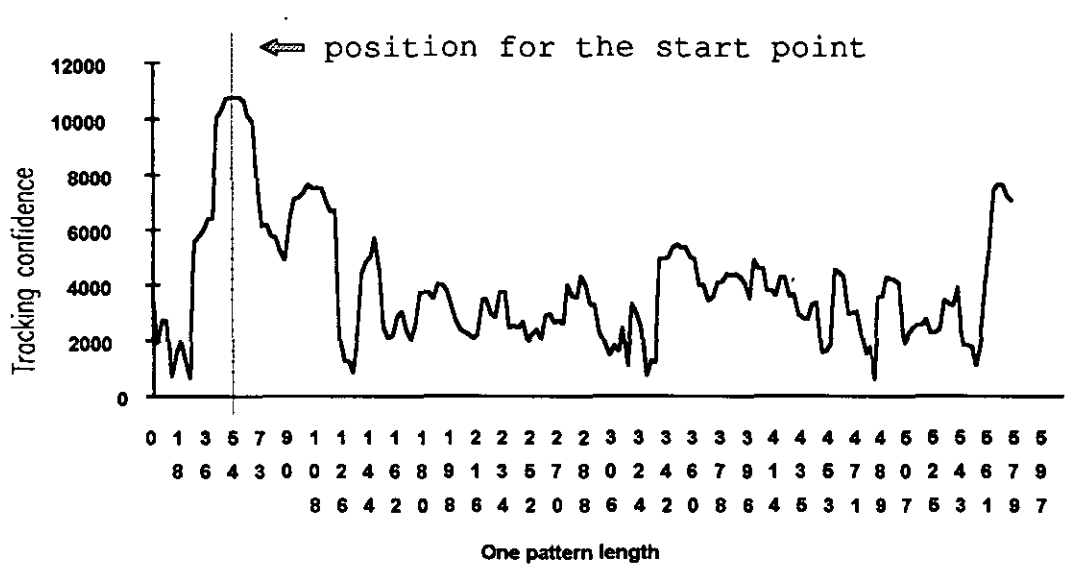


a: tracking confidence

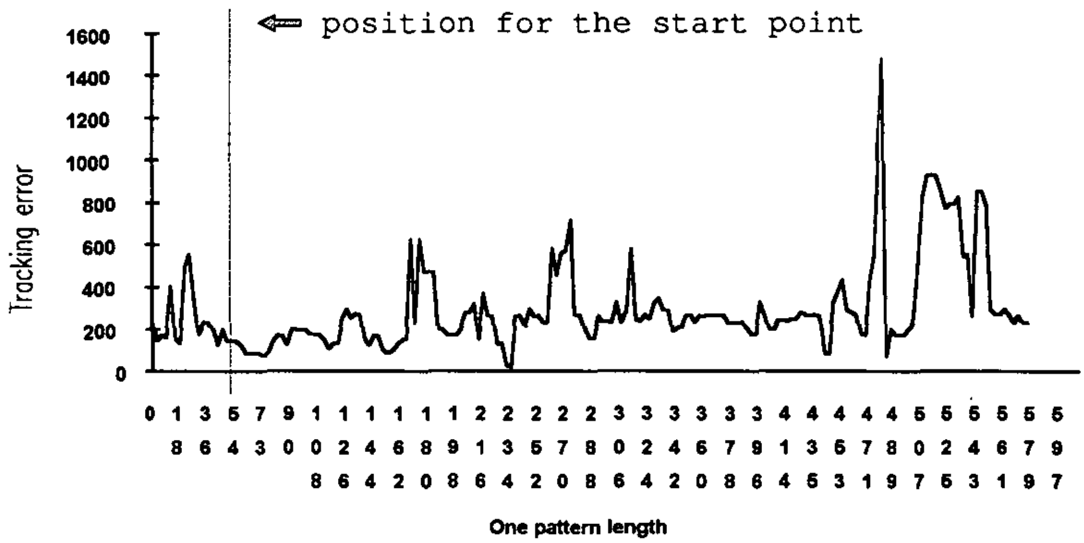


b: tracking error

Figure 5.11 Find the start point by 40 line tracking all over the map

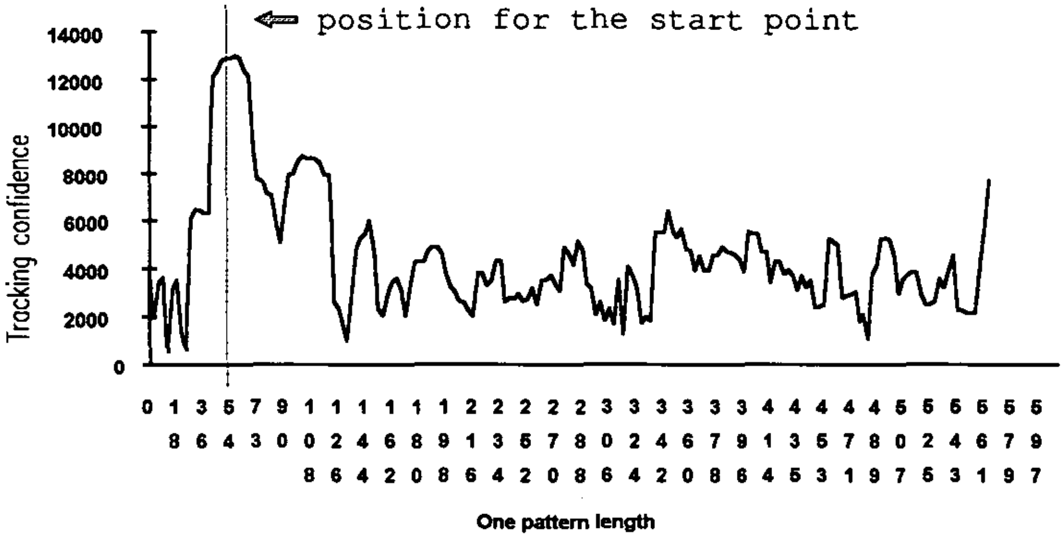


a: tracking confidence

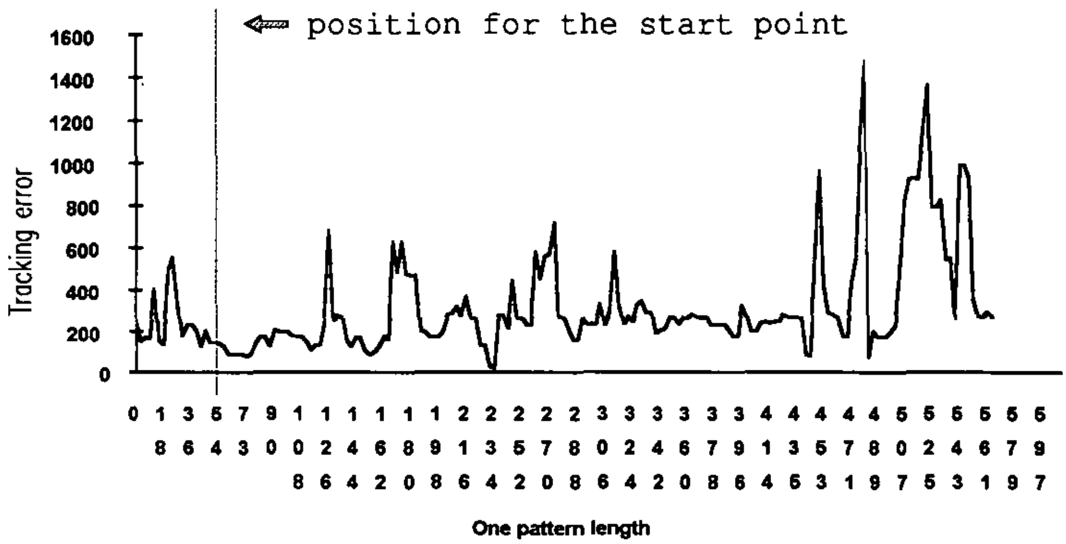


b: tracking error

Figure 5.12 Find the start point by 50 line tracking all over the map

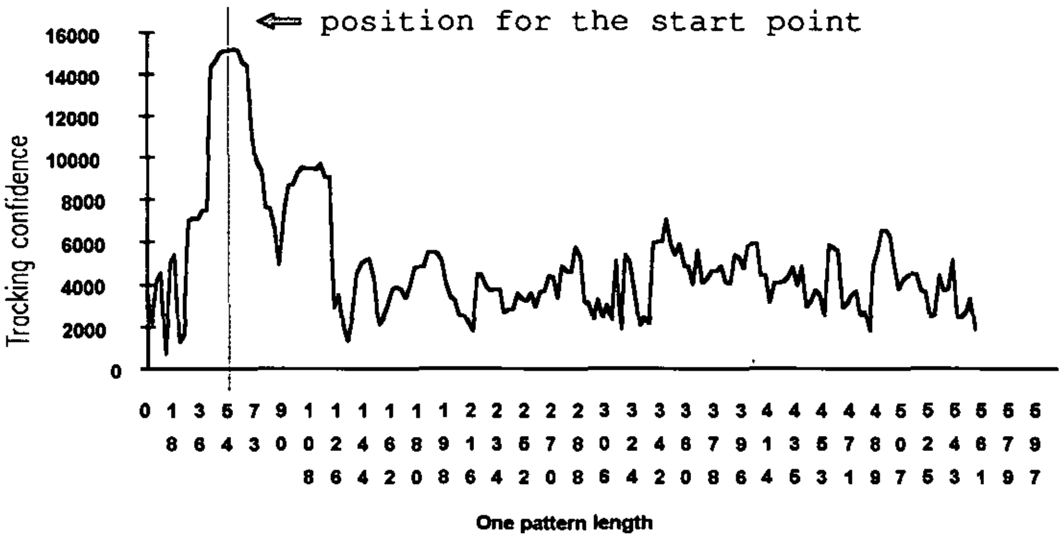


a: tracking confidence

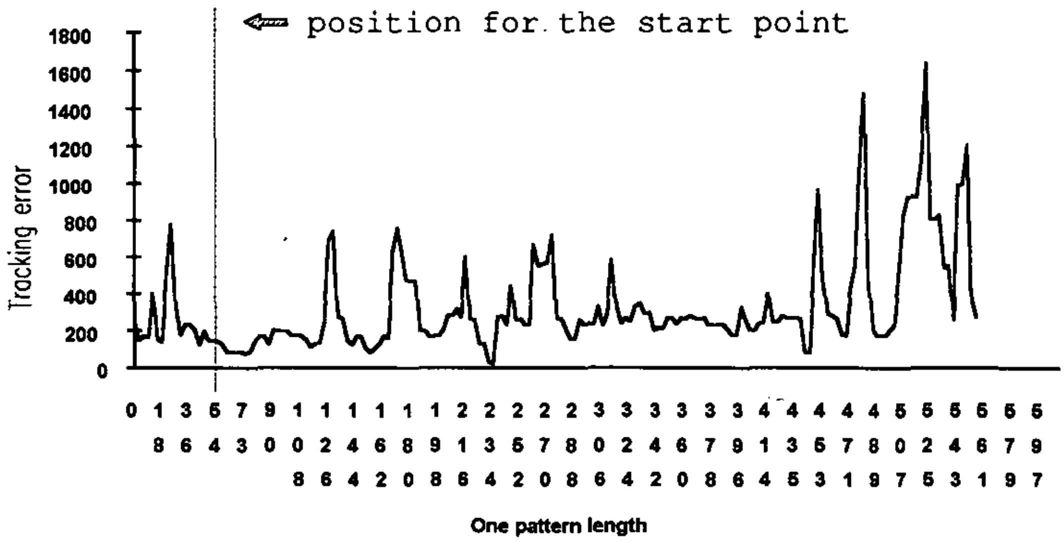


b: tracking error

Figure 5.13 Find the start point by 60 line tracking all over the map

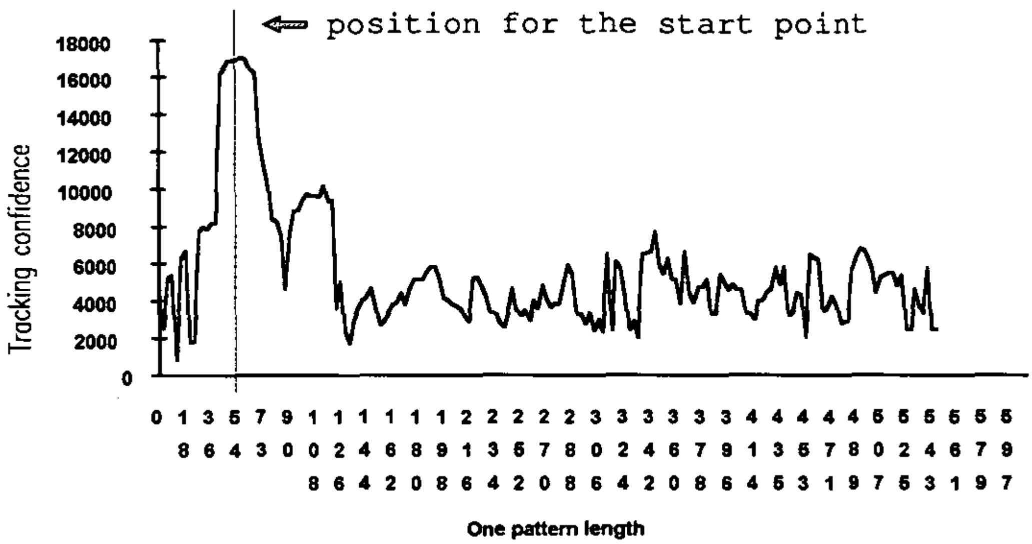


a: relative tracking confidence

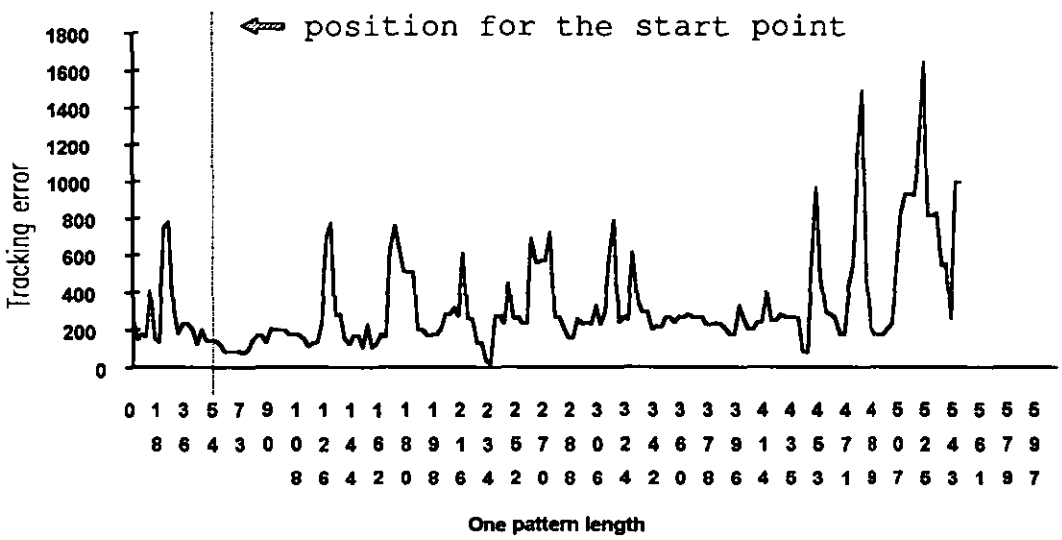


b: tracking error

Figure 5.14 Find the start point by 70 line tracking all over the map

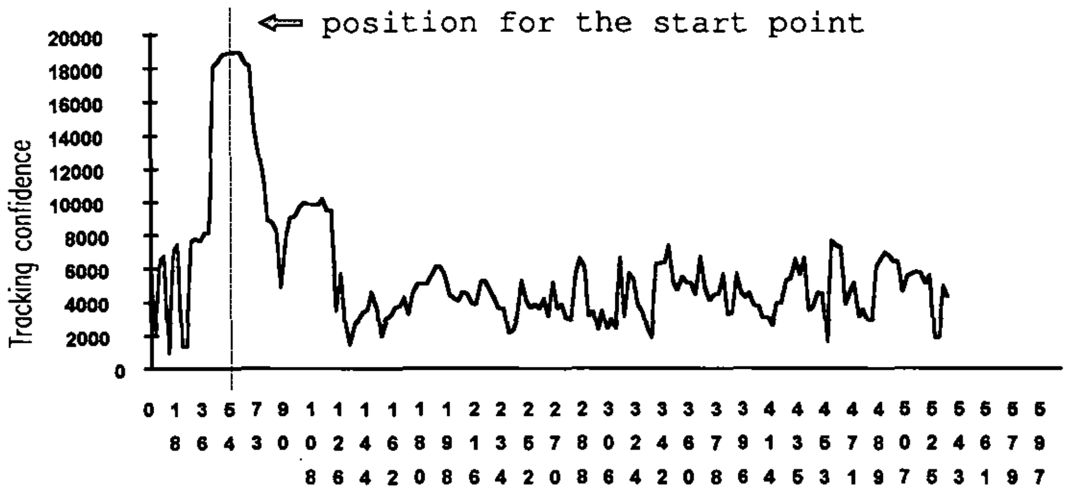


a: tracking confidence

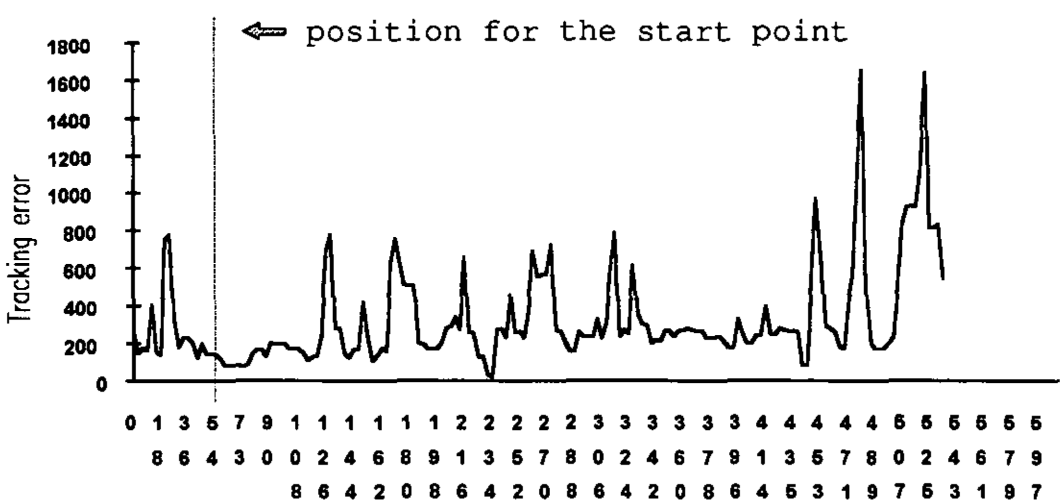


b: tracking error

Figure 5.15 Find the start point by 80 line tracking all over the map



a: tracking confidence



b: tracking error

Figure 5.16 Find the start point by 90 line tracking all over the map

Chapter 6

STABILISED WORKING CONDITIONS

6.1 INTRODUCTION

Maintaining stabilised working conditions for the vision system is the third research topic addressed in this thesis. Most vision systems have unique working parameters, such as lighting intensity, and an operational wavelenght range. When a working condition exceeds the working range of a vision system, its performance will deteriorate. For example, a land vehicle can be guided successfully by a vision system on the test road using a colour segmentation method when the sun is shining and there are no clouds in the sky, but when the road is shadowed by clouds the vision system cannot detect the road properly (Turk 1988) because of the change of object colour. When researchers face the fact that a vision system meet excessive working conditions, they should make a decision, whether to increase the working range of the vision system or to control the working condition or to do both. If a vision system is going to work outdoors and the working condition cannot be controlled fully, then increasing the working range is the preferred solution. In the land vehicle example, it is not possible to control the weather on the test road so not only a colour segment method but also other segment methods, such as an edge segment method, are needed to increase the working range of vision system. The working range of a vision system can broaden as technology is pushed forwards, but sooner or later the newly developed vision system will be faced with new limitations so working condition control for vision systems is an important area for research efforts, in order that vision systems can work successfully. In some indoor cases, a 'controlled' working condition will simplify a vision system. For example, a strip light (Batchelor 1985) can help a 2-D vision system to measure height and width of block in 3-D. But it is important to keep the working conditions stable. In the block example, the angle between the block and the strip light should be the same for each measurement. Otherwise different angles will produce different results for the same block. The vision system studied in this thesis is working in 'controlled' conditions so how to keep working conditions stable has to be studied.

The definition of 'stable working conditions' varies from vision system to vision system, and there is no universal way to keep stable working conditions for all vision systems. A systematic approach to maintain stable working conditions is to find which variables affect the performance of the vision system first, then to search for methods to stabilise them.

In this thesis there are four vision systems (see chapter 3). Vision system 1 is used to investigate various algorithms, such as tracking, Hough transform, and to evaluate the various parameters for vision system 2 and vision system 3, such as resolution, grey or binary image, at the beginning of the research. Since vision system 1 uses a stationary lace sample and longer conversion time (2 minutes) the requirement for stable working condition for this particular vision system was even light distribution. A light box for photograph provided a reasonable light distribution. Vision system 2 and vision system 3, which are virtually the same except a correlator board in vision system 3, are target systems for pattern cutting. Vision system 4 is used to evaluate both twin-beam and high-speed correlator tracking performances. The working conditions discussed in this chapter refer to vision system 2, 3 and 4.

The working conditions discussion will start form the working procedure of test rig 3 on which real lace is going to be cut. When a lace web passes under a camera, a vision system extracts information to determine the cutting path on the lace web. Then when the same part of lace web passes through the cutting zone, the laser beams cut lace at the required positions to scallop or centre cut. In chapter 4 and 5 the approaches to finding start point and tracking have been proposed and investigated. All discussions in chapter 4 and 5 imply that the incoming image for processing is similar to the map which is stored in DSP memory. The similarity of incoming image requires :

1. A lace sample positioned at any place in the light source working range will produce a similar image so that the vision system is not affected by variation in lighting intensity;
2. The digital image after the A/D conversion is similar to a digital image obtained previously if all other condition are the same;
3. The image is independent to the lace moving speed.

Additionally the sampling rate should be related to the lace movement, that is, the vision system should have feedback of the lace movement. All these parameters must be used to stabilise working conditions for the vision system.

The main issues for the stabilising the working conditions for the vision system are:

- an even and non-flickering lighting source;
- a threshold method to cope with lighting drift over a long time period;
- a synchronising method to sample lace in constant distance interval whilst maintaining a constant integration time for every sample.

In section 6.2 a study of the light source characteristics that produces suitable illumination for the vision system is presented. In section 6.3 research work on deciding the threshold to produce a consistent digital image is shown. The method for lace movement control of the vision system sample rate and maintain a consistent integration time for the camera are investigated in section 6.4.

6.2 LIGHT SOURCE

The function of illumination is to provide a stable, clear image to the camera so the importance of correct illumination to the success of a vision system cannot be overemphasised. The definition of 'correct illumination' varies from system to system. For example, a strobe light synchronised with the shutter of a high speed camera can produce a sharp image when the speed of the moving object is very high (see analysis in section 3.4.2). Another example is that a normal TV camera, which runs at 25-30 frames per second, works well without apparent lighting intensity change on the TV image with the lighting driven by AC, which produces 100 or 120 Hz ripples in illumination. The basic requirements of the illumination are that it must be stable in a period of time and uniform in the work region. How stable and how uniform will depend upon the vision system. In general, the requirement for a grey level vision system could be lower than for a binary vision system because the distribution of the brightness of the grey level image can be adjusted to some extent after acquisition (although this needs extra computation). The stability of illumination is more complicated than the uniformity problem. On the one hand, the variation of grey levels between frames or lines can be eliminated in a grey level vision system (in a manner similar to correcting for non-uniform illumination); on the other hand, the higher the speed of a vision system, the more difficulty is encountered in achieving stability of the illumination.

For lace fabric scanning, the vision system is working at a high line scan frequency, say 10 KHz, and with binary image processing. The analogue voltage level of a single pixel in a camera is the integrated result of the input light intensity in the area of this pixel during a certain period of time (Horn 1986), the "integration time". The integration time is the reciprocal of the line scan frequency in the camera. The camera will be sensitive to any light change with a frequency lower than the line scan frequency. In order to make the system's task possible, a good light source must produce a relatively uniform light output both across the line scan camera's working region and over a period of time, and preferably at a reasonable price.

6.2.1 Lamp Tests

Three types of lamps were investigated; normal fluorescent tubes, high frequency fluorescent tubes, and light emitting diodes (LED). Most tests were carried out directly via the CCD line scan camera which is used in the vision system in order to get a workable solution. The outputs of the line scan camera were observed and recorded by a Hameg 408 oscilloscope. The arrangement of camera and light source is shown in Figure 6.1. The camera is working at 20MHz pixel rate so the complete period to output 2048 pixels is 102.4μs. In order to give an approximate idea of the image length seen by the line scan camera corresponding to a time period on the oscilloscope, a filament lamp with 120mm length, 290mm from the mounting surface of the camera produces a 35μs waveform.

6.2.1.1 Fluorescent Tube with an Inductive Ballast

As expected, a normal fluorescent tube has reasonably even light distribution along the tube but has big 100Hz ripples, the ratio between maximum and minimum of the output of the camera is nearly 2:1, which could been seen by a line scan camera with scan rate higher than 100Hz. This makes the output of the comparator meaningless unless an extra control mechanism is introduced to compensate for the ripple effect. Figure 6.2 is the light distribution along the tube and the 100 Hz ripple. The 57μs represents 195mm length of light tube.

6.2.1.2 Fluorescent Tube with an Electronic Ballast

The fluorescent tube with an electronic ballast driven by 240v AC main power is more suitable for the line scan camera than the normal fluorescent tube with an inductive ballast. The electronic ballast drives a lamp at 20-40KHz (Osram 1991, Philips 1991) frequency which is 2 to 4 times higher than the line scan frequency and any light variation caused by that frequency will be hardly detected by the line scan camera. The light distribution along the tube and the low frequency ripple are shown in Figure 6.3. The 36μs represents 120mm length of light tube. There is 25mm length of tube with very low light intensity. The unevenness of the light intensity along the tube and the 100Hz frequency ripple is still there but is lower than that produced by a tube with a normal ballast. A DC driven electronic ballast was therefore seen as a possible way to get rid of the low frequency ripples.

A test on a 12v DC driven fluorescent tube with an electronic ballast proves that the low frequency ripple can be minimized but the light intensity of the output of the tube was not high enough for the high frequency line scan camera because integration period is reduced. In order to achieve the high light intensity output from the fluorescent tube and to avoid the low frequency ripple of the light intensity a 240v DC power supply was used to drive a fluorescent tube with an electronic ballast. Its performance is better than the previous results (In Figure 6.4 a photodiode was used to obtain these curves of the light intensity).

6.2.1.3 LED Array

The light emitting diode (LED) is one of the possible light sources which is driven by DC power and the light distribution can be adjusted by changing the value of the current resistors. Two different types of LED were investigated. Firstly an integrated LED array that is the light source of a GS-4500 document scanner shown in Figure 6.5a was tested. 48 LEDs are placed on the same plate which maintains their alignments, and a built-in half-cylindrical lens focuses all the LED's light in the same direction and the same distance. The light intensity distribution of the integrated LED array was recorded by shining it onto the camera (see Figure 6.6a). The light spots in Figure 6.6a are noticeable. But the LED array works well in the scanner and the bright spots are not seen in the scanned image because of the surface lit application. Secondly, a home-made ultra-bright LED array was tested as shown in Figure 6.5b. The intensity is sufficient for our 10KHz line scan requirement. The bright spots in Figure 6.6b are still there even if the light intensity of single LEDs is adjusted carefully. Figure 6.6c is the record of the ultra-bright LED array which does not show significant low frequency ripples and which is better than the electronic ballast fluorescent tube driven by AC power or even the DC one in this respect. The main problem with the LED light source is how to get rid of the bright spots in the view field.

The ultra-bright LED tested here has a hemispherical head in front and a reflectional bowl behind the light emitting body to make light in a small area. Figure 6.7 is an attempt to broaden the area of the bright spot of a single LED and to improve the light intensity distribution of the LEDs by removing the head of the LED. There is, however, no significant difference in oscilloscope records, comparing the record in Figure 6.7 with the record Figure 6.6b.

In order to know the light intensity characteristics of the LED the light intensities of individual LED on the naked CCD chip (the camera without a lens) are shown on Figure 6.8. There were four LEDs tested. 'A' is the bright area on CCD. There are quite large difference among the 'A'. The 'B' is the variation of brightness on the central area. Most LEDs give two peaks in their records of light intensities. The variation of the LEDs will cause significant problems if a uniform light source is to be made from LEDs.

6.2.2 Combined Light Intensity of LEDs

Non-uniformity along the line scan camera working region is the major problem for a LED array. In order to understand this problem a model for a LED array is presented here.

In this simple model, an area illuminated by a LED is a circle on a plane which is perpendicular to the LED axis and at a distance far from the LED. The light intensity in this area has a Gaussian distribution. If the light intensity levels are the same for every LED in the array, the vectors of beam centre from the LED for all LEDs are parallel, all LEDs lie on a line and the distances between every pair of adjacent LEDs are the same, there is a position along the vectors where a fairly uniform line can be found (Figure 6.9 is a two LEDs model, as the distance between two LEDs changes the combined light intensity is changed. The upper curve is the combined light intensity, the lower intersected curves are the individual light intensity output from individual LEDs. Position 3 shows the best uniformity of the combined light output judged visually).

If any of the factors, such as individual intensity, beam direction or intensity profile, change, the uniformity of light intensity will also change. Actually, small changes of any of these factors in a LED array will be likely when the array is made (Figure 6.10 is a six LEDs model but the distances between adjacent LEDs are slightly different from one another. The output of combined light intensity show the uniformity of light intensity is easily upset by a small distance change).

In practice, it is difficult to make every single LED perform in the same way. So bright spots will always be noticeable.

6.3 DECIDING THE THRESHOLD LEVEL FOR BINARY IMAGING

The video output from the camera varies with the light intensity of the lamp, the distance between the camera and lamp, the integration period of the camera and the pattern (or geometrical information) of the lace. From the pattern recognition or object location point of view, an ideal image for extracting the cutting path information on the lace only contains the geometrical information of the lace irrespective of variations of lamp intensity or voltage of power source within a certain range. But the light intensity of the lamp cannot be totally controlled because it is a function of the voltage of the power supply, the temperature of the room and hence the temperature of the lamp itself and the age of the lamp (Atkinson 1944). There are two kinds of variations of the lamp intensity, which can be seen by the line scan camera run up to 10 KHz line scan action, 1) 100Hz variation which is mainly caused by AC power; 2) the low frequency variation which is caused by the temperatures and age. The DC power supply described in section 6.2 has been used to eliminate the 100Hz ripple of the light intensity from the fluorescent lamp which is caused by AC power supply. But the low frequency drift of the voltage cannot be controlled by the high voltage DC power supply; the temperature of room and lamp cannot be controlled or can only be controlled at high cost and the temperature control will introduce complexity to the whole system. The variation effect due to the age of the lamp is difficult to quantify. Is it possible to produce a stable binary image by considering all of the unwanted factors? This is an important question for a real vision system guided laser cutting machine.

Here only the factors which change at very low frequency (i.e. do not change within several minutes) are considered (see Figure 6.11). In the test (Figure 6.11) a DC powered high frequency fluorescent tube was used as a light source, a photo diode, A/D converter and a computer were used to record the variation of the light source. During the test, DC voltage was purposely not adjusted. This therefore shows a typical variation of the fluorescent tube. In the first 10 minutes, the intensity reaches the maximum. Then the intensity reduces. After one hour, the average intensity achieves a stable value with very few change.

Clearly a consistent image cannot be produced by using a fixed threshold throughout a four hour period when the intensity of the lamp is changing.

There are several possible ways to produce a consistent image for tracking in this case. One is to use a hardware closed loop controller which can be implemented by a variable gain operational amplifier or a group of operational amplifiers and a photo detector. The photo detector can be used to transduce the light intensity to a electronic signal. Then the image signal from the camera is sent to an input of an operational amplifier. The gain of this amplifier is controlled by the signal from the photo detector to maintain the output of this amplifier consistent when the light intensity is changed. The system will produce a stable image if the whole system is carefully arranged and adjusted. But the positions of the camera, the lamp and the photo detector should be fixed or the whole system will need careful adjustment whenever the position of one of them is changed.

The second possible way is to utilise a photo detector, an analogue to digital converter (ADC) and software control. The function of the photo detector is the same as in the example above. An operational amplifier will be required if the electronic signal is too weak to be measured directly by the ADC. The ADC is used to convert the analogue signal from the photo detector to digital form to be read into a computer. Then the software decides a threshold level for the comparator according to the digital form of the light intensity. In a similar way to the hardware closed loop controller method, this approach will work if the positions of the lamp, the camera and the photo detector are fixed.

The first and second methods assume that there is a fixed relationship between the light level on the photo elements of the camera and the electronic signal level of the photo detector if the positions of the lamp, the camera and the photo detector are fixed. When even one of these positions is changed the relationship between the light level of the image and signal level of the photo detector will change. The relationship will be complicated if the light sensitivity of the camera and the photo detector are not the same.

A third method is to use the image signal from the camera to cope with the light intensity variations of the lamp directly and dynamically.

In this section, the principle of detecting intensity variations and the testing of the third method will be discussed and a study of deciding the threshold to produce a stable image will be presented.

6.3.1 Principle of Detecting Light Intensity Variation and Determining the Threshold Level

The output from the camera is a 'lace pattern modulated light intensity distribution' containing, 1) light intensity on the surface of the camera during the previous integrating period, which reflects the variation of the light intensity; 2) the pattern of a section of a lace when the lace is placed between the light source and a line scan camera. The image used for start point finding and tracking should contain pattern information only. Therefore, the pattern information must be separated (or demodulated) from the light intensity variation in the video output of the camera. Since the lace pattern is modulated with a variable light intensity of a light source (see Figure 6.11), the lace pattern demodulation should ideally be achieved by setting the threshold in accordance with the variation of the lighting. The lighting variation can be found in a grey level image, but not directly in the binary image used for start point search and pattern tracking. Therefore, in section 6.3.1.1 one method to form a grey image from the binary image by changing the threshold systematically is described. Then in section 6.3.1.2 two methods to decide the threshold are investigated.

6.3.1.1 Grey Level Image

The information about average light intensity cannot be found in a binary image in an obvious way, but a grey level image could tell us more about average light intensity. We can build a grey level image based on our binary image acquisition as follows. The analogue levels of the image from the camera are converted to binary image data by setting an analogue threshold level of an analogue comparator. A digital value for the threshold is sent to the comparator via a DSP and a D/A converter. A grey level image from the camera can be produced if the threshold level can be varied between zero to 255 which is the digital form of the comparator range (Figure 6.12) while the lace is not moved. The horizontal axis of Figure 6.12 is the lateral direction of the lace. The vertical axis is the grey level.

After that, 256 lines of image are in the DSP memory, and each line of the image corresponds to one threshold level of the comparator. All these lines can logically form a matrix (see table 6.1). The rows of the matrix represent the threshold level, the columns represent the pixels, and the transition row of each column represents the grey level of this pixel. For example, the grey level of pixel 14 is 3 (above this position all elements in this column are 0).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
line0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
line1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
line2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
line3	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
line4	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0
line5	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0
line6	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
line7	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0
line8	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0
line9	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
line10	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
line11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.1 Grey level image

6.3.1.2 Two Methods for Deciding the Threshold Level

The grey level image from the camera at one scan position contains the information of light intensity and the pattern of the lace acquired by the photo elements of the camera. There are at least two approaches to determine the threshold according to grey level image. One approach is to separate the light intensity information from the pattern information in a grey level image. Then a threshold level is determined according to the intensity-threshold relation obtained previously to produce a binary image. Another approach is to determine threshold directly from the grey level image. The latter approach is discussed here because of its simplicity.

In fact, for a given lace, there are many possible scanning positions when the grey level image is produced. For example, the repeat length of the lace is 100mm, and the resolution in longitudinal direction is 3 lines/mm. There are 300 different possible positions when a grey level image of this lace is produced. It is important to make the threshold level decision in one stationary position from the production point of view. So a successful method applied to threshold decision should be independent of the scanning position on the lace.

There are several ways to decide the working threshold level by using the grey level image. One method is to find the ratio of black pixels and white pixels for each threshold level. This is a tangent law typed curve with a slow ratio change at one end of the curve and a rapid ratio change at another end. For example, in table 6.1 '1' represents the white pixel and '0' represents the black pixel. When threshold level is set on 0 (line0), all pixels are white and black/white ratio is 0 for this threshold level. The black/white ratio increases as the threshold level increases. It is difficult to find one characteristic of the curve to set the threshold level which is independent of the lace position and light intensity. Two practical methods are discussed in this section.

The first way to decide the threshold level is to find the 'edges' along one scanning image and make a curve of the number of edges. For each grey level, the edge points at which the image is changed from black to white or from white to black are counted. All the counts are put together to form an edge curve. Then the region of the edge curve where its variation is less than a predetermined value, *d*, can be used to decide the working threshold level. Figure 6.13 is a typical edge curve. There are three different parts in this curve, flat part with zero edge level, part with little edge number change and part with rapid change. The flat parts with zero edge level indicates that there is no edge in the image (all white or all black). The part with rapid change of edge number means that the image is on a transition band between the stable image and the blank image (white or black). The part with little change of the edge number (not zero edge) implies that the image is stable if the threshold level change in this range. The edge number in this part also is, or is likely to approximate to, the number of edges of the pattern in the scanning area. This is a part of geometric information of the lace pattern. In order to suit the light intensity variation afterwards (see Figure 6.11) the threshold level should set in the middle of this part.

As the light intensity varies the edge curve for a particular part of the lace pattern will move along the threshold axis. There is little change in the profile in the desired part of the curve (see Appendix 3). In fact this profile reflects the lace optical characteristics, the lace geometrical characteristics, and the optical characteristics of the whole vision system. All these characteristics are independent of the light intensity. But the position of the desired part of the curve is related to the light intensity. One characteristic of this profile is the variation of this part of the curve, or 'roughness'. We can use the 'roughness' to search the desired profile and to decide the threshold level that is related to the light intensity.

The 'd' is the predefined 'roughness' of desired part of curve. If the flat part with zero edge number is ignored, the 'd' can be used to search the flat part with a little edge number change in the curve. In Figure 6.13 the part of curve from 'begin' to 'end' is matching the 'd' requirement. For a certain edge curve different 'd' can produce different 'begin' and 'end'.

If the same 'd' is applied to different parts of the lace pattern (assuming that the thick and thin parts of threads are evenly distributed) under the same illuminating condition, the 'begin' and 'end' should have no difference from one part to another. But the illuminating conditions will not be the same, the sensitivities of CCD elements are the variables of the temperature, and the thick and thin threads are distributed with pattern. All these will cause the change of the edge number. So the 'begin' and 'end' will vary from one place to another and from time to time. The threshold level for a stable binary image should be decided by using equation 6.1 and 6.2. Equation 6.1 is used to get a common grey level range for whole pattern of lace.

$$\begin{aligned} begin_{com} &= Max(begin_i) \\ end_{com} &= Min(end_i) \\ 1 &\le i \le n \end{aligned} \qquad \text{eq.6.1}$$

The n is the number of scanning positions within a non-repeated pattern. If a threshold level is decided by equation 6.2, there is the biggest chance to keep the binary image stable when the light intensity is slightly changed.

$$threshold = (begin_{com} + end_{com}) / 2 \qquad \text{eq.6.2}$$

If the differences between 'begin's are small and the difference between 'end's are small from one position to another, the '$begin_{com}$' and 'end_{com}' can be replaced by the '$begin_i$' and 'end_i' in equation 6.2 to decide the threshold level.

The second way to decide the threshold level is to find a range in which the whole line of the image is neither all zeros nor all ones. Then the working threshold level is decided at some position within this range. The position of transition from all ones to not-all-ones (which is the bottom line on Figure 6.12) of the image is decided by the optical characteristic of the thickest part of the lace. The position of transition from all zeros to not-all-zeros (which is the top line on Figure 6.12) is decided by the optical characteristic of the thinnest part of the lace. If the distribution of the thickness of lace is arranged in such a way that there is at least one spot of the thinnest part of the lace and one spot of the thickest part of the lace in one line scan range, the two transition positions should be independent of the geometrical information of the lace pattern and decided by the light intensity level and the thickness of the lace. Therefore, the geometrical information can be separated. Then the threshold level will be decided by equation 6.3.

$$threshold = top \times \alpha + bottom \times \beta \qquad \text{eq.6.3}$$

The α and β are determined by tests.

6.3.2 Threshold Test

Tests on three different scanning positions with different light intensity levels were carried out (Figure 6.14). Method 1 and method 2 for determining the threshold used. The data and curves for method 1 are listed in Appendix 3.

Table 6.2 lists the beginning and end grey levels for method 1.

The 'd' is a arbitrary margin to decide the beginning and end of the flat part of the curve (see Figure 6.13).

In table 6.2 the middle of the flat part of the curve is stable for different scanning positions under the same illumination condition. This suggests that the middle point of a scanning position may be used as a threshold level instead of the middle point of the common range without major problems. This improvement can save a lot of time to avoid scanning the whole non-repeated pattern.

	170v			185v			200v		
	b	m	e	b	m	e	b	m	e
position1	194	211	227	184	209	229	174	200	227
position2	194	214	234	184	207	229	174	200	226
position3	199	214	229	191	208	225	179	200	220

Note: b: begin m: middle e:end d=10

Table 6.2 Test results of method one for threshold determination

Table 6.3 is the top and bottom grey levels for the method 2. *P* is determined by equation 6.4. The α and β for the lace are 5/8 and 3/8 decided by the test.

$$p = top * 5/8 + bottom * 3/8 \qquad \text{eq.6.4}$$

The values of α and β are taken to make the 'p' as near as the 'm' by using the method 1.

	170v			185v			200v		
	b	p	t	b	p	t	b	p	t
position1	161	216	250	148	210	249	135	204	247
position2	162	216	251	149	211	250	135	204	247
position3	163	214	251	149	206	243	137	201	240

Note: b: bottom p: position t:top the p is obtained by equation 6.4.

Table 6.3 Test results of method two for threshold determination

Comparing method 2 and method 1 the maximum difference of the threshold level is 5 and the 'p' positions chosen by method 2 within the desired part of the curve (see section 6.3.1.2) near the middle of it. Method 2 has been chosen because of its simplicity. The images produced by employing method 2 are successfully used for start point and track the cutting path when the voltage of the DC power supply varies from 170v to 200v which is bigger than any voltage shift caused by the variation of the AC voltage (comparing Figure 6.11 with Figure 6.15). The working procedure is that the lace stays fixed in one position, the camera is driven by the DSP and the grey image data are processed by DSP. After that a threshold level is set by the DSP. Then the lace is moved, the camera is driven by an encoder and the binary images are sent to a DSP to find the start point and then track the cutting path.

6.3.3 Discussion

The implementation of method 2 in DSP code is simpler than that of method 1, so method 2 has been used.

Deciding on the threshold level must be carried out at beginning of every run before start point finding and cutting path tracking. After the threshold level has been determined, the image processed by the vision system switches back to binary. One question is whether it is necessary to make threshold decisions during tracking, if the light intensity level could change by an intolerable amount during this working period. The interface board can only produce either grey scale image or binary image and the DSP can only deal with image data for either tracking or threshold decision at one time. It is impossible to use DSP for both tracking and to monitor the light intensity change at the same time.

An alternative way to monitor the light intensity change is to employ a photo detector and an ADC. When the DSP decides the threshold for the comparator, the photo detector and ADC also communicate the light intensity level to the host computer (the Syntel LC850 based on the Motorola 68020). When the DSP is doing the tracking task the computer looks at the output of the ADC every minute or after a certain period of time. If the absolute difference of the current light intensity level and the stored light intensity level is bigger than a certain amount, the tracking will stop and threshold decision algorithm will execute to determine a new threshold.

6.4 SYNCHRONISING THE CAMERA WITH THE MOVEMENT OF THE LACE

The synchronisation between the scanning action of the line scan camera and the feeding of the lace is one of the important parts of the vision system and the whole lace cutting system not just to enable the start point search and tracking programs to work and but also to enable a CO_2 laser beam to cut lace in the determined position.

6.4.1 Synchronising Problems

Due to the real-time application of the vision system, the synchronisation between the vision system and the feeding of the lace cannot be overemphasised. The image processing in the vision system will be meaningless if there is no synchronisation. If all other parts of the vision system are triggered by the output of the interface between the camera and the DSP, the synchronisation of the vision system with the movement of the lace can be considered as the synchronising of the camera action with the movement of the lace, which is discussed in this chapter.

Maintaining a constant physical distance between any two consecutive lines of the image is a basic requirement of the synchronisation. One approach used in the tracking program tests to achieve a constant distance between any two consecutive lines of the image was to feed the lace at a constant speed while the scanning action of the camera was kept at a fixed frequency. To achieve this a line scanning signal with fixed frequency was generated by the timer on one of the DSP chips. This arrangement cannot be used in the practical case because of the lace velocity changes during the operation. First the lace should move a short distance to feed enough image to DSP and let the DSP find the start point for tracking. After that the lace speed is increased from zero to maximum to track a predetermined path on the lace. Since the speed of lace varies, the travelling distance on the lace correspondent with two consecutive lines of the image will vary and the whole system will fail.

Another method to maintain the constant inter-line distance is to vary the scan rate of the camera as the speed of lace varies. In section 6.4.2 a method for maintaining a reasonably constant distance is discussed.

Whilst this latter approach offers a potential solution there is a further problem. As the scan rate of the camera varies with the variation of the lace movement, the exposure period, which is the same as the integration period, of the photo elements in the camera will be varied. The variation of the integration period will cause an inconsistency of the image being sent to the DSP. So in section 6.4.3 a method is presented to enable a constant integration period to be maintained when the lace speed varies.

Section 6.4.4 discusses a method of DSP control of the scanning action of the camera to produce a grey level image for determining the threshold level (see section 6.3).

6.4.2 Monitoring Lace Movement

There are two kinds of movement sensors, contacting and non-contacting. Sensors based on magnetic or electric field changes cannot be used to sense lace movement because of the physical characteristics of the lace. A line scanning camera is one possibility for a non-contacting sensor (Bretschi 1979). The principle of the line scan camera as a sensor would be that the movement of the lace is monitored by a camera and measured by matching several predetermined patterns of the lace. In this measurement procedure, the distance to be measured by a line scan camera should be bigger than the resolution of the measuring system itself, which needs to measure several positions on the lace across a predefined distance interval to produce a signal to trigger the tracking. The problem of using a camera as a movement sensor is that a vary large amount of image data would need to be processed, even more than the image data for tracking, and this makes such a vision system approach too complicated. Even if the computing speed were not a problem, there are still some problems that have not been solved yet by using line scan camera only, such as that whether the lace is actually stretched or whether it is moving at very slow speed when an apparently stretched image is acquired (the images are similar in these two cases). Maybe an area scan camera could be used to estimate the speed of the lace, but there is no such high speed camera, say at least 5000 frames per second (at 256x256 pixels, the pixel rate for this kind of camera would be at least 330MHz), on the market, and the computing power to process this kind of information will be higher than that required to track the cut path on the lace.

Optical measurement for the displacement or velocity based on Laser technology is another non-contacting method. Laser interferometers can measure the displacement with wavelength accuracy by comparing a laser beam reflected from the object to be measured with a reference beam. The direction of the reflection should be the same direction of the motion of the object (Dakin 1988). This method cannot, however, be used to measure movement in the direction perpendicular to the reflecting surface (furthermore the lace surface is not an idea reflecting surface) as with our lace movement (Figure 3.13). The Doppler Laser sensor has been used to measure the velocity of liquid (Dakin 1988), in which a laser beam can be directed into a liquid flow at an angle. But the liquid to be measured should have an isotropic character which is not the case with lace.

An optical encoder (Ruocco 1987) is one type of contacting sensor which is widely used in position and speed control applications because it is easy to interface to a computer. Optical encoders can be classified into two groups, linear encoders and angular encoders. The linear encoder is not suitable for measuring the displacement of an endless lace loop (or longer lace) due to its limited movement. An angular encoder with a very low movement moment of inertia can measure lace movement by using a rubber typed wheel mounted on the encoder shaft and running on the surface of the moving lace to drive the encoder directly. This is a straightforward way to detect the movement of the lace.

In theory there may be some loss of synchronisation between the lace surface and the rubber wheel during some period of the movement of the lace due to friction (Hannah 1984). In order to investigate this kind of behaviour, the author performed tests on test rig 2 in which the linear table was driven by a servo motor with a closed loop position control. The encoder output gives pulses at much finer distance increments than required for triggering the line-scan camera. These are then divided down by a "distance counter". The line scan action of the camera is triggered by the output of the distance counter rather than the direct output of the encoder. The test result (see Figure 6.16) shows that there are initially no pulses output from the distance counter. After 3mm of lace movement the ratio of pulses to lace movement is nearly constant. The reason for no outputs being produced at the beginning could be that the driving force on the wheel produced by the deformation of the lace is increased as the lace holder moves further; after 2.75mm the driving force remains nearly constant as the lace holder moves. The first line scan action will be triggered at the 3mm position then repeated at a quite constant increments of lace movement so that the image will be comprised of equi-spaced lines. It is no problem for an unidirectional transportation system. For bidirectional transportation system further study is needed.

In test rig 3, a pair of plastic strips are employed to increase the angle of wrap or contact angle, α, in order to increase the friction force on the wheel. The diameter of the wheel should be as small as possible. The smaller the wheel, the lower moment of inertia and the more rapid the response of the encoder. The lower moment of inertia of the wheel will also reduce the distortion of the lace caused by the encoder wheel because of low driven force from the lace. The more rapid response will provide a more accurate movement signal.

6.4.3 Maintaining a Constant Integration Period

A major problem associated with the synchronisation between camera scanning action and lace movement is how to keep a constant integration period for the line scan camera. This is one of the factors which determine the analogue level of the video signal from the camera. In general, the longer the integration period, the higher the analogue video output from the camera if other factors remain the same. The method of section 6.4.2 can keep the physical length of lace between any two consecutive scans constant while the speed of the lace movement varies but raises a problem about the variation of the integration period due to the variation of the speed of the lace. The worst case is that when the lace stops for any reason, the integration on the camera will cause saturation of its photo-sites (or photo elements). The method for converting an analogue video signal to a digital signal used in this thesis is to employ an analogue comparator to produce a binary image using a preset threshold for the comparator (see Appendix 4). There are two methods to keep the binary image consistent: constant integration period or varying the threshold level. To implement the second method is quite difficult especially in real-time. If a time counter for the integration period is set independently of the distance counter but is triggered by the distance counter, the first method can be implemented.

In principle, a line scan CCD camera works in two different phases to produce a line of image. In the first phase, the CCD's photo elements collect photons which are converted into electronic charge. The level of electrons in one element is proportional to the number of the photons falling onto the element in this period of time, providing the electrons do not exceed the saturation level. This period is called the integrating period. Then, in the next phase, the CCD transfers the electrons, namely image data, from the photo elements to one or two shift registers in parallel and clocks the image data out from the shift registers in series. One line of image is produced during two phases that can be indicated by two line scanning signals. The first scanning signal starts the integration, the second signal ends the integration and starts the second phase. Because the integrating and clocking out are working on different areas of the CCD, one line of image data is clocked out while the next line of image data is being integrated. The output of the image seems synchronised with the line scanning signal, one line scanning signal for one line of image data. The longer the period of one line scan the higher the image level that is clocked out in the next line scanning period. If a signal from a distance counter, which maintains the constant intervals of the physical distance on the lace, is used as a line scan signal for the camera, the physical distance between the consecutive image lines will be constant. But this method will vary the integration period of the CCD and introduce further distortion to the image which will make the image processing task complicated. Figure 6.17 is an example. The n, $n+1$, $n+2$ are the scan positions that are of constant physical distance intervals. The pattern is a straight line. The time intervals of the scan positions are varied because of the speed so the video output is varied. A single threshold is used so two lines of the binary image are different but the pattern being scanned is the same.

One approach to solving this problem is to send two line scan signals (double pauses) to the camera for every signal from the distance counter. The image correspondent with the first line scan signal is thrown away. Only the image correspondent with the second line scan signal is sent to the DSP for further processing (Figure 6.18, Figure 6.19, Figure 6.20). The reason to throw the first part of the image is that the integration time to produce this part of image is not constant. It is important to keep the time interval between the two line scan signals constant by counting a system clock in spite of the variation of the time intervals between any two consecutive signals from the distance counter. Actually, the first line scan signal is synchronised with the movement of lace. The video signal from the camera after the first line scan signal is meaningless because it is integrated in the camera before the first line scan signal. The video signal from the camera after the second line scan signal is correct because it is integrated between the first and second line scan signals that are at a fixed time interval and the beginning of the integration is synchronised with the movement of the lace.

Another requirement is to avoid the saturation of photo elements in the camera due to a long time exposure, even if the light intensity does not change. Observation of the camera output after saturation showed that there are several saturated lines at the beginning of the image if line scanning signal is not active at normal operation frequency. The method of anti-saturation is that a line scanning signal with fixed frequency is used to drive the camera to clock out the image signal continuously but the image signal is not sent to the DSP if no synchronising signal is generated by the distance counter. When the synchronising signal arrives, double pulses are triggered. The second pulse will send the image integrated between the first and second pulses to the DSP for processing.

The images being taken at various lace speed from nearly zero to 1 m/sec have been shown experimentally to have no significant differences using the methods that are discussed here.

6.4.4 Triggering the Camera Directly from the DSP

One case in which the camera needs to be triggered by the DSP is to produce a grey level image for threshold determination. In order to produce a grey image on a single position of the lace by using the method described in section 6.3, the scanning action of the camera will repeat while the lace does not move. The encoder signal cannot be used in this case. The best way to control the scanning action is to use the DSP so that in producing the grey level image the DSP can control both threshold level and scanning signal and the threshold can be set before the next scanning signal is issued (the schematic diagram of the synchronising board is shown in the Appendix 5).

6.4.5 Valid Scanning Rate

The camera scanning rate, *f*, controlled by the DSP must satisfy inequality 6.5.

$$f < \frac{1}{2p} \qquad \text{eq. 6.5}$$

where p is the integration period of camera (see Figure 6.18). Since the double pauses are used to produce one line of image, the minimum time interval between two contiguous line scanning signals is $2p$. The p is determined by equation 6.6 in this case.

$$p = \frac{n}{f_{pixel}} \qquad \text{eq. 6.6}$$

where f_{pixel} is the pixel rate of the camera and n is the number of the pixels including delay pulses before valid video appears on the output pin of the camera. If n=2100, f_{pixel} =20MHz, f must be lower than 4.76KHz. The inequality 6.5 must also apply to the scanning rate issued by the distance counter.

6.5 CONCLUDING REMARKS

Several aspects of stabilising working conditions of the vision system, light source, threshold and synchronisation, have been investigated in this chapter.

The high frequency fluorescent tube driven by DC power will be the best choice from the even light distribution point of view. The line scan camera is not affected by high frequency ripple caused by an electronic ballast because the frequency of the tube is at lest 4 times higher than the frequency of the camera line scan action.

The LED array is bright enough for the 10KHz line scan frequency of the line scan camera and the light intensity is quite stable, but bright spots in the field of view are a major problem.

Threshold determined by upper and lower bounds works when the DC voltage level is changed from 170v to 200v. If the working temperature is not changed quickly, the threshold decision algorithm can be used at the beginning of every use of the start point finding and cutting path tracking algorithms to cope adequately with the changes in illumination caused by tube ageing and the working temperature drift.

The encoder method can produce a scan signal for every constant incremental distance travelled by the lace, the double pulse method can keep the integration time constant and by switching between the distance counter and DSP the vision system can work in different modes for threshold decision, start point finding and tracking. These methods enable the image to be kept consistent regardless of variation of the lace speed. In order to avoid saturation of the photo elements of the camera due to longer exposure period caused by a lower lace feeding speed, an exposure control mechanism based on a time counter has been used; the camera works in a continuous scan mode with a fixed frequency and the image produced by the camera is not sent to the DSP to be processed until a synchronising signal from the distance counter arrives. The drawback of the double pulse method is that the maximum rate of system line scan action reduces from 10KHz to 4.76KHz. But this is not a problem for vision system 2 or vision system 3 because 4.76KHz line scan rate can produce the resolution as high as 4.76 pixel/mm when the lace moves at 1m/s. It is certain that the vision system has sufficient resolution from stationary conditions up to 1m/s.

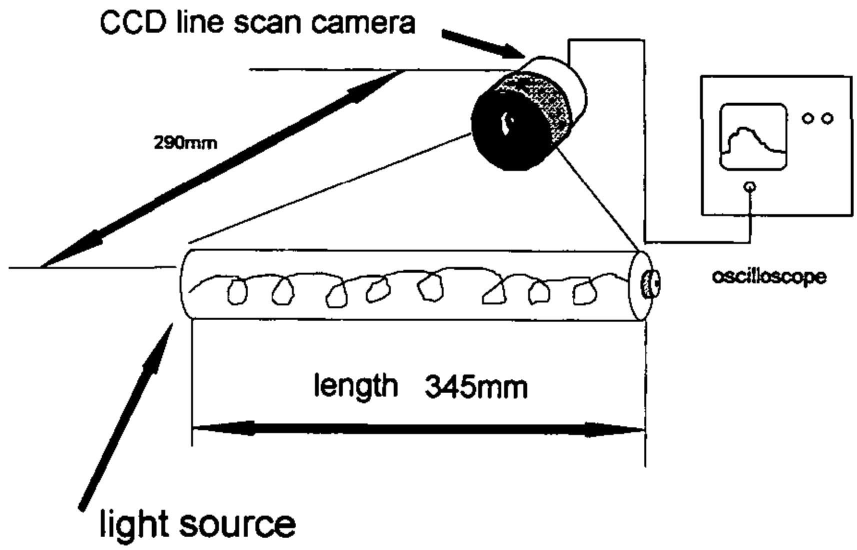
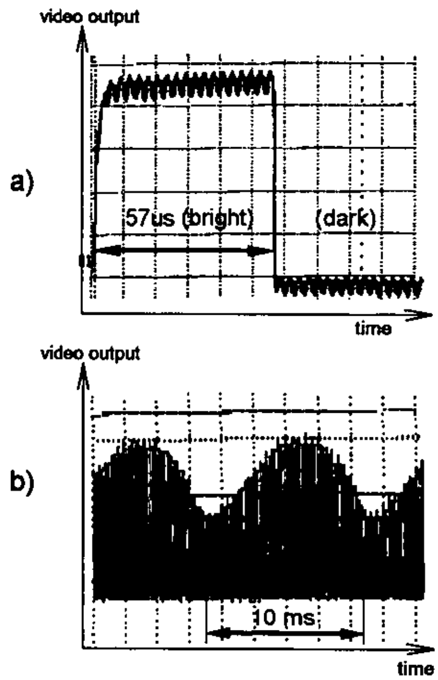


Figure 6.1 Layout of lamp test

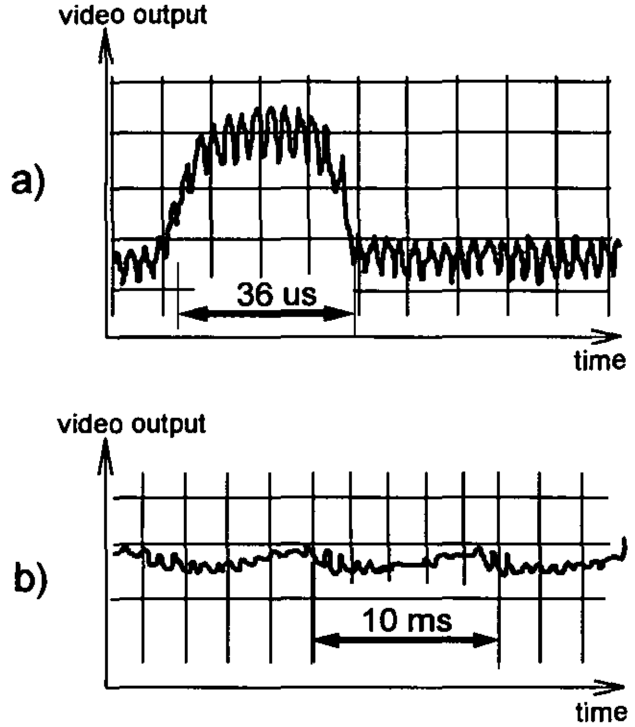


The type of lamp is Osram Dulux S 11W. The length of the tube is 195mm.
The light intensity distribution of the fluorescent tube with a traditional ballast was recorded by an oscilloscope.

a) recorded with 10us timebase to show the distribution along the tube. Note: the high frequency ripple in a) is caused by that the pixel frequency in the video is higher than the sampling rate of the oscilloscope.

b) recorded with 2ms timebase to show the 100Hz ripple.

Figure 6.2 Fluorescent tube with a normal ballast



The type of lamp is Osram Dulux EL 20W. The length of tube is 120mm.

The light intensity distribution of the fluorescent tube with a electronic ballast was recorded by the oscilloscope.

a) recorded with 10us timebase to show the distribution along the tube.

b) recorded with 2ms timebase to show the 100Hz variation.

Figure 6.3 Fluorescent tube with an electronic ballast driven by AC power

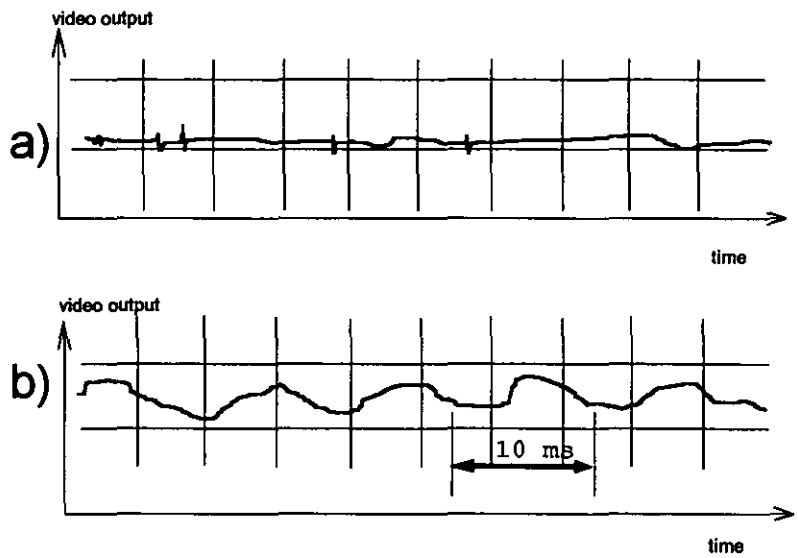


Figure 6.4 a: Light intensity of tube driven by DC
b: Light intensity of tube driven by AC

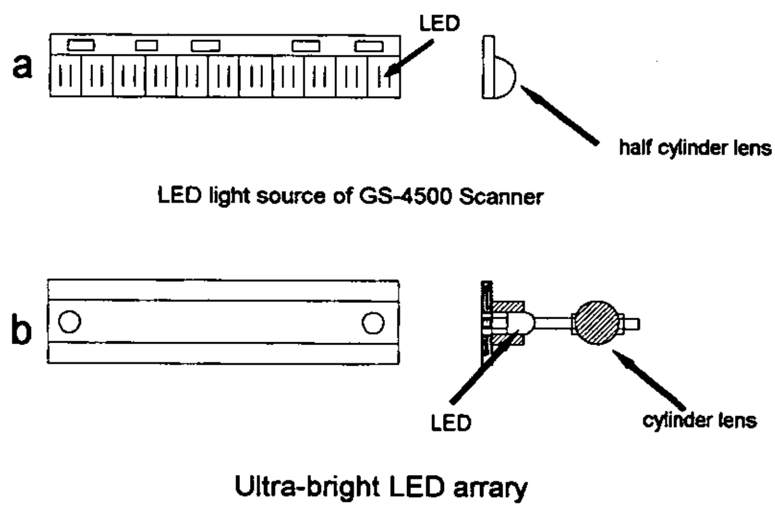


Figure 6.5 Structures of LED array

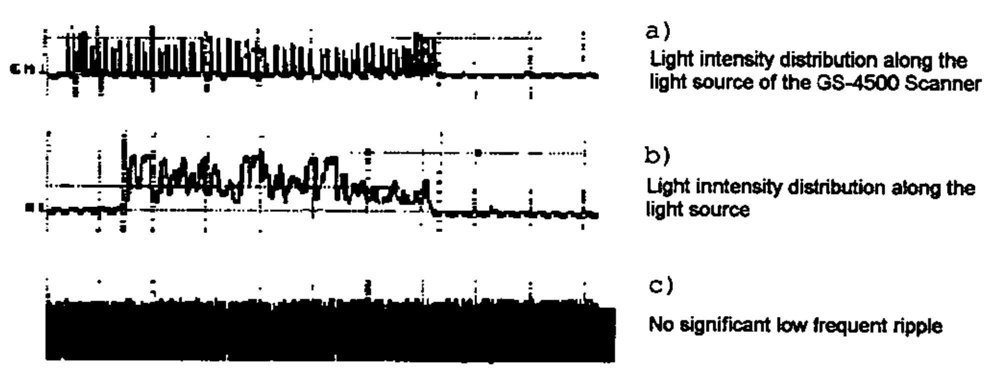


Figure 6.6 LED light intensity distribution

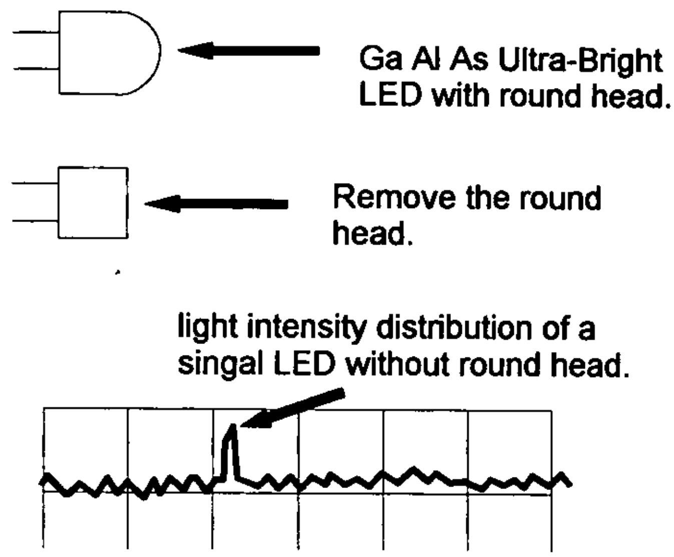
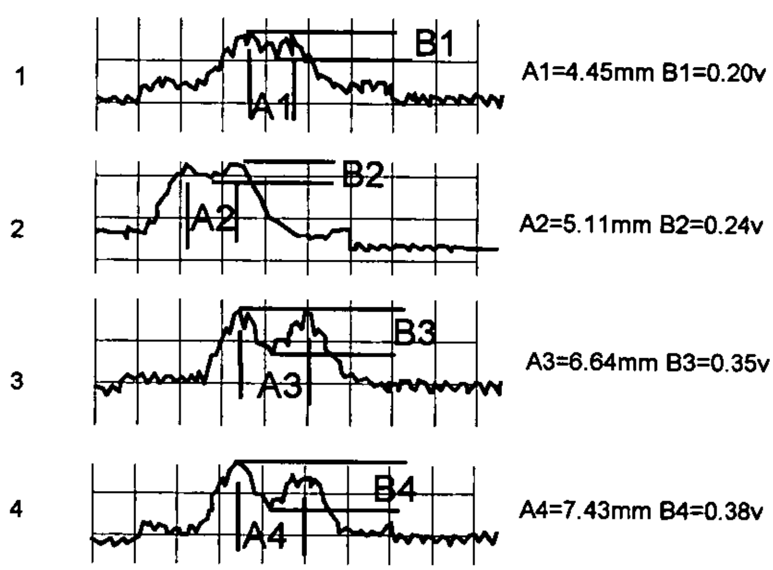


Figure 6.7 LED with and without round head



Test condition: distance between LED and CCD chip is 85mm; driving voltage is 5v; resistor is 200R0.
Note: B is the video output voltage.

Figure 6.8 Light intensity distribution of single LEDs

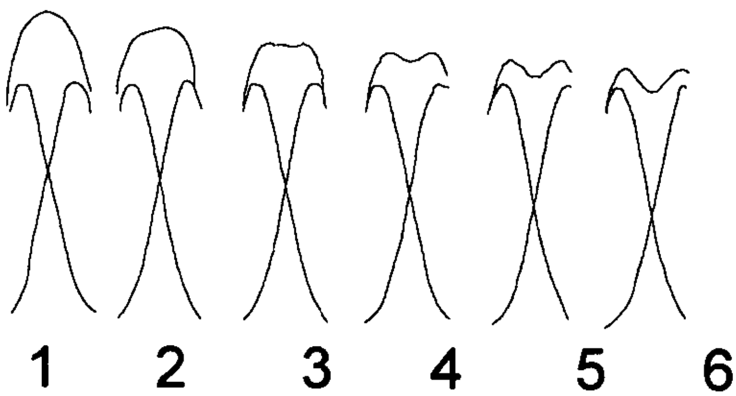


Figure 6.9 Uniformity vs. change of distance between two LEDs

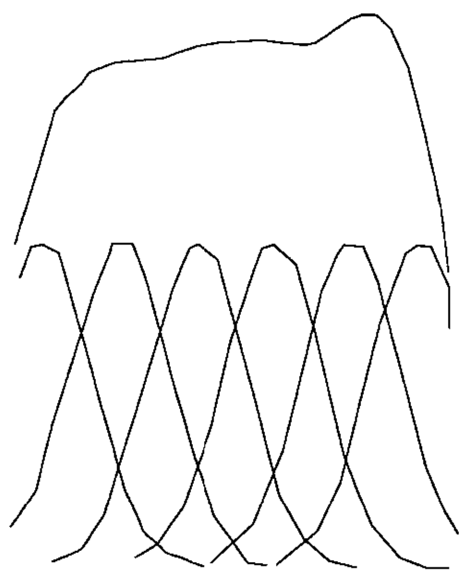


Figure 6.10 Change of uniformity

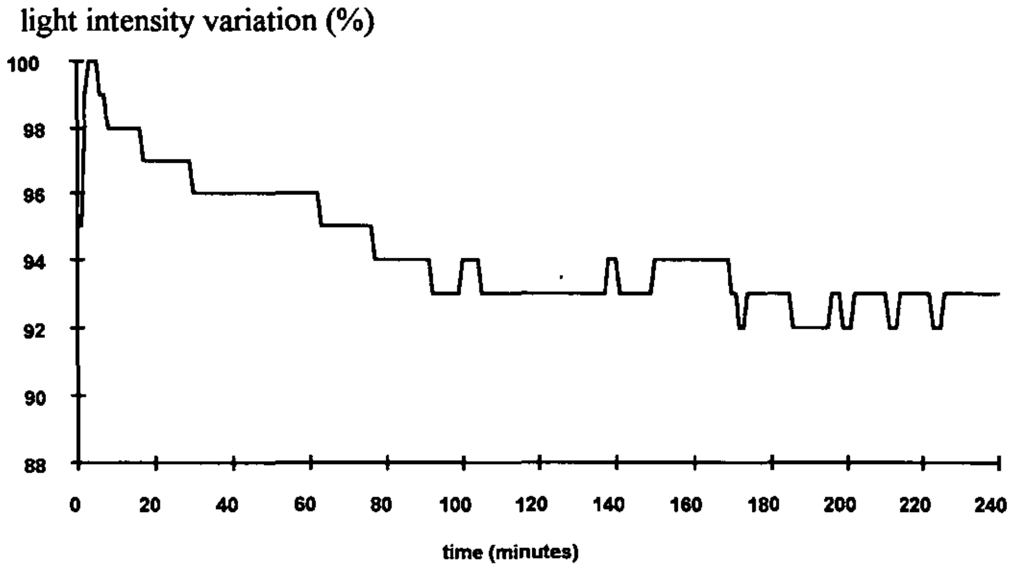


Figure 6.11 Light intensity change with time

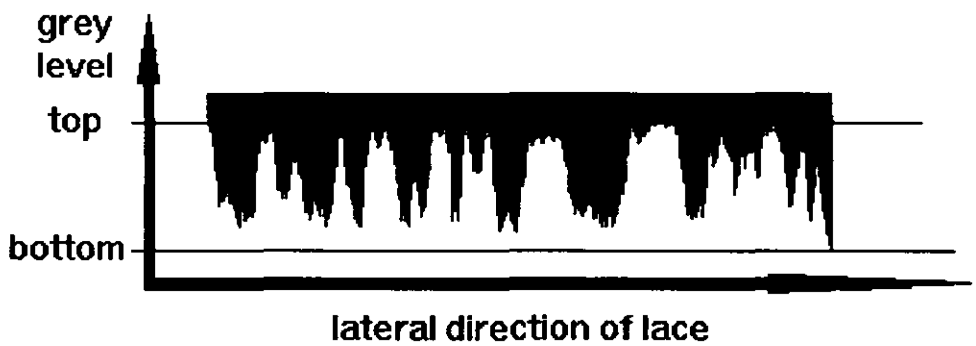


Figure 6.12 Grey level for one line of lace

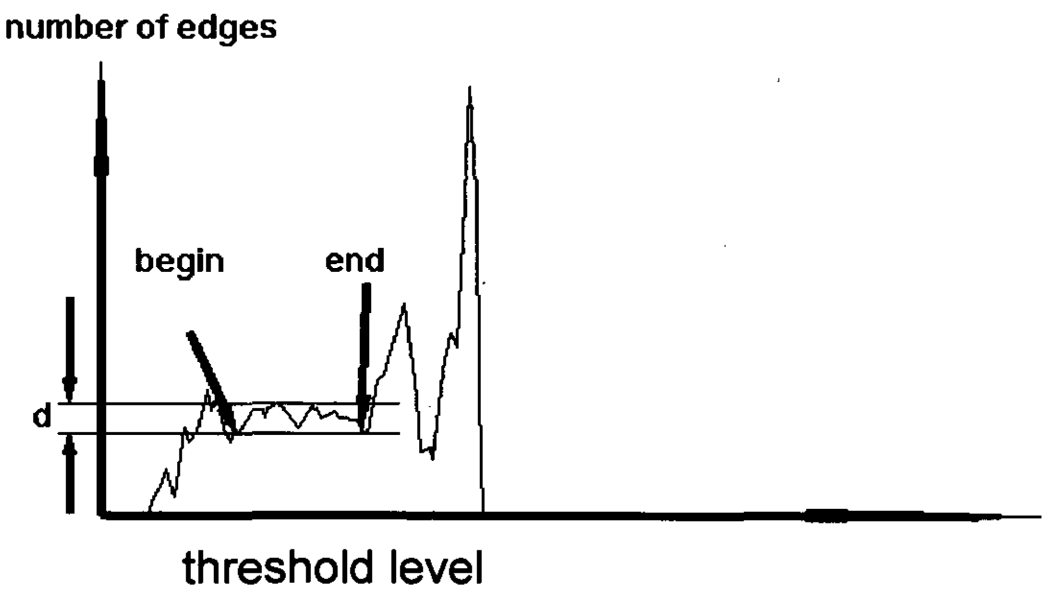


Figure 6.13 Edge curve

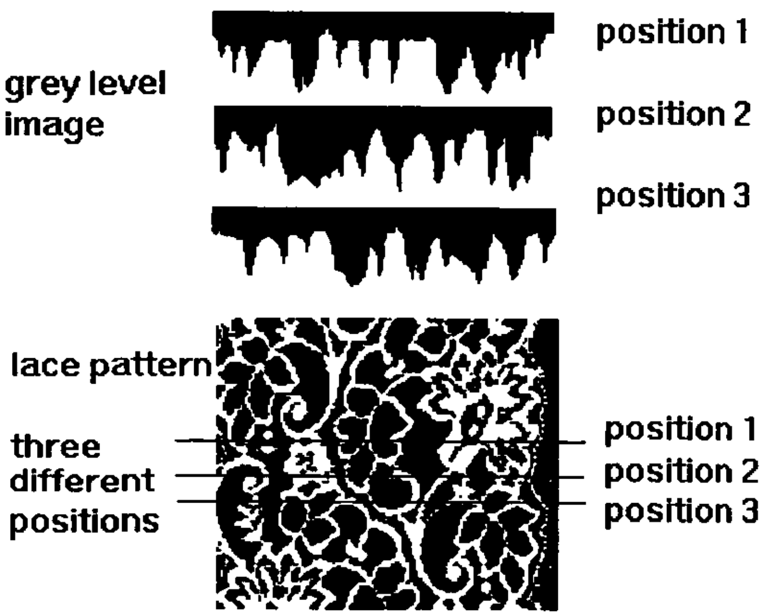


Figure 6.14 Test positions

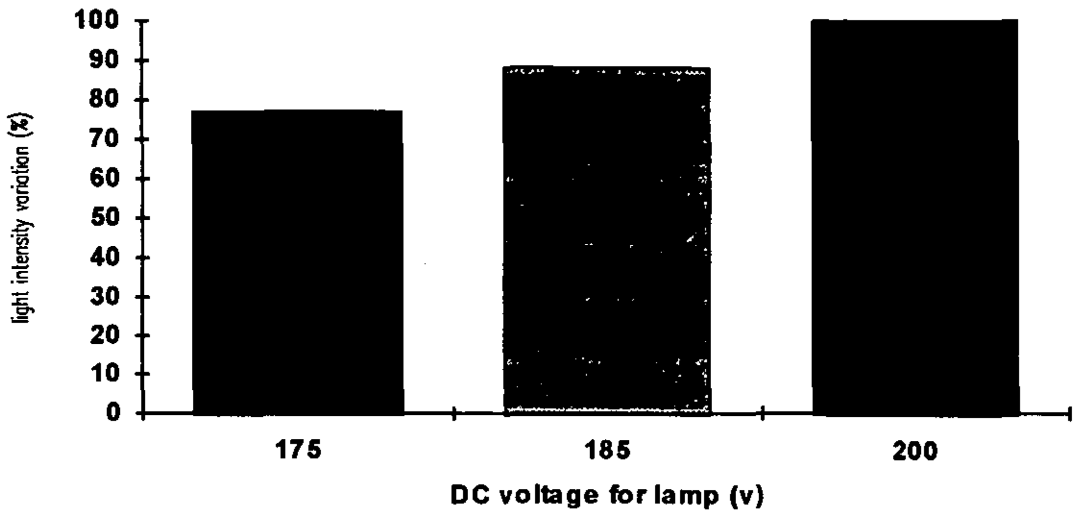


Figure 6.15 Test voltages

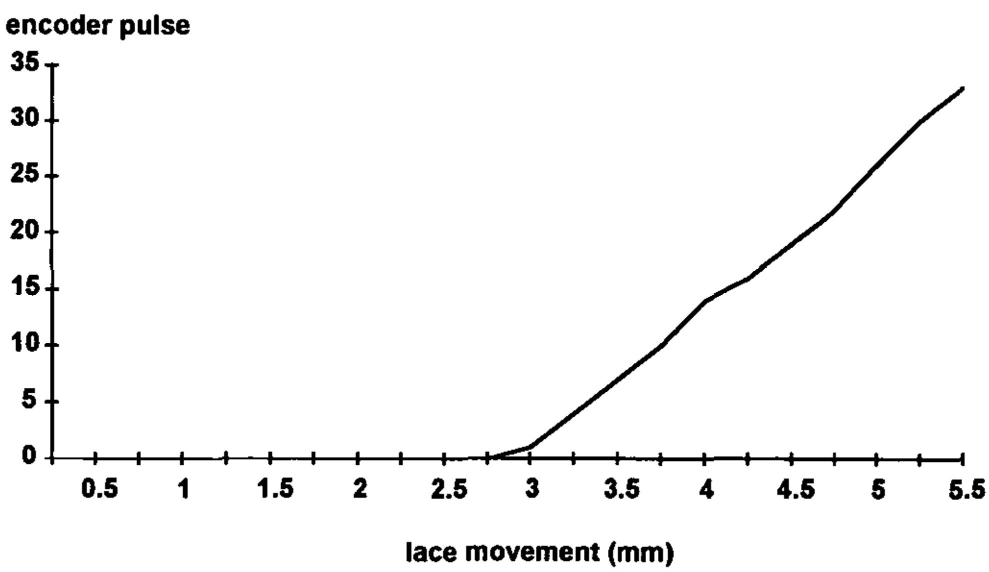


Figure 6.16 Relation between encoder and lace

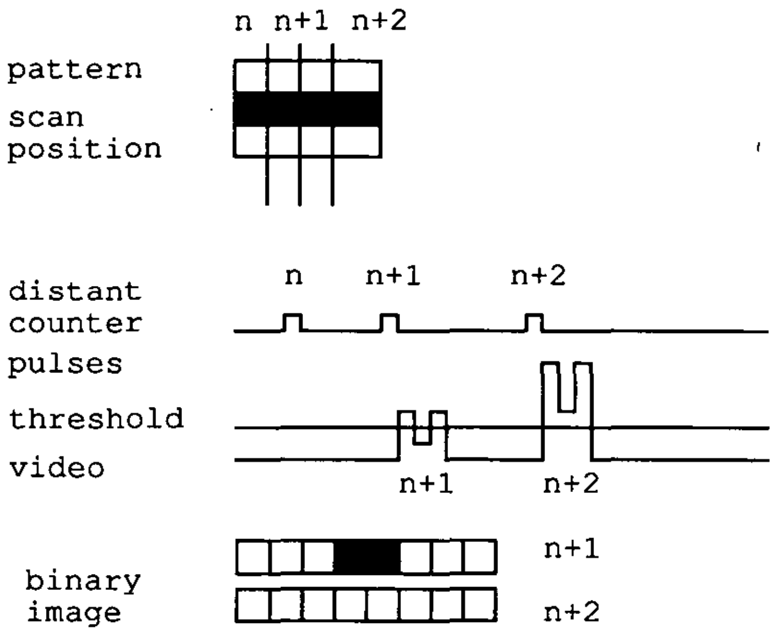


Figure 6.17 Variation of the image caused by integration

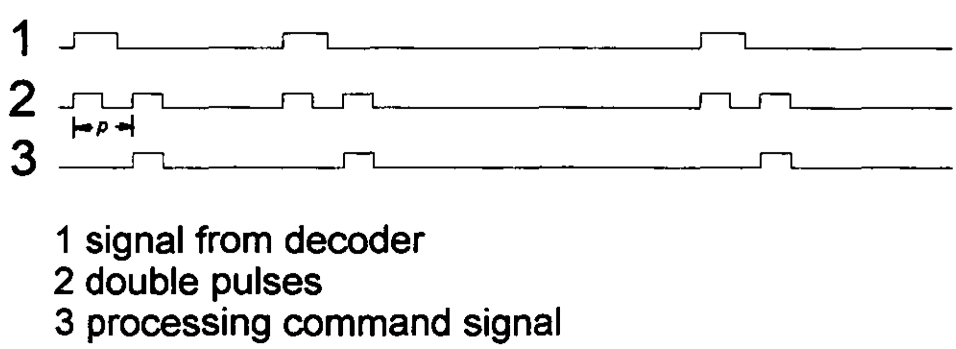
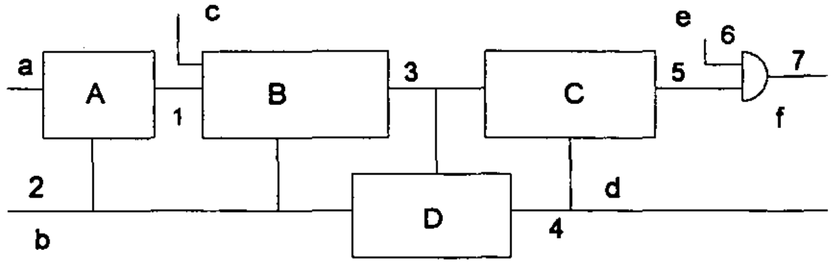


Figure 6.18 Timing of the double signal generator



Note: the number will correspond to the number on Figure 6.20.

A: decoder B: synchronizing with system clock
C: controller D: line scan signal generator
a: signal from encoder b: system clock
c: line scan from DSP d: line scan to camera
e: image from camera f: image for DSP

Figure 6.19 Principle of synchronising board

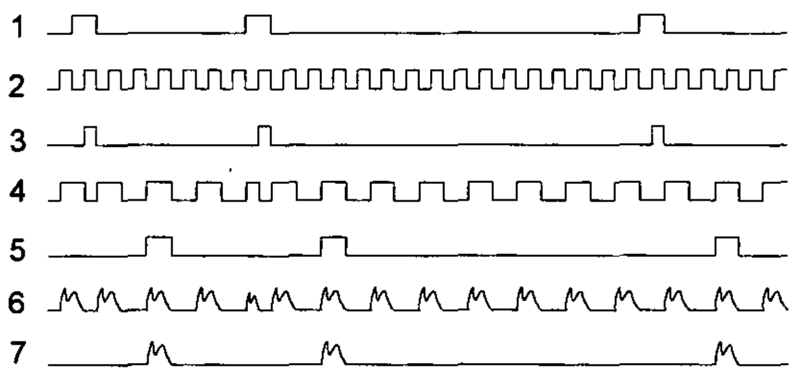


Figure 6.20 Timing of synchronising board

Chapter 7

ENHANCED PATTERN TRACKING SPEED

7.1 INTRODUCTION

Pattern tracking is the key part of this high speed vision system. Therefore, improvement of the tracking speed is an important research task. Two approaches to speeding up the tracking, optimised code and a front end correlator, have been investigated.

In section 7.2 the influence of the speed of the tracking on the speed of the whole vision system is analysed. In section 7.3 an approach to optimise the tracking computer program code is presented. In section 7.4 a hardware correlator approach to speeding up the tracking is investigated. In section 7.5 conclusions about speeding up the tracking are listed.

7.2 TRACKING SPEED

Let us consider the second vision system (see Figure 3.10). The functions of DSP1, DSP2, and Motorola 68020 are as follows. The DSP1 looks after the line scan camera and puts the image data into the global memory which can be accessed by all DSPs. The DSP2 does the tracking and puts the tracking result at the end of a queue (a memory block between DSP2 and 68020) and the 68020 sends the first data in the queue to the galvanometer via a D/A to control the cutting beam position. The length of the queue, q, is generally determined by the physical distance d between the camera zone and cutting zone and resolution of vision system, r (determined by an encoder, see chapter 6).

$$q = d / r \qquad \text{eq 7.1}$$

If we follow a line of image on its "travel" through the vision system, we can get a travelling path which illustrates the relationship amongst the parts of the vision system (Figure 7.1).

When the lace feeds at a constant speed v, the time, t_{lace}, taken for a particular part of the lace to travel from the scan zone to the cutting zone is given by equation 7.2.

$$t_{lace} = d/v \qquad \text{eq 7.2}$$

The whole processing time of a single image line in the vision system, $t_{process}$, is the sum of the processing times for the all functional blocks in Figure 7.1. There are q image lines between the camera zone and cutting zone, so the $q \times t_{process}$ must be less than t_{lace} (inequality 7.3). Otherwise the cutting beam will never hit the correct point of the lace continuously (supposing there is only one direction of movement of the lace).

$$q \times t_{process} < t_{lace} \qquad \text{eq 7.3}$$

As we discussed in chapter 3, the shorter the distance between scan zone and cutting zone, the less change of the lace after scanning. Unfortunately, the distance, d, will be constrained by the lace feeding speed v and the whole processing time of the vision system. If the t_{lace} in inequality 7.3 is replaced by equation 7.2 d will be determined by inequality 7.4a.

$$d > t_{process} \times v \times q \qquad \text{eq 7.4a}$$

(The smoke in the cutting zone is another constraint for the shortest distance between cutting zone and scan zone. Firstly the cutting smoke is quite thick, which will blur the image if the two zones are too close to each other. Secondly any extraction hood employed to contain the cutting smoke will interfere with the scan zone due to its size.)

The inequality 7.4a can be rewritten to inequality 7.4b to see effect of the $t_{process}$.

$$v < d/(t_{process} \times q) = r/t_{process} \qquad \text{eq 7.4b}$$

In inequality 7.4b we can see that the lace feeding speed v depends on the processing time $t_{process}$ and the system longitudinal resolution r. In order to see which part of the vision system is the biggest time consumer in $t_{process}$, let us divide $t_{process}$ into t_{scan}, t_{data}, $t_{tracking}$ and $t_{transform}$.

t_{scan} is the time between the trigger signal from a distance counter (see chapter 6) to the first image word sent to DSP1 which is 105.6μs. t_{data} is the time from the input trigger of DSP1 to set the flag in global memory to broadcast the new line of image to DSP2 which is 200μs. $t_{tracking}$ is the time from getting a new line signal to putting the tracking result into the global memory which is 2450μs. $t_{transform}$ is the time in which the 68020 gets tracking data in the DSP board, transforms it and sends it to the galvanometer, which is around 20μs. Up to now, the reader can clearly see that $t_{tracking}$ *is* the biggest portion of the $t_{process}$. Therefore $t_{tracking}$ dominates the lace feeding speed v.

Parallelization, one way to speed up the vision system, will be discussed in future work.

Investigations have been carried out to increase the speed of tracking. There are two approaches to speed up the tracking. One is to optimise the DSP code to make it faster but there is a limitation for a DSP; another is to employ a hardware correlator as a front end accelerator.

7.3 OPTIMISING THE DSP CODING

The tracking code for DSP_2 can be functionally divided into two parts. One is to perform the binary cross correlation between map and image data; another is to do IIR filter and longitudinal registration decision. Of the 2450μs execution time per line of input image, approximately 2250μs is spent performing the cross correlation whilst the rest of the code consumes the remaining 200μs**.

Clearly, reducing the time taken by the cross correlation is the primary target for optimising the DSP coding.

** The excution time quoted here is obtained from the simulation results of the Motolora simulation packege, SIM56000.EXE.

Because there is no operating system on DSP board, the coding is a hardware dependent task. So before describing how to optimise the DSP coding, it is necessary to introduce the Motorola DSP 56001 and the DSP board.

7.3.1 DSP Architecture

DSP is a special microprocessor for digital filtering calculations. The typical calculations are multiply and add, which can be done in the same instruction cycle supported by its own hardware architecture (Cooling 1991).

Figure 7.2 shows the difference between the internal data, address bus and external bus in a DSP. The X, Y internal memory use different data buses but the X, Y external memory must share the same data bus. This difference affects the code execution time. For example the (MAC X1, Y1, A X:(R0),X1 Y:(R4),Y1) is a parallel instruction set, which does $A=A+X1*Y1$, moves data from memory location pointed to by X:(RO) to a data register X1 and moves data from memory Y:(R4) to another data register Y1 in one instruction cycle. If R0, R4 point to the internal memory, the instruction execution time is 2 clock cycle because all internal memory sources can be accessed at the same time. But if R0, R4 point to the external memory, the execution time will be much longer because only one external memory access can occur per bus request (Motorola 1989).

7.3.2 Key Points for Optimising Tracking Coding

The main methods to be used to optimise the tracking coding on DSP are as follows:

1. Put the most frequently used data into the internal memory;
2. Use the parallelism of instructions;
3. Use reference registers (R0-R7) instead of immediate data transfer.

After applying these coding rules the execution time for the cross correlation is reduced to around 800μs. The total tracking execution time is of around 1ms/line and hence tracking at 330mm/s (or very nearly 500mm/s with a 40MHz DSP) is possible.

7.4 Implementing Tracking by Using Front End Correlators

After the DSP code for the incremental tracking has been optimised several times, the execution time for every incoming line of the video data is reduced from 2.5ms to 1ms using a 27MHz Motorola DSP56001, and hence tracking at 330mm/second with a line resolution of 3 lines/mm is achieved.

In order to further increase the tracking speed, an approach based on the Harris HSP45256, binary correlator, has been investigated. The correlator card has been designed, built and tested and the DSP code to address the card has been written and debugged.

7.4.1 Working Principle of the Card

In order to describe the working principle of the card a simplified schematic diagram of the card with one HSP45256 correlator is shown in Figure7.3. There are three HSP45256 correlators on the complete card. The HSP45256 is capable of performing 256 stage binary correlations or 32 stage 8-bit correlations (also other configurations in between) by suitably configuring its eight 32-bit registers via A0-2 and DCON0-7. Here the correlator is configured as a 128 stage 2-bit correlator (see Figure 7.4). The mask data, which determines the weighting table for correlation, is sent to HSP45256 via DCON0-7 pins. The map data is sent to DREF7 via a 24 bit latch and parallel-in-serial-out shift register. The video data is sent to DIN7 and DIN3 pins via an external serial-in-serial-out shift register. The result of correlation can be calculated by equation 7.5.

$$Correlation = \sum_{i=0}^{127} \overline{D_i \otimes R_i} \times W_i \times M_i \qquad \text{eq.7.5}$$

where the D_i is the video data in bit i of the data shift register, the R_i is the map data in bit i of the reference register, the M_i is the masking table in bit i of the mask register, and the W_i is the weighting function according to the configuration of the HSP45256 (Harris 1991). The *Correlation* will be presented at the pin CASOUT0-7. In every clock pulse the data in the data shift register will be shifted one side step and a new video data pixel will be put into the last cell of the shift register. At the same time a new correlation result will be presented on the CASOUT0-7.

In practice there are two problems to be solved about how to interface to the Motorola DSP56001 before the HSP45256 can be used for the tracking. One problem is that the DSP cannot read correlation results, which are 8-bit quantities at 20MHz (i.e. the camera clock rate), another problem is how to pick up the required portion of the correlation results that is around the cutting path.

The maximum clock rates for the two available versions of the HSP45256 are 25MHz and 33MHz. The clock rate for the DSP is 27MHz, therefore the map data will be clocked into the reference register of a correlator at this rate. The 33MHz HSP45256 correlator is chosen to accommodate this data rate. The output rate of correlation results is the same as the video data rate, namely 20MHz, but the highest data-in rate for the DSP is 5MHz (Data β 1991). For this reason, a 'clock gear' is designed to let the card work in two phases. In the first phase, the video data is clocked into an external shift-register and the internal shift-register in the correlation chips at 20MHz clock rate and no 'strobe' pulse is issued so that the correlation results being produced in this period are not sent to the DSP. In the second phase the correlator clock is reduced to a rate commensurate with the DSP ability to accept the results from the correlator and 'strobe' pulses are issued so the correlation results are sent to the DSP.

The 'clock gear' combined with a programmable counter has been used to choose the required portion of the correlation results of the current video line. According to the tracking algorithm, the centre of the interesting search area, Y, can be predicted by equation 4.8.

So the correct portion of the current line for the correlation can be described by equation 7.6.

$$Y - l/2 + d \le y \le Y + l/2 + d \qquad \text{eq.7.6}$$

where Y is the predicted centre of the current line of the video data in the DSP memory, y is the portion of the current line of the video data, l is the length of the portion to be searched for, and d is the origin difference for a line of video data between that stored in the DSP memory and used for the correlator. The principle of using the 'clock gear' and programmable counter to choose the required portion of the current line is the following. The length of the effective internal shift register plus the external shift register equals l, and both internal and external shift registers are driven by the same clock from the 'clock gear'. Before the line scan pulse comes, the programmable counter has been set to $Y+l/2+d$. Once the line scan pulse arrives, the counter is counted down and the output of the 'clock gear' is 20MHz. As soon as the counter reaches 'zero', the 'clock gear' changes to low gear and at that moment the required portion of the current line just fills the whole area of the effective internal and external shift registers. At that time a first valid correlation result will be sent to the DSP. Then every following clock will shift one bit of video data in the external shift register into the correlator and produce a valid correlation result to be sent to the DSP until all the data, which has been stored inside the external shift register before the 'clock gear' changes from high to low, is shifted into the correlator.

The timing diagram for the correlator is shown in Figure7.5.

According to the tracking algorithm, a current line of the video data should be correlated with three consecutive lines of the map. For this reason in the real correlation card there are three correlators working in parallel. The schematic diagram is in the Appendix 6.

7.4.2 Primary Test

The experiment arrangement is as shown in Figure 7.6. The DSP1 is used to control the camera and the interface board, and the DSP4 is used to communicate with the correlators. The experiment sequence is as follows:

1. DSP4 sets the programmable counter, configures the correlators, and sends map data and weight table to the correlators. The DSP4 sets itself ready to accept the data from the correlators.

2. DSP1 sets the threshold level on the interface board. Then a single line scan signal is issued by DSP1.
3. As soon as the interface board gets the valid video data, the valid video data will be sent to the DSP1 and sent to the correlators individually.
4. After the line scan pulse has arrived, the correlator board will put the required portion of the current video data into the internal and external shift-register. Then the clock rate changes to lower rate and the correlation results are sent to DSP4.

The aim of the experiment is to test the functions of the correlation card. The following is one set of the experimental results, the data is presented in the hexadecimal format.

map in correlator0: 000000 000000 000000 000000
map in correlator1: ffffff ffffff ffffff ffffff
map in correlator2: 007fd5 fffffd 7fffff 5fff5d
weighting table (for all correlators):
1 1 1 1 1 1 1 1 1 1 1 1

video data in DSP memory

dd7fff	ffff5f	d57d7d	d01555	57ffff	dfffff
fffffd	7f7fff	ffffff	ffffff	ffdfff	ffffff
ff7fff	ff**ffd5**	**555fff**	**ffffff**	**ffffff**	**fdffff**
fffff5	5557ff	ffffff	ffffff	ffffdf	f5ffff
ffffdf	ffffd5	7fffff	fd7fff	f7fffd	000000

correlation data from the correlator0

0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 09 09 08 08 07
07 06 06 05 05 04 04 03 03 02 02 01 01 01 01 01

correlation data from the correlator1

56 56 56 56 56 56 56 56 56 56 56 57 57 58 58 59
59 5a 5a 5b 5b 5c 5c 5d 5d 5e 5e 5f 5f 5f 5f 5f

correlation data from the correlator2

4b 43 4b 45 4f 47 4f 49 4d 4b 4d 4c 4e 4d 4f 4e
50 4f 51 50 52 4f 51 4e 4e 4d 4d 4c 4c 4c 4c 4c

The data of the video data in the DSP memory being highlighted, which is 128 bit wide, has been chosen to correlate with the maps in each correlator, which is 96 bit wide. The correlation results can be checked by equation 7.5. For example, the first 96 bits of the video data, ffd5 555fff ffffff ffffff fd, is correlated with the map data, 007fd5 fffffd 7fffff 5ffff5d in the correlator 2 to produce the first correlation result in the correlator 2 in the following way:

video data ffd5 555fff ffffff ffffff fd
in binary
1111111111010101 010101010101111111111111
111111111111111111111111 111111111111111111111111 11111101
map data 007fd5 fffffd 7fffff 5ffff5d
in binary
000000000111111111010101 111111111111111111111101
011111111111111111111111 010111111111111101011101
exclusive-nor
000000000101010101111111 010111111111111111111101
011111111111111111111111 010111111111111101011111
correlation=$\sum$ 1's = 0x4b

Then a program using the same method has been coded to calculate the correlation results from the video data being stored in the DSP memory and the map data being sent to the correlator, and the simulation results of the program are the same as the correlator results being sent to the DSP4.

7.4.3 Hybrid Tracking Coding

The tracking program contains several functional blocks (Figure 7.7).

Two things, the weighting function and map buffer, should be mentioned, while the other function codings are straightforward.

7.4.3.1 Weighting Function

The weighting function in chapter 4 can be implemented in the HSP 45256 by setting the mask register (see Figure 7.4). For example Figure 7.8 shows a setting of the mask registers to realise 96 bit matching width with weights 1-2-3-3-2-1.

The setting format of the mask register depends upon the configuration of HSP45265. As the 128 stage 2bit correlator which is configured for tracking, the mask registers form 2 parallel 128bit mask registers and the mask settings for two mask registers are input from DCON7 so the same bit on two mask registers must have the same mask value 1 or 0 which cannot perform the weighting function in Figure 7.8a.

One way to set every bit individually in the mask registers is to take advantage that the change of configuration of the correlator will not change the contents in the registers. The method is as follows. Firstly the correlator is configured as a 8 row by 32 correlator in which each of the 8 correlators has its own mask input (see Figure 7.8b). The rows of the matrix in Figure 7.8b represent the mask registers. The columns in the matrix represent the bits of the mask registers. Secondly each cell of the matrix is filled with 1 or 0 according to the requirement of weighting function (see Figure 7.8). Thirdly the data in Figure 7.8b are seen as a hexadecimal data column by column and sent to DCON0-DCON7. The DCON7 is the most significant bit (MSB) and the DCON0 is the lest significant bit (LSB). For example the data in column 0 is 01101100, that is 6c (Hex). Finally the correlator is configured as a 128 stage a 2 bit correlator in which the mask register will be the same as Figure 7.8a.

7.4.3.2 Map Buffer

One advantage of the HSP45256 is that the reference registers are double buffered (Harris 1991) which allows new reference data to be loaded while the current correlation is in progress. This can be used to preload map data into the correlators for future correlation.

The reason for this is that after the sequence of map lines in the correlators for current correlation has been decided the sequence for preloading map data is predictable whatever the decision which results from the correlation, such as "wait", "step", "jump". For example, Figure 7.9 shows the map in correlators before the correlation. Figure 7.10 shows the map in correlators after "wait", "step", "jump".

If this map preloading method is used directly, the problem of the program efficiency will arise because the DSP will control which correlator should be sent map data, and which correlator should be read from according to the previous map in the correlators. This takes time.

If the physical correlator0, correlator1, correlator2 are separated from the logical cor0, cor1 and cor2, the DSP is only interested in the logical correlator. A ring-counter is used to manage the physical correlators' input and output according to the logical correlator of the DSP, so the efficiency of the tracking can be improved. The logical correlator0 refers to a physical correlator that has the lowest sequence number of the map in its reference register, and so on. For example, in Figure 7.9, the logical correlator0 is the physical correlator0. In Figure 7.10 after "step" or "jump" action the logical correlator0 is the physical correlator1 or correlator2 respectively.

7.4.4 Performance of Hybrid Tracking

A Hewlett-Packed Automatic Synthesizer (HP3330A) was used to measure the execution time of the hybrid tracking program. The time from the start of 'make decision' block to the end of 'set counter send map' block is 120μs (see Figure 7.7). The fastest execution time for every new image line is 226μs (DSP time + Correlator time), which is fast enough for 1m/s speed with 3 line per mm resolution. During the real lace cutting tests, the hybrid tracking demonstrated over 1m/s tracking capacity.

7.5 CONCLUDING REMARKS

The methods for speeding up the tracking have been studied in this chapter. The coding optimisation can push DSP to track near 1000 line/s. This should be potentially very good because the operational speed of the microprocessor will increase dramatically in the near future. The cost to port the tracking algorithm to other microprocessors to speed up the tracking is smaller than the cost to design and make a hardware for the same purpose.

The correlator front end tracking is capable of tracking 1m/s lace movement speed and faster than the pure software tracking. The research and development cycle is still shorter than a pure hardware solution.

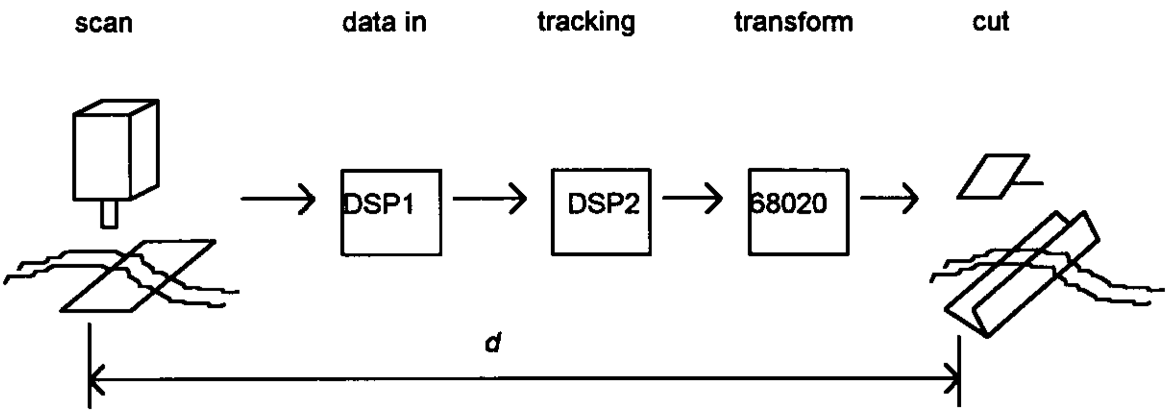


Figure 7.1 Functional blocks of the vision system

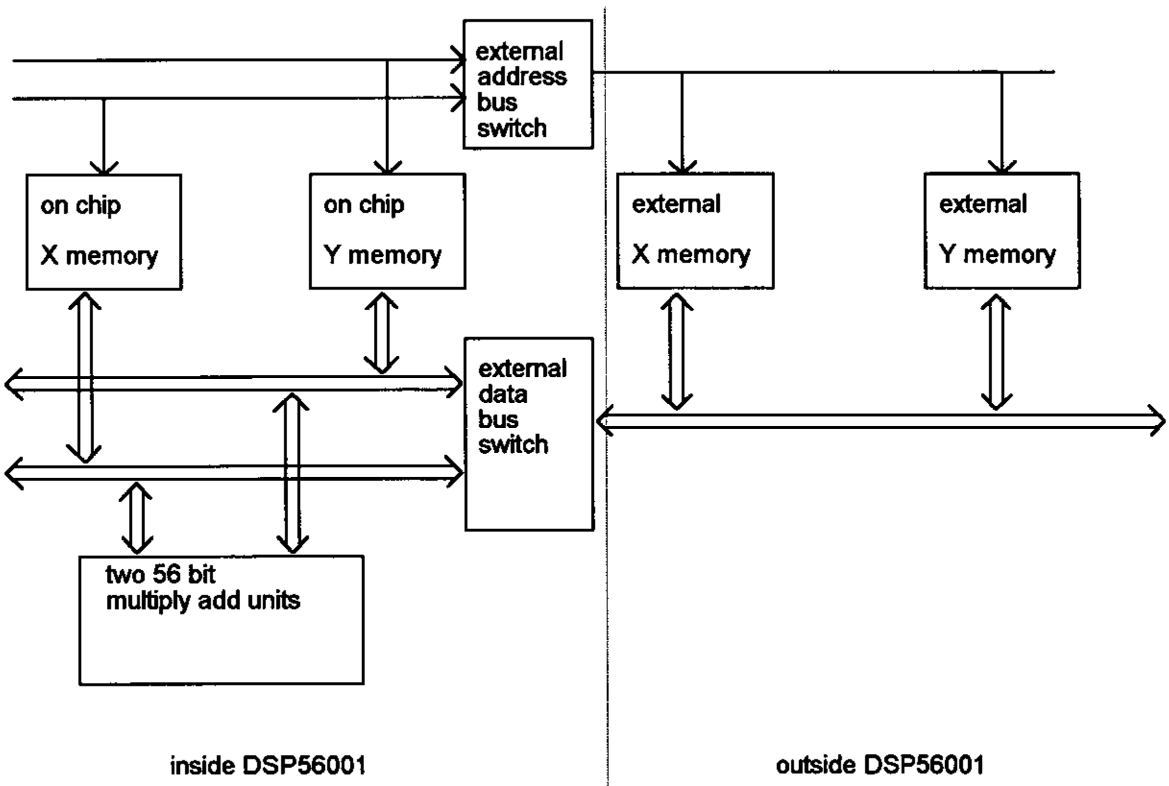


Figure 7.2 Bus organisation

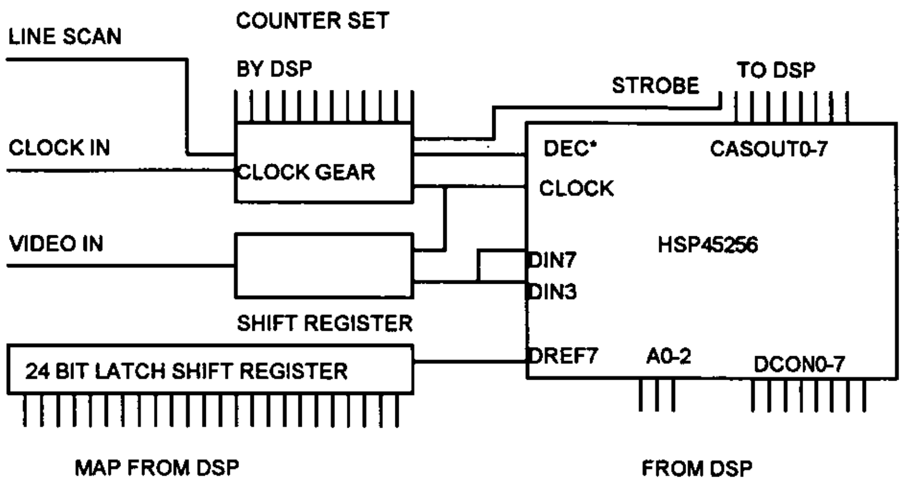


Figure 7.3 Diagram of one correlator

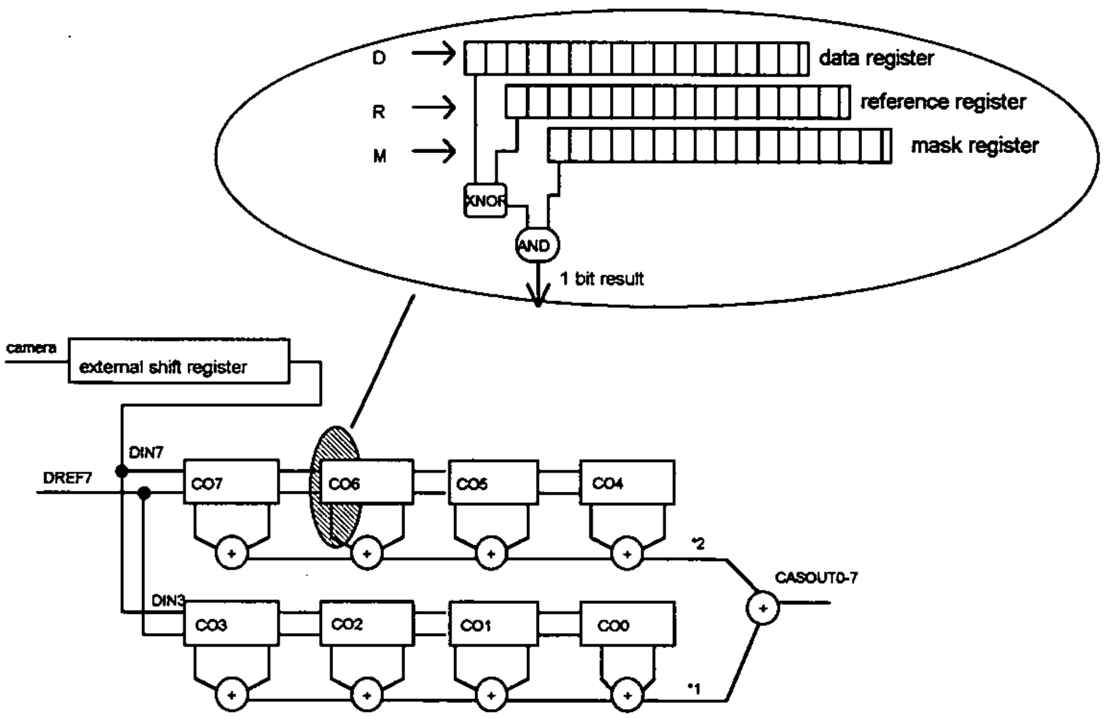


Figure 7.4 Correlation

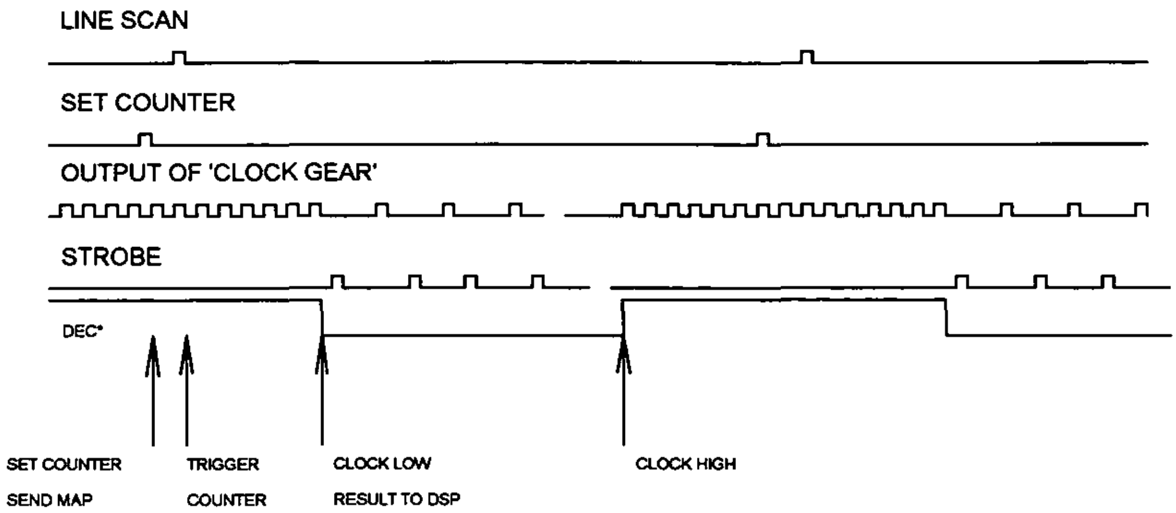


Figure 7.5 Timing diagram for correlation

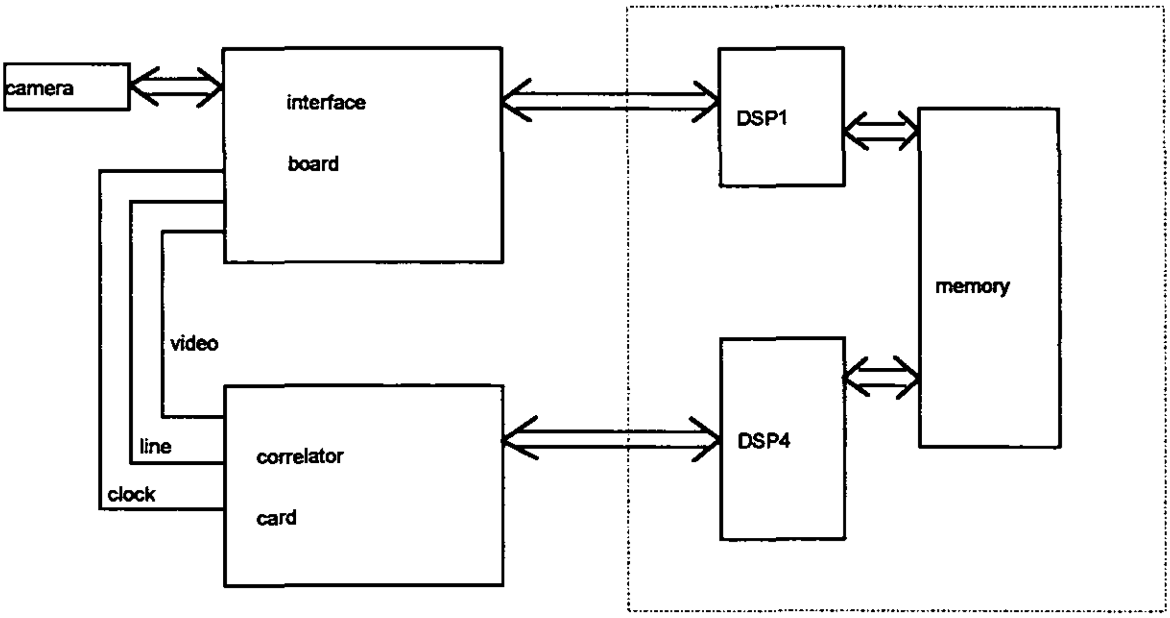


Figure 7.6 Experiment layout

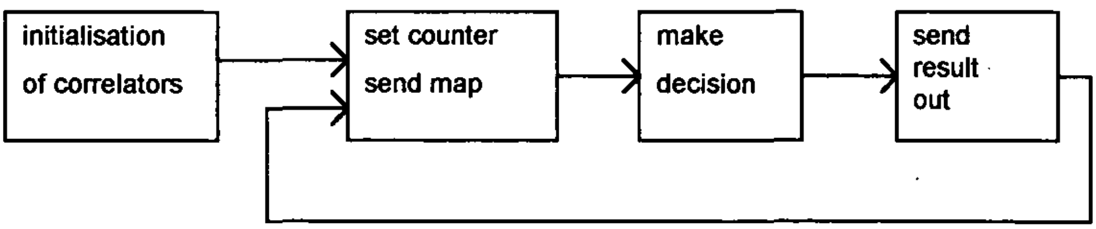


Figure 7.7 Functional blocks of hybrid tracking algorithm

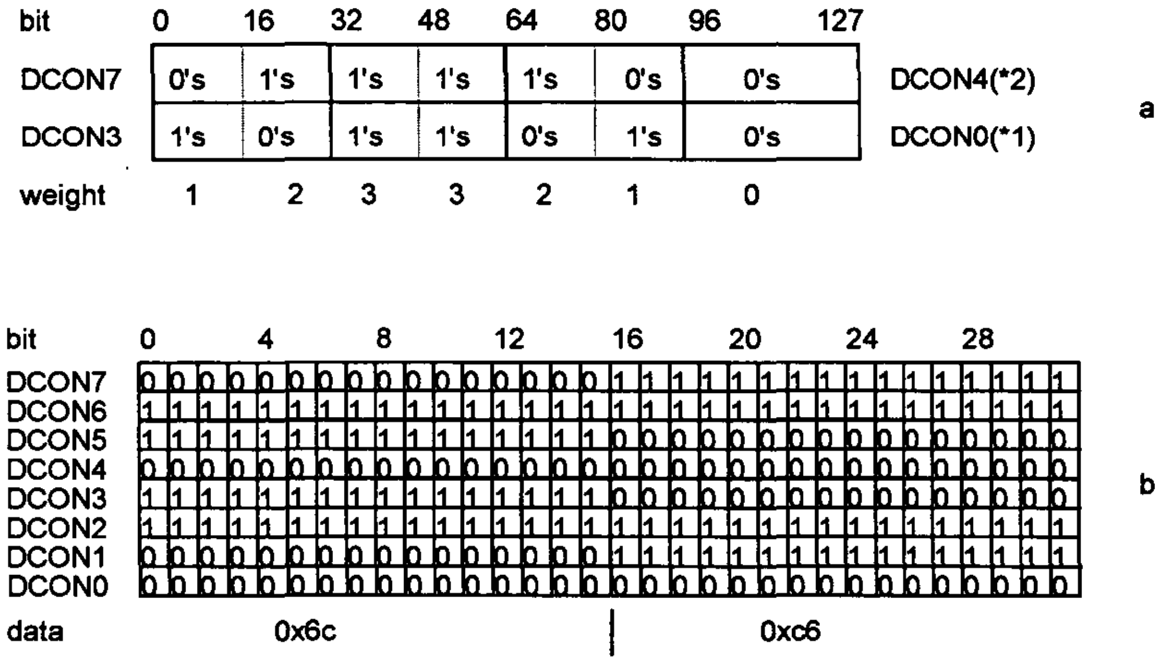


Figure 7.8 Weighting function

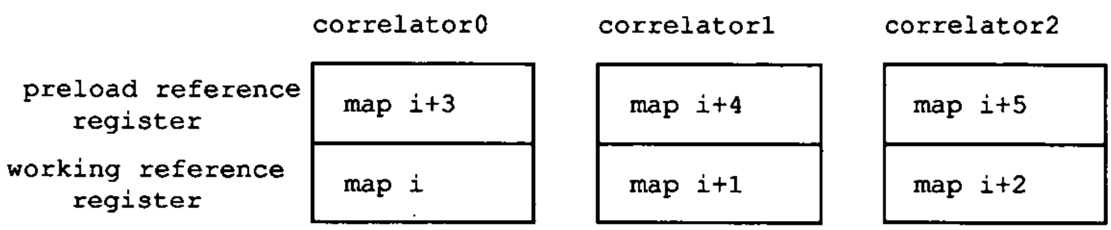


Figure 7.9 Map buffer

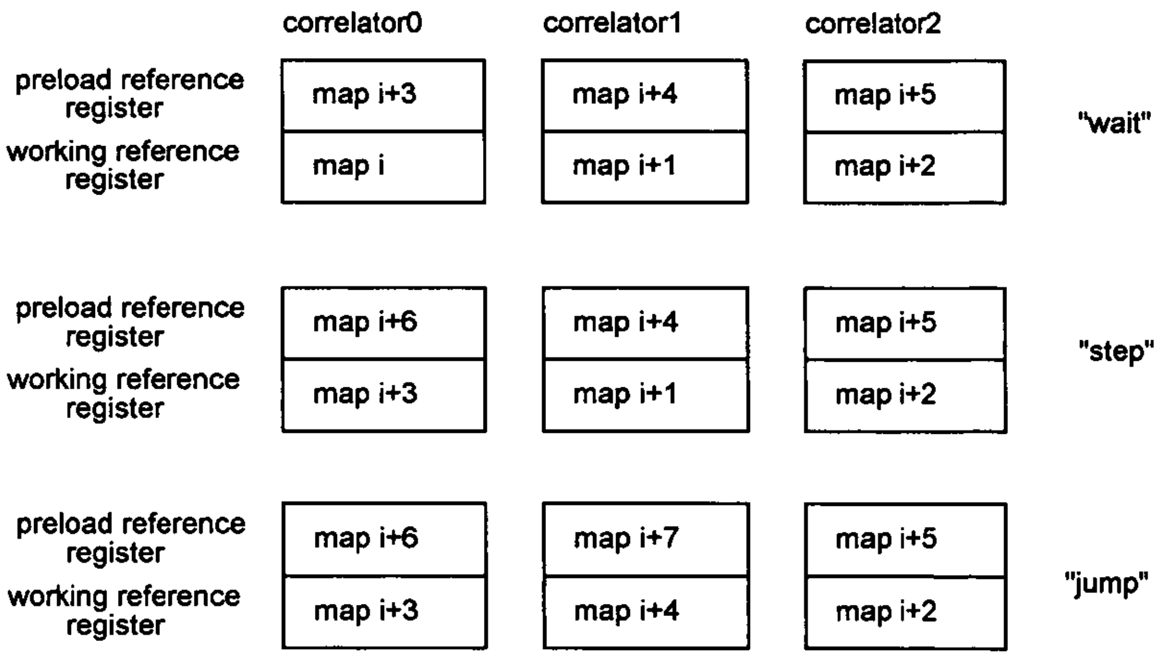


Figure 7.10 Map buffer after action

Chapter 8

PERFORMANCE ANALYSIS

8.1 INTRODUCTION

In addition to cutting lace which has pattern deformation, the speed of cutting and the capability to cut multiple paths are two other requirements of this research. Therefore, the latter requires parallel processing of separate tracking algorithms. The vision system must be able to track a single path or multiple paths in parallel. This ability needs to be integrated with the research into the laser, optics, and lace transportation system. The research work to increase the speed of tracking has been presented in chapter 7. The testing of the integrated system will be presented in this chapter, this is based around the testing of the prototype machine (Figure 3.17 & Figure 8.1). The research work on multiple tracking will be described in section 8.2.

Cutting lace with the required accuracy is another requirement of this research. There are many factors affecting the final cut result; tracking error is one factor for consideration.

Let us consider the definition of cutting tolerance. Lace is a two dimensional object and the cutting path is a two dimensional curve hence the cutting tolerance is defined in two dimensions.

An ideal cutting path, *Pid(x)*, lies in the x y coordinate plane. A circle with δ radius and centred at *Pid(x)* moves along the ideal cutting path; two parallel curves in both sides of the path produced by movement of the circle form the tolerance band; any cutting point within the tolerance band is acceptable (see Figure 8.2). The definition of tolerance, *T(x)*, band is given in equation 8.1.

$$T(x) = Pid(x) \pm \Delta(x) \qquad \text{eq. 8.1}$$

where $\Delta(x)$ is determined by the curvature and gradient of the *Pid(x)* at point (x, y).

Lace is a deformable material so the ideal cutting path *Pid(x)* does not equal the reference path, *Pma(ξ)*, in chapter 4. The ideal cutting path *Pid(x)* can be defined in two ways. One is to define it according to *Pma(ξ)*, which can assess the difference between found path, *Pim(x)*, and *Pid(x)*, which is the tracking accuracy, within the vision system; another is to define it according to real lace, by which the cutting error includes all the other errors in the system. Since the magnitude of the tracking error is in the similar band of the magnitudes other errors, the latter method is difficulty to quantify the pure tracking error. But it is useful to judge the cutting error of the prototype.

In section 8.3 a method to transfer tracking results to D/A data used in test rig 3 (Figure 3.16) and the prototype (Figure 3.17 & Figure 8.1) is presented for guiding laser to cut lace. Investigation of cutting error is discussed in section 8.4. Further analysis of cutting related errors is presented in section 8.5. Some conclusions are drawn in section 8.6.

8.2 MULTIPLE TRACKING

A lace cutting machine with a multiple beam cutting ability will be able to increase the production in direct relationship to the number of heads. The economic and technical advantages of a multiple beam machine are as follows:

1. Increase productivity;
2. Lead the machine development towards re-entrant cutting, a potential new market.

A vision system with twin-path-tracking ability is an example of multiple tracking. The research aim of multiple tracking is to establish how to configure the vision system to realise multiple tracking and how to solve the special problems that occur for the multiple tracking concept.

8.2.1 Configuration of the Vision System for Twin Beam Cutting

Vision system 2 and 3 (Figure 3.10 & 3.12) are for single beam cutting. These include a line scan camera, a DSP board on which one DSP acts as an image processing engine, an interface board between the camera and DSP, and a host computer. There are several ways to configure a vision system to realise multiple tracking:

1. Duplicate the existing vision system (see Figure 8.3a);
2. Duplicate the camera and the interface board (see Figure 8.3b);
3. Use an extra DSP in the existing system (see Figure 8.3c).

The choice of the vision system configuration will depend upon the cost, the viewing range, and the software development.

If the distance between the two cutting paths is so large that a single camera cannot see both paths at the same time with a reasonable resolution or that two lighting sources must be used, the configuration of Figure 8.3a or Figure 8.3b is better than the configuration of Figure 8.3c in which the limitation of the camera viewing range and the single thresholding level for the analogue to digital conversion cannot cope with a large viewing range demand or two different illuminations unless some optical arrangement or extra control method is applied. If the two tracking paths are inside the viewing range of a single camera and the illumination is even across the viewing range, the configuration of Figure 8.3c is more economical than the others as it requires less hardware.

To judge the three configurations from the software development point of view will mainly depend on how difficult it is to rewrite the DSP processing code to suit the multiple tracking. The vision software includes two parts: 1. DSP code; 2. host processor code. Generally speaking, the software work to develop multiple tracking involves both DSP code and host processor code. Usually developing the code for the host processor can take advantage of the operating system and higher level language and software development facilities on the host computer, such as DOS, OS9, C, Debug. The researcher can put more efforts into the data structure and some key code to produce successful software.

Unfortunately development of DSP code for an embedded vision processor is more difficult than on the host computer. The DSP code development is at the assembler coding level. Although there is a higher level language, such as C, on the market for the DSP, the fact is that the execution speed of the C code, which is important to the image processing task, is slower than that of the assembler code. This problem will get worse when multiple DSPs are working together. The DSP board with four Motorola DSP56001s, which has been used for the single tracking, is a simple DSP array without a well developed operating system to control the whole DSP array. There is only a simple monitor for each DSP. There is no evidence to suggest that the manufacturer of the DSP board will develop a real time operating system or a parallel higher level language for this board in the near future. Development of this kind of operating system or parallel higher level language is neither a simple nor a fast software task but it would certainly help the development of DSP code. Any change of the DSP code in assembler to implement multiple tracking is not a simple task. The configuration of Figure 8.3a, in which there are two independent single track vision systems, needs only a modification to the code for the host microprocessor for the implementation of multiple tracking and does not involve any DSP code development. Therefore from the software development point of view this is the easiest configuration, but the hardware cost is higher.

The summary of comparison among three methods is presented on table 8.1.

	Figure 8.3a	Figure 8.3b	Figure 8.3c
Hardware cost	highest	middle	lowest
Viewing range	large	large	small
Software on DSP	easy	difficult	difficult

Table 8.1 Comparison of three configurations for tracking twin paths

The configuration on the Figure 8.3c is chosen to implement the multiple tracking because the distances between two adjacent cutting paths of most lace samples are inside the single camera view range. This will prove the concept of multiple tracking by adding an extra DSP on the existing vision system and re-writing some parts of the code without any hardware change.

8.2.2 Multiple Tracking by Using Multiple DSPs

The vision system is a multiple micro-processor environment in which three DSPs and one host processor work together. The main data streams are image data and result data (see Figure 8.4). There are two things to be considered when the system is going to be built. One is the data stream and another is the software organisation.

There are two bottlenecks in the vision system. One is the host processor and another is the global memory on the DSP board, which can be accessed by all microprocessors. The tasks of the host microprocessor are as follows:

1. Provide an interface for a human operator;
2. Control three DSP processors;
3. Display the cutting path in real time;
4. Drive two galvanometers in real time;
5. Display images for system diagnosis.

A small amount of data is involved in the tasks 3 and 4; other tasks are not time critical.

Compared to the host microprocessor the global memory on the DSP board involves a large amount of data. The memory has one data bus and one address bus therefore only one microprocessor can access the memory at any one moment. Ideally the memories and buses should be arranged according to data streams and microprocessors, such as local bus and dual port memory between microprocessors (see Figure 8.5) to allow more microprocessors to access the same data or different data at one time. It is difficult to find this kind of product on the market. If this kind of board is built here, it will cause higher costs and a longer development cycle. As a realistic alternative to prove the concept of twin path cutting a master slave arrangement for microprocessors and an event driven mechanism are used based on the arrangement shown in Figure 8.4 to organise the data streams and actions between the microprocessors.

Comparing Figure 8.3a, 8.3b and 8.3c, the arrangement of Figure 8.3a has the slight advantage of less memory conflict than the other two due to two separated memories on the DSP boards so Figure 8.3a has a chance to work faster in multiple path cutting than the other two, but will be limited to the same maximum speed as the vision system for single path cutting.

The host microprocessor acts as a master that controls the actions of all DSP processors. Three DSPs act as slaves that are switched on and off by the master processor.

The tasks of the vision system can be classified into setting up camera threshold level; taking the camera data into the memory; searching for a start point for tracking; tracking the cutting path; driving the galvanometers; interfacing to an operator; initialising the DSP. All these tasks are done on different microprocessors and operated in a sequence (see Table 8.2). The operation sequence of the tasks starts from the far left column and stops at the far right column. The tasks that fall into one column can operate in parallel. So every column can be seen as an event. The status board on the Figure 8.4 is used to display the current event. All microprocessors are looking at the status "board" to decide what task can be operated according to the sequence.

8.3 TRACKING RESULT TRANSFORM

The vision and cutting system (in rig 3 or prototype) works as follows.

The vision zone is upstream of the cutting zone by typically 20mm. The linescan camera system acquires data for each strip of the pattern. As each strip of pattern moves towards the cutting zone, the DSP extracts the cutting information from the incoming image by using the tracking program. As the lace passes the cutting zone, the CO_2 laser cuts the lace at the required point.

There are three different coordinate systems involved in the whole process:

1. System coordinate for the test rig, *Xs*, *Ys*. The camera, galvanometer are fixed in this system and lace is fed into this system (Figure 8.6).
2. Image coordinate (see section 4.2), *Xi*, *Yi*. In this coordinate image of lace is described by pixel and line.

3. Cutting coordinate, *Xc*, *Yc* in which the cutting positions are described by units (for D/A converter) and line.

initiator	task	time 1	time 2	time 3	time 4	time 5	time 6
host	human interface	---------	---------	---------	---------	---------	---------
host	initialise DSP	---------					
host	driving G					---------	
host	terminat ion						---------
DSP1	setting camera		---------				
DSP1	get camera data		---------	---------	---------	---------	
DSP2	start point				---------		
DSP2	tracking					---------	
DSP3	start point				---------		
DSP3	tracking					---------	
operator	move lace			---------		---------	
operator	cut lace by using laser					---------	

Note: '--------' means processor in action. If there are more than one '----' in one column, that means more than one processor in action at the same time.

Table 8.2 Timing of software function blocks for two beam cutting

The geometry relationships between the three coordinate systems are fixed after the camera and galvanometers are fixed on the rig. The most important relationship is between the image coordinate and cutting coordinate systems. This is necessary in order to cut lace at the correct position according to the data from DSP. There is no direct theoretical geometric relationship between these two systems. Therefore experimental calibration is necessary. The following, details the calibration method. The system coordinate is used as a reference in this procedure.

First find the geometry relationship between system coordinate and image coordinate and relationship between system coordinate and cutting coordinate because camera and galvanometer are fixed in the system coordinate. Then find the geometry relationship between image coordinate and cutting coordinate.

In general, two right hand coordinate systems can be described by each other if they have a fixed geometric relationship. Let us consider a two dimensional system only (see Figure 8.7). For example, a point P in space can be described in an x, y coordinate as $P(x, y)$ and in another x', y' coordinate system as $P(x', y')$. If the difference between origins of two coordinate systems, Xo, and Yo, scale Sx, and Sy, and the angle of two systems, α are known, any point in $x\ y$ coordinate can be described by x', y' coordinate using equation 8.2.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \times \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \times \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} Xo \\ Yo \end{bmatrix} \qquad \text{eq. 8.2}$$

In our case, α=0. If the origin of the system coordinate is set at the origin of the image coordinate, the eq. 8.2 can be simplified as equation 8.3. Xs, Ys are for system coordinate and Xi, Yi for the image coordinate. S_{Xsi} and S_{Ysi} are scale factors between the system coordinate and the image coordinate.

$$\begin{bmatrix} Xs \\ Ys \end{bmatrix} = \begin{bmatrix} Sx_{si} \times Xi \\ Sy_{si} \times Yi \end{bmatrix} \qquad \text{eq. 8.3}$$

Similarly, if the cutting coordinate is set parallel to the system coordinate, the relationship between the system coordinate and the cutting coordinate is equation 8.4. The subscript c represents the cutting coordinate.

$$\begin{bmatrix} Xs \\ Ys \end{bmatrix} = \begin{bmatrix} Sx_{sc} \times Xc \\ Sy_{sc} \times Yc \end{bmatrix} + \begin{bmatrix} Xo_{sc} \\ Yo_{sc} \end{bmatrix} \qquad \text{eq. 8.4}$$

Then a geometrical relationship between the image coordinate and the cutting coordinate, equation 8.5, can be derived from equation 8.3 and 8.4.

$$\begin{bmatrix} Xc \\ Yc \end{bmatrix} = \begin{bmatrix} \frac{Sx_{si}}{Sx_{sc}} \times Xi \\ \frac{Sy_{si}}{Sy_{sc}} \times Yi \end{bmatrix} + \begin{bmatrix} -\frac{Xo_{sc}}{Sx_{sc}} \\ -\frac{Yo_{sc}}{Sy_{sc}} \end{bmatrix} \qquad \text{eq. 8.5a}$$

Equation 8.5 can be simplified into equation 8.5b.

$$\begin{bmatrix} Xc \\ Yc \end{bmatrix} = \begin{bmatrix} Ax \times Xi \\ Ay \times Yi \end{bmatrix} + \begin{bmatrix} Bx \\ By \end{bmatrix} \qquad \text{eq. 8.5b}$$

If Xc and Xi use the same scale derived from the encoder (see chapter 6), Ax equals one and Bx is the number of lines equivalent to the distance between scan zone and cutting zone (relevant discussion of Bx, queue length, in section 7.2). The Ay and By can be determined by test. The test procedure is as follows.

A template, printed on a transparency film using a laser printer, and consisting of a series of parallel bars, is placed on the lace registration surface in the camera field of view (scan zone) and over the laser beam cutting axis (cutting zone, see Figure 8.8). The width of all bars is 5mm and the space between any two consecutive bars is 5mm. The bar chart is fixed on the system coordinate, perpendicular to the scan zone and cutting zone. The camera acquires image of the bar chart, and a He-Ne laser points to several different positions on the bar chart by feeding different data to the D/A converter, such as 0, 2048, 4095. Corresponding points are then found, Yi in image coordinate with Yc. If two pairs of (Yc_1, Yi_1) and (Yc_2, Yi_2) are found, the Ay and By can be decided by using equation 8.6.

$$\begin{bmatrix} Ay \\ By \end{bmatrix} = \begin{bmatrix} \frac{Yc_2 - Yc_1}{Yi_2 - Yi_1} \\ Yc_1 - Yi_1 \times \frac{Yc_2 - Yc_2}{Yi_2 - Yi_1} \end{bmatrix} \qquad \text{eq. 8.6}$$

The following is an example:

image of bar (in hexadecimal format, 720 pixel width)

```
ffffff ffffff ffffff fffff0 0001ff ff7000 01ffff
c00007 ffffc0 0007ff ff0000 0fffff 00001f fffe00
001fff fe0000 3ffffc 00007f fffc00 007fff fc0000
7ffffc 00007f fff800 007fff fc0000 7ffff8 00007f
ff7800 007fff
```

Every hexadecimal number in the image represents four binary pixels. For example '0001ff' means that there are 15 dark pixels on the left and there are 9 bright pixels on the right. A bar in the image is defined by two transition positions. One position is from bright to dark ('begin'), and another is from dark to bright ('end'). All bars in the image can be determined by searching the 'begin' and 'end' positions.

coordinate of bar in y direction ('begin' and 'end' in pixels)

1) 92 110	2)132 150	3)170 188	4)209 228	5)248 267	6)288 306
7)327 346	8)367 385	9)406 424	10)446 464	11)486 506	12)526 544
13)565 584	14)606 624	15)646 664	16)685 704		

After this the bar chart is registered in the camera co-ordinate system. Then the He-Ne points to two desired positions on the bar chart by sending data to D/A converter. If the pixel numbers of the two desired positions and the data sent to D/A converter are put into equation 8.6, A_y and B_y can be determined.

data pairs (D/A and Image)

$(Y_{c1}\ \ Y_{i1})$ $(Y_{c2}\ \ Y_{i2})$

(1311 445) (2845 267)

Ay=-8.61797 By=5146

The relation between the image coordinate and the cutting coordinate is as follows:

$Yc = 5146 - 8.61797 \times Yi$

8.4 CUTTING TESTS, TRACKING ERROR AND CUTTING ERROR

Vision system 4 (Figure 3.13) combines vision system 3 (Figure 3.12) with a twin beam vision system (Figure 8.3c). This system can switch between twin path tracking and high speed single path tracking by using different software. In the test programme this system was used to guide two laser beams for dual edge cutting at 250mm/s, and also one laser beam for single edge cutting in excess of 1m/s.

Two observations have been made during the test programme. One was about the host computer. According to chapter 7, the predicted speed for correlator implementation should be over 1m/s. The test indicated that lace could be cut successfully at speeds below 850mm/s. As soon as the lace feeding speed exceeded this value, the cutting path started to lose synchronisation with the lace pattern. The cause of this was that the display of cutting path on the graphic screen in real time (see section 8.2.3) took too long. After this function was disabled, the cutting speed functioned well at 1070mm/s.

The second observation was about the cutting error. There are many factors that cause the cutting error in the test. Tracking error is one of them. Other factors will be analysed in the next section.

Several methods to evaluate the tracking error are discussed as follows:

1. Since the tracking error is defined as the difference between *Pim(x)* and *Pid(x)* in image coordinates, a natural way to assess the tracking error is to compare *Pim(x)* with *Pid(x)*. The advantage of this method is that other errors, such as ripple on the lace affecting the cutting process, can be ignored. A difficulty of this method is to define *Pid(x)* in the image.

2. Two dimensional cross-correlation is one possible method to locate the best matched part on image against the map. After that the reference path *Pma(x)* can be located on the image as *Pid(x)*. A major problem is that the method does not take account of any variation of lace and the *Pid(x)* defined by 2-D, so best match produced by cross-correlation might be

worse than *Pim(x)* that is produced by tracking because *Pim(x)* is designed to take account of the variation.

3. To define *Pid(x)* on incoming image manually, this is similar to the definition of *Pma(x)*, is another possible way. The disadvantage of this method is that some of differences between *Pim(x)* and *Pid(x)* may be caused by the subjective judgement of the operator. In other words different operator may get different *Pid(x)* from the same image.

In order to avoid directly defining *Pid(x)* on incoming image, one method is developed to assess the difference, tracking error, between *Pma(x)* with *Pim(x)* as follows.

For a given curve, *G(x)*, and a given Feature, *F(x)*, there is difference, *D(x)*, between *G(x)* and *F(x)* in *y* direction (see Figure 8.9). Average of *D(x)*, $\overline{D}$ (see equation 8.7), represents the statistical geometric relationship between *G(x)* and *F(x)*. If equation 8.7 is applied to *Pma(x)*, *Pid(x)* and *Pim(x)*, we can get *Amap*, *Aid*, and *Aim* (see equation 8.8). In theory *Amap* equals *Aid* so that statistically the difference between *Amap* and *Aim* is the average tracking error, σ_A, (see equation 8.9).

$$\overline{D} = \frac{1}{n}\sum_{i=1}^{n} D(i) = \frac{1}{n}\sum_{i=1}^{n}(G(i) - F(i)) \qquad \text{eq. 8.7}$$

$$Amap = \overline{Dmap} = \frac{1}{n}\sum_{i=1}^{n}(Pma(i) - Fma(i))$$

$$Aid = \overline{Did} = \frac{1}{n}\sum_{i=1}^{n}(Pid(i) - Fim(i)) \qquad \text{eq. 8.8}$$

$$Aim = \overline{Dim} = \frac{1}{n}\sum_{i=1}^{n}(Pim(i) - Fim(i))$$

$$\sigma_A = Aid - Aim = Amap - Aim \qquad \text{eq. 8.9}$$

But the definition of *D(x)* is a one dimensional error which is different from an ideal tracking error definition similar to Figure 8.2 in two dimensional form. When the rate of change of *G(x)* is small, the two definitions exhibit little difference; when the rate of change of *G(x)* is higher, the *D(x)* is bigger than the ideal tracking error.

The average tracking error of real lace on prototype and vision system 4 determined by equation 8.9 is less than 1 pixel (see Appendix 8).

Another method to assess the tracking error is to assess the cutting error. This method can only estimate the tracking error but further analysis of cutting error can help in the real machine design. The *Pid(x)* in this case should be defined according to real lace. For example if scalloping, *Pid(x)* is defined on the scalloping *edge* of lace.

Figure 8.10, and 8.11 show two cutting results of one lace design. The average cutting error is less than 0.5mm. The biggest error is 1mm from the edge. So the biggest tracking error will be to equal or less than 1mm or 4 pixels (if lateral resolution is 0.25mm per pixel). The next section examines possible factors that can produce cutting errors.

8.5 EXAMINATION OF CUTTING ERROR

A list of possible factors, which affect the cutting error, in test rig 3 and prototype are as follows.

1. Lateral distortion of the lace.
2. Variation of illumination.
3. Longitudinal variation of the lace.
4. Encoder error.
5. Variation of laser spot.
6. Driving method of galvanometer to produce step effect.
7. Dynamic response, non-linearity of galvanometer.
8. Parallelism between camera and galvanometer.
9. Surface ripple of the lace.
10. Lateral movement of the lace.
11. Thermal drift of the test rig frame.
12. Thermal drift of the galvanometer.

Factors 11 and 12 affect the cutting error in a longer time scale. Other factors cause cutting errors in a shorter time scale.

8.5.1 Lateral Distortion of the Lace

The first error has been examined in section 4.4. The maximum tracking error for all the tests at 10% stretch was 3 pixels or less, with or without simulated noise, when β values of 0.8-0.9 were used. When the variation is less then 5%, the maximum tracking error is less than 2 pixels (0.5mm).

8.5.2 Variation of Illumination

Two extreme cases of the second error are that, the incoming image is all '1' or all '0' (if illumination is too bright or too dark), in which case *Pim(x)* can be anywhere due to lack of landmarks to find reference to the current position (see section 3.2.3). The same conclusion can be drawn by putting all '1' or all '0' for *Ig(i,y-q)* into equation 4.7. The image in between these two extremes contains the lace feature. When the image goes from bright to dark for whatever reason, the width (see section 6.3) of the feature becomes wider. This also affects the similarity between the image and the map. The tracking error due to this factor is similar to the error caused by factor one.

8.5.3 Longitudinal Variation of the Lace

The analysis of the third error is as follows. Assuming the lateral distance in the map between *j*th and *(j+1)*th reference path is *d* and on the real lace the *i*th image is in the middle between *j*th and *(j+1)*th map, the ideal cutting path, *Pid(i)*, on the image is *Pma(j)+d/2+offset* (see Figure 8.12). But according to section 4.3, the *Pim(i)* will be *Pma(j)+offset* or *Pma(j+1)+offset* depending on best match or 'wait', 'step' actions. So the tracking error will be *d/2*. If *d*=0, the tracking error due to longitudinal variation is zero. As *d* increases, the tracking error will get larger. If *d*=1mm, the tracking error will be 0.5mm.

8.5.4 Encoder Error

The encoder error will vary the distance between two consecutive lines of the image (as the third error), which affects tracking accuracy and which will affect *Xco* in equation 8.4. A test of encoder error was carried out on test rig 2 (see Figure 3.15) in which an extra encoder was used to record the position of the table. The largest single pulse error of the encoder used to monitor the position of the lace is 2 pulses in 0.25mm lace movement (average pulses for 0.25mm is 13 pulses); and total error in 83 mm is 4 pulses (about 0.077mm). The test indicated that most encoder error (97%) is within ±1.335 pulses for every 13 pulses (see Appendix 9). This error has little effect on the cutting error.

8.5.5 Variation of Laser Spot

The next factor is the shape of the laser spot. If the shape of the spot is a circle when laser beam is perpendicular to the cutting surface, the spot changes from the circle to an ellipse as galvanometer changes its axis angle and the beam is no longer perpendicular to the cutting surface. Another factor is the spot size. As the laser output power increases and the lace speed does not change, the effective spot size increases, which will affect the cutting. Theoretically the spot shape of the laser under perfect condition should be circle. But in real world the shape is not a circle, the shape will vary the cutting edge. In one test, when D/A input changes from 2048 (perpendicular to the cutting surface) to 4095 (upper range of D/A, 70mm apart from the first position) the width of spot changes from 1mm to 1.2mm. The effect of the spot shape on the cutting accuracy is shown in Figure 8.13. It is analogues to the radius of a machine tool in mechanical cutting.

8.5.6 Driving Method of Galvanometer to Produce Step Effect

The driving method used for the galvanometer is another factor in the cutting error assessment. The program only changes the D/A value when new cutting data is extracted (see chapter 7). This method produces a real cutting path command with step of changes of D/A value with perfect positional accuracy. Figure 8.14 is an example. The circles represent positions for the laser spot, the centre of laser spot moves as a series steps which will affect the cutting accuracy. Every D/A value change will cause high speed lateral movement of laser at that point which will reduce the cutting effectiveness. These steps can be eliminated by using a position control method to link any two consecutive cutting points. This continuous control can also improve cutting quality.

8.5.7 Dynamic Response, Non-linearity of Galvanometer

The dynamic response and non-linearity of a galvanometer are other factors which can cause the cutting error. The dynamic response concerns the execution delay and the difference between the required angle and actual angle of the galvanometer shaft when the frequency of the input command varies (Inman 1994). Figure 8.15 shows a result of a test. The galvanometer with a mirror was driven by a sinusoidal signal generator. A He-Ne beam was reflected to a scaled plate by the mirror. During the test the frequency of the sinusoid wave was varied, while the amplitude of it was kept constant. The test shows that the width (or the amplitude of angle of the galvanometer) of the scanned He-Ne beam on the scaled plate is varied with the input frequency. When the frequency changes from 10Hz to 50Hz, the width changes from 82mm to 90mm, which is 9.7% larger. For example, a cutting command is defined according to a map, which produces 10Hz frequency when lace is fed at 100mm/s or 50Hz when lace is fed at 500mm/s. If this command is used to cut a lace well when lace feeding speed is 100mm/s, it will cause 9.7% cutting error when lace is fed at 500mm/s due to the dynamic response of the galvanometer.

The error of non-linearity is defined by equation 8.10.

$$\sigma = h \times (\tan\alpha - 0.01745\alpha) \qquad \text{eq. 8.10}$$

α	5°	10°	15°	20°
σ	0.00022h	0.00179h	0.006149h	0.0149h
σ(h=250mm)	0.055mm	0.4485mm	1.537mm	3.725mm

α--- mirror angle in degree; h--- distance from mirror to cutting plate.

Table 8.3 Non-linear error due to angle of galvanometer mirror

If h=250mm, the table 8.3 shows the error with the angle.

The setting up error of the galvanometer, which causes additional angular movement of the laser beam, is similar to non-linearity discussed above.

8.5.8 Parallelism between Camera and Galvanometer

Let us consider the error caused by the case where camera plane is not parallel to the galvanometer plane. Ideally the camera scan zone and galvanometer cutting zone are parallel to each other and also perpendicular to the lace movement direction. This parallelism problem occurs when assembling the test rig. The parallelism can be measured by α in Figure 8.16. After assembly, the angle is fixed, Δx, Δy (the cutting error) caused by this angle is a function of l (in cutting the l is changed). Equation 8.11 shows the relationship between l and Δx, Δy.

Let us see the effect of Δx, Δy by showing an example.

Assuming α=1°, table 8.4 is result of Δx, Δy with l.

l(mm)	5	10	15	20	25	30	35	40	45	50
Δx(mm)	.087	.175	.262	.349	.436	.52	.611	.698	.785	.873
Δy(mm)	.0007	.0015	.002	.003	.0038	.0046	.0053	.006	.0068	.0076

Table 8.4 Cutting error due to setting of camera and galvanometers

Δx has greater effect on cutting accuracy than Δy in this case.

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} l \times sin\alpha \\ l \times (1 - cos\alpha) \end{bmatrix} \qquad \text{eq. 8.11}$$

8.5.9 Surface Ripple of the Lace

Ripples, or creasing, on the surface of the lace can cause cutting errors. Ideally the lace should lie flat on the cutting plane, even when being transported. But ripples, caused by various reasons, will change the height of viewing and cutting plane (see Figure 8.17). The difference between image and cutting, caused by ripple, is determined by equation 8.12.

$$\Delta y_1 = \Delta h \times \frac{l}{h_1}$$
$$\Delta y_2 = \Delta h \times \frac{l}{h_2} \qquad \text{eq. 8.12}$$
$$\Delta l = \Delta y_1 - \Delta y_2 = \Delta h \times l \times \frac{h_2 - h_1}{h_1 \times h_2}$$

where h_1 and h_2 are the distances between galvanometer and cutting plane and between camera and cutting plane respectively. Δh is the height of ripple and l is the distance between ripple and centre of camera or galvanometer (if two centres on the same y of the system coordinate). If h_2 = 580mm, h_1 = 200mm and l=50mm, $\Delta l = \Delta h \times 0.16$. If Δh=2mm, Δl=0.32mm. Some ripple is as high as 5mm which can produce 0.8mm cutting error.

8.5.10 Lateral Movement of the Lace

The lateral movement of lace, in y direction, is another factor of cutting error. The ideal lace movement is perpendicular to the scan and cutting zone. When the lace moves laterally, cutting error occurs (see Figure 8.18). The lateral movement Δv refers to cutting error. V is variation on a roller at one end of cutting plane. l is the length of cutting plane. l_1 is the distance between scan zone and cutting zone. Then Δv is decided by equation 8.13.

For example, l=500mm and l_1=30mm, a set of v and Δv are table 8.5.

v mm	1	4	1	14	1	5	6	12	12	9	11	12	7	8
Δvmm	.06	.24	.06	.84	.06	.3	.36	.72	.72	.54	.66	.72	.42	.48

Table 8.5 Cutting error caused by a lateral movement

$$\Delta v = v \times \frac{l_1}{l} \qquad \text{eq. 8.13}$$

8.5.11 Thermal Drift of Rig Frame

Thermal effect can be divided into frame drift of test rig and electronic drift of galvanometer.

A linearly variable differential transducer (LVDT) is used to monitor the variation of distance between top frame, which holds camera and galvanometer, and middle frame, which holds a cutting plate and a lamp (see Figure 8.19). The resolution of LVDT is 0.001mm. Table 8.6 is a test.

time	2:00	2:08	2:11	3:30	4:45	4:55	5:00	5:10	5:15	32:50
reading (mm)	0.349	0.40 7	0.490	0.852	0.84 6	0.856	0.51 5	0.428	0.390	0.314
T(point)	22°	22°	23°	31°	31°	31°	28°	27°	26°	19°
T(room)	22°	22°	22°	22°	22°	22°	22°	22°	22°	19°
lamp	on	on	on	on	on	off	off	off	off	off

T------- temperature in Celsius.

Table 8.6 Frame thermal drift

In the table 8.6 the biggest drift in *z* direction of system coordinate is 0.51mm. Because this drift makes top frame not parallel to the middle frame, this needs to be transferred to the y direction to the cutting error. There are three different cases in Figure 8.20. Figure 8.21 to 8.23 show each case.

Case 1:

$$\Delta l = \Delta h^2 / l \qquad \text{eq. 8.14}$$

If $\Delta h=0.5$ and $l=350$, $\Delta l=7.14\times10^{-4}$mm.

Case 2:

$$\Delta l = \Delta l_1 + \Delta l_2 \approx \Delta h^2 / l + \Delta h \times \frac{\cos\beta}{\sin(\beta-\alpha)} \qquad \text{eq. 8.15}$$

If $\Delta h=0.57$mm, $l=400$mm and $\beta=70°$ $\alpha=0.08°$, $\Delta l\approx0.208$mm.

Case 3:

$$\Delta l = \Delta l_2 - \Delta l_1 = \Delta h \times \frac{\cos\beta}{\sin(\beta+\alpha)} - \Delta h^2/l \qquad \text{eq. 8.16}$$

If Δh=0.42mm, l=300mm and β=70° α=0.08°, $\Delta l \approx$0.152mm

The cutting error in case 1 can be ignored due to small quantity. But in case 2 and case 3 the cutting error might be noticed.

8.5.12 Thermal Drift of Galvanometer

The second thermal drift is that of the galvanometer. Test in table 8.7 shows the drift in three hours.

D/A(in)	D/A(out)	Feedback	time	temperature
3072	4.992v	1.3011v	2:05	22°
3072	5.000v	1.315v	4:55	36°

Table 8.7 Galvanometer thermal drift

The feedback refers to the position of mirror which is mounted on the shaft of the galvanometer. The linear relation between feedback of galvanometer and movement in y direction of system coordinate is 0.043v/mm. The 0.014v drift represents 0.325mm drift in cutting error.

8.5.13 Summary of Cutting Errors

The Figure 8.24 shows an example of cutting error. The table 8.8 lists the worst case of cutting errors caused by the above factors based on quantitative analysis or test. Some errors related to the steep ratio between lateral and longitudinal resolutions, such as error 3 and 6, can be reduced by increasing the longitudinal resolution. Error 7 can be reduced by a well calibrated lookup table. Error 8 can be reduced by a carefully setup. Reduction of the distance between cutting and scan zone will decrease the cutting error caused by item 10. The ripple can be reduced by using an even tension material handing system to flatten the lace.

1	Lateral variation of lace	0.5mm (5% variation)
2	Variation of illumination	no significant error with 5% voltage variation
3	Longitudinal variation of lace	0.5mm (8/1 ratio)
4	Encoder error	less than 0.25mm (constant and low lace feeding speed)
5	Variation of laser spot	0.2mm
6	Driving method of galvanometer to produce step effect	1mm (8/1 ratio)
7	Dynamic response and non-linearity of galvanometer	0.5mm (α=10°)
8	Parallel between camera and galvanometer	0.5mm
9	Surface ripple of lace	0.8mm
10	Lateral movement of lace	0.84mm
11	Thermal drift of rig frame	0.2mm
12	Thermal drift of galvanometer	0.325mm

Table 8.8 Worst cases of cutting errors

8.6 CONCLUDING REMARKS

The hardware configuration of single camera plus multiple DSPs is sufficient to prove the concept of multiple tracking on the lace patterns with narrow width.

The software method of the master slave arrangement of the microprocessors and the event driven mechanism is one way to organise multiple microprocessors for the multiple tracking.

A reference bar chart can be used to calibrate vision system and transform the cutting data extracted by DSP to D/A data to drive the galvanometers.

Average D(x) method is one way to assess the tracking accuracy by comparing average Dim(x) and Dma(x). The average tracking error is less than 1 pixel. The cutting error by measuring the real cut lace is less than 1mm.

The ripple on the lace surface and lateral movement of the lace produce the largest portion of the cutting error.

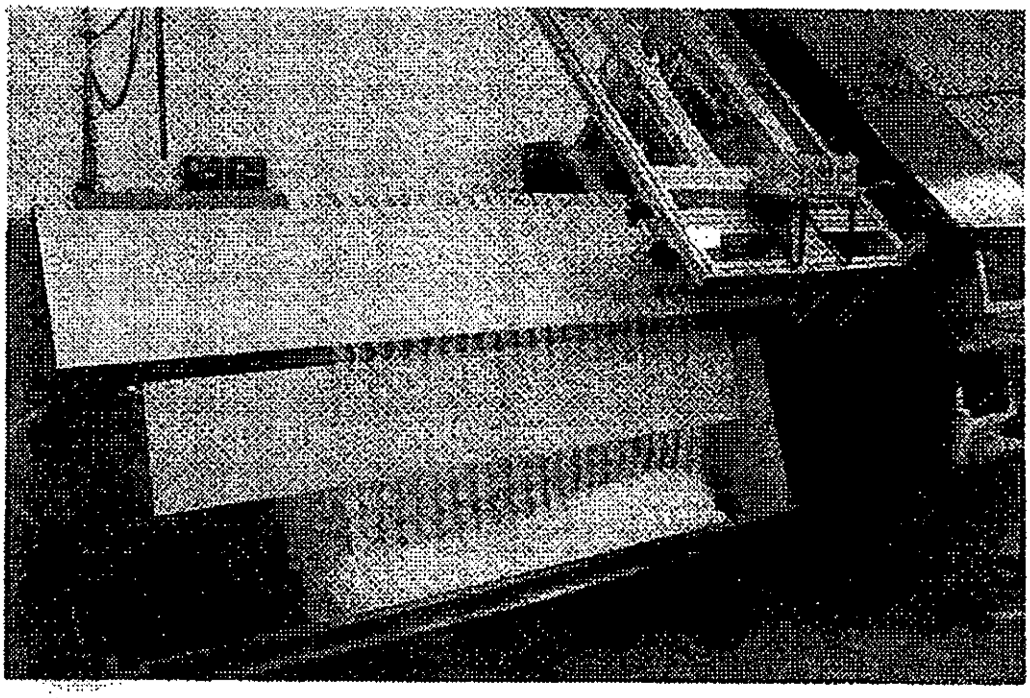


Figure 8.1 Front view of the prototype

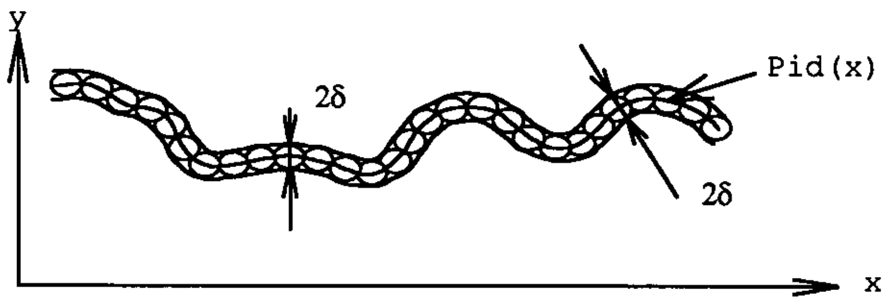


Figure 8.2 Cutting error definition

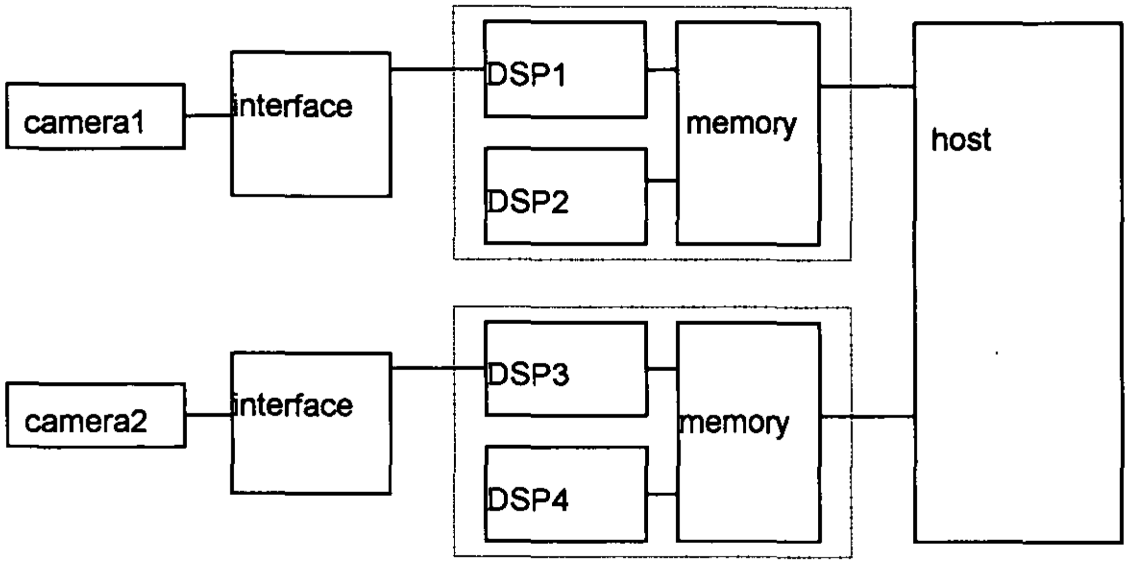


Figure 8.3a Two sets of camera and DSP

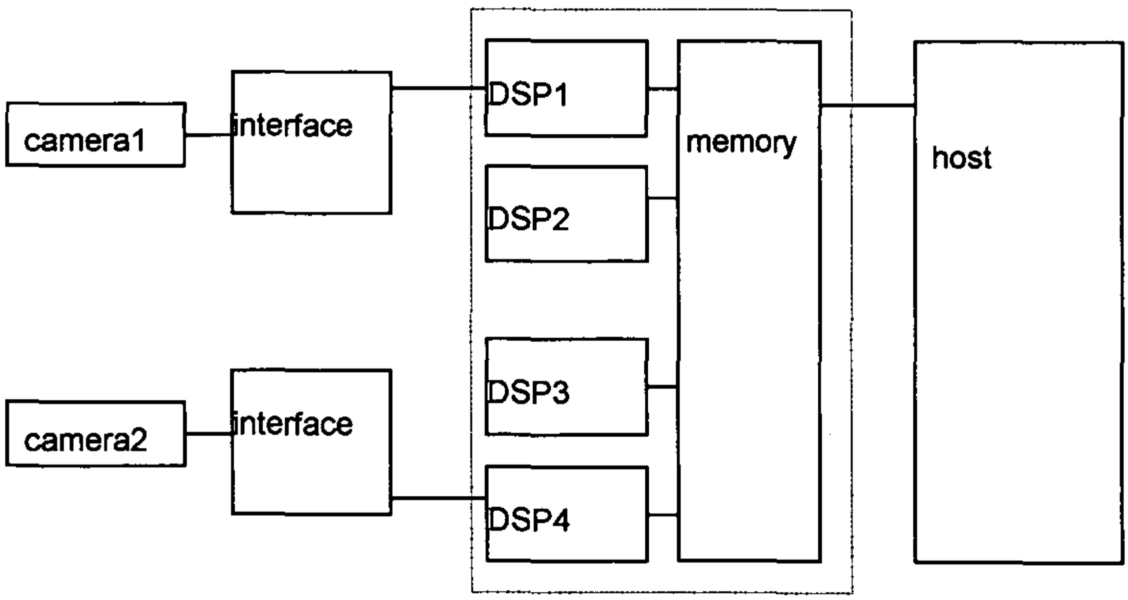


Figure 8.3b Two cameras and one set of DSP

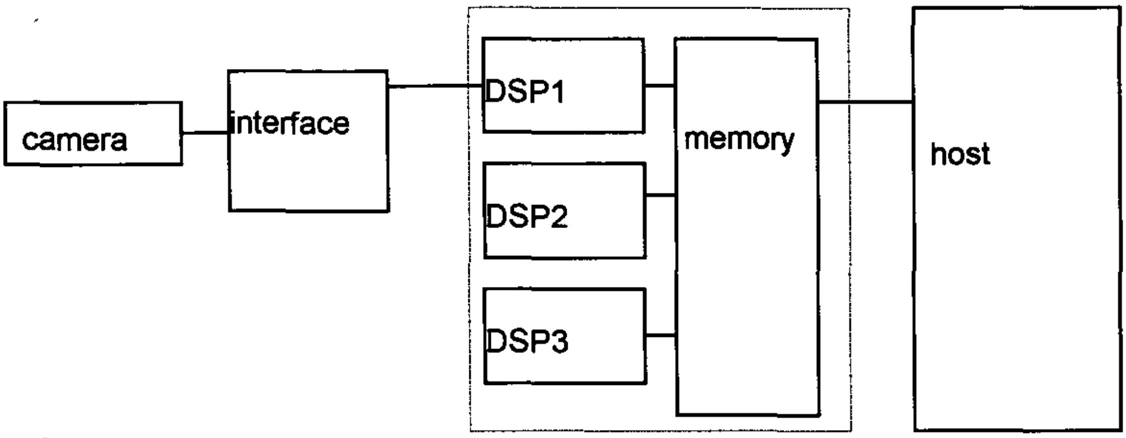
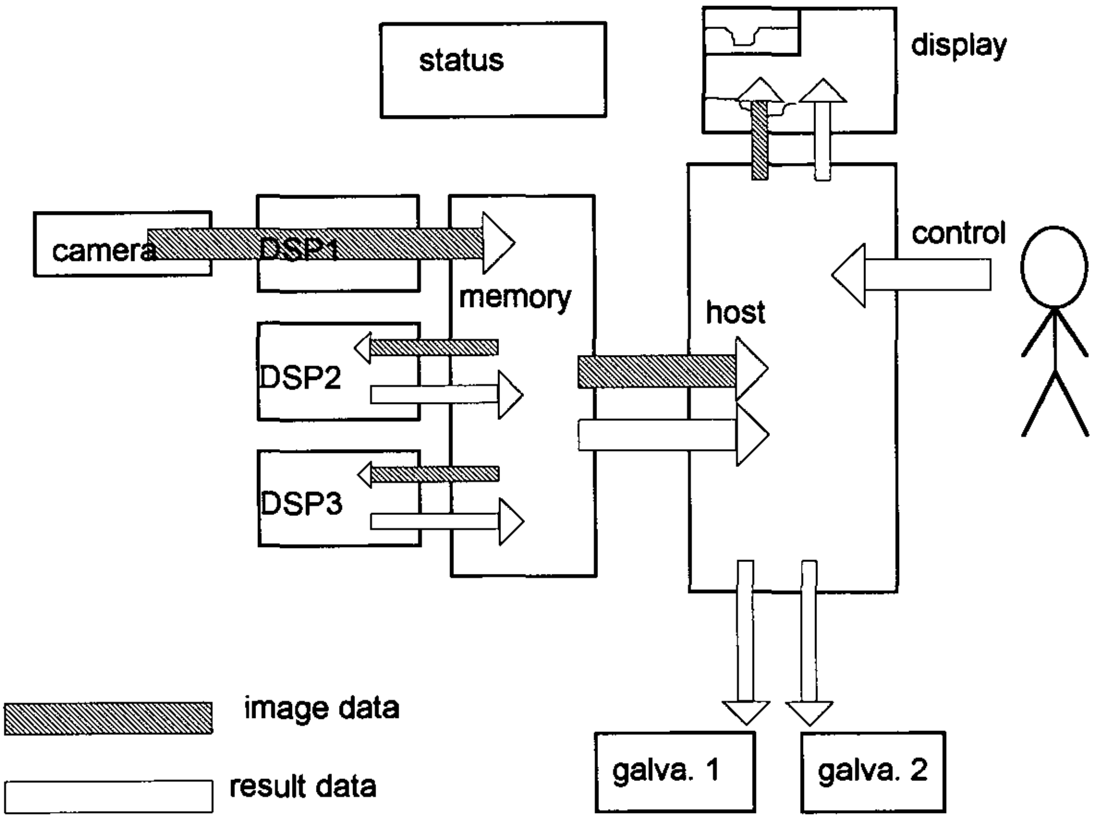


Figure 8.3c One set of camera and DSP



Note: galva. stans for galvanometer.

Figure 8.4 Data streams

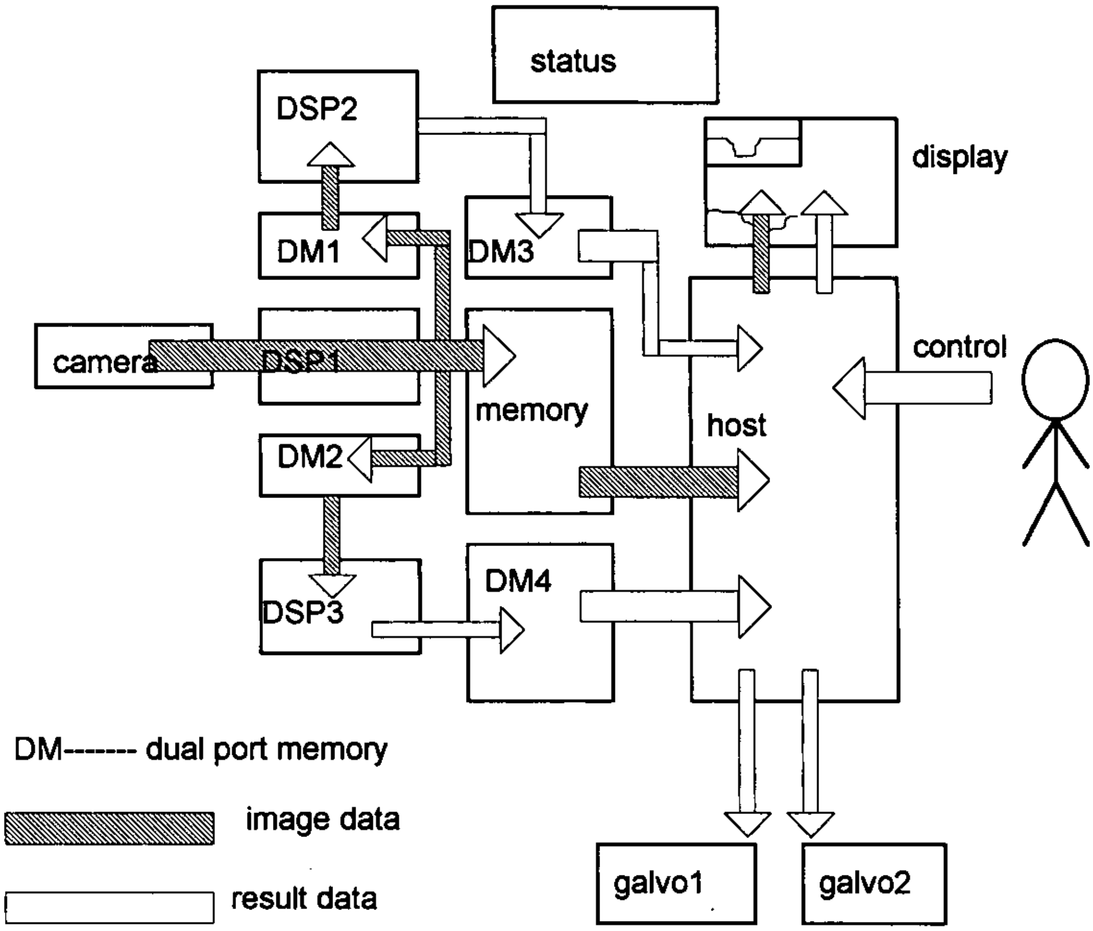


Figure 8.5 Ideal bus and memory for vision system

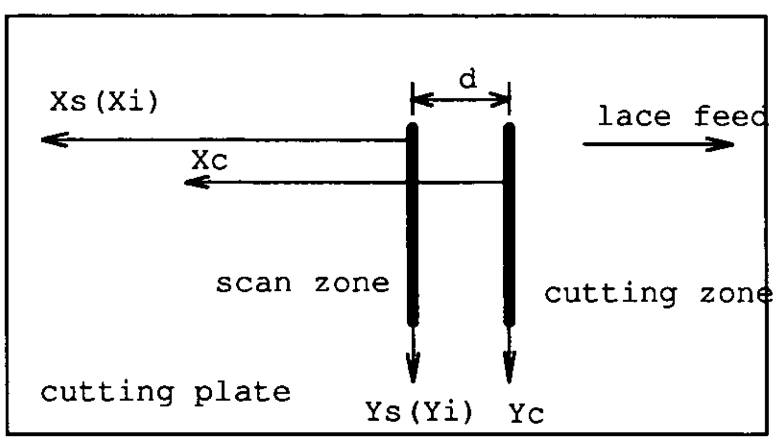


Figure 8.6 Three coordinate systems

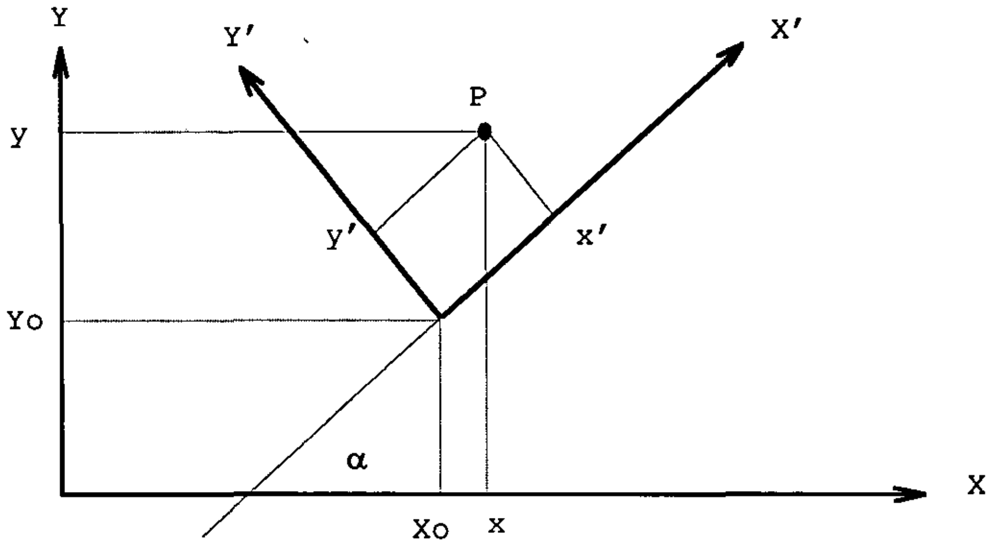


Figure 8.7 Coordinate transform

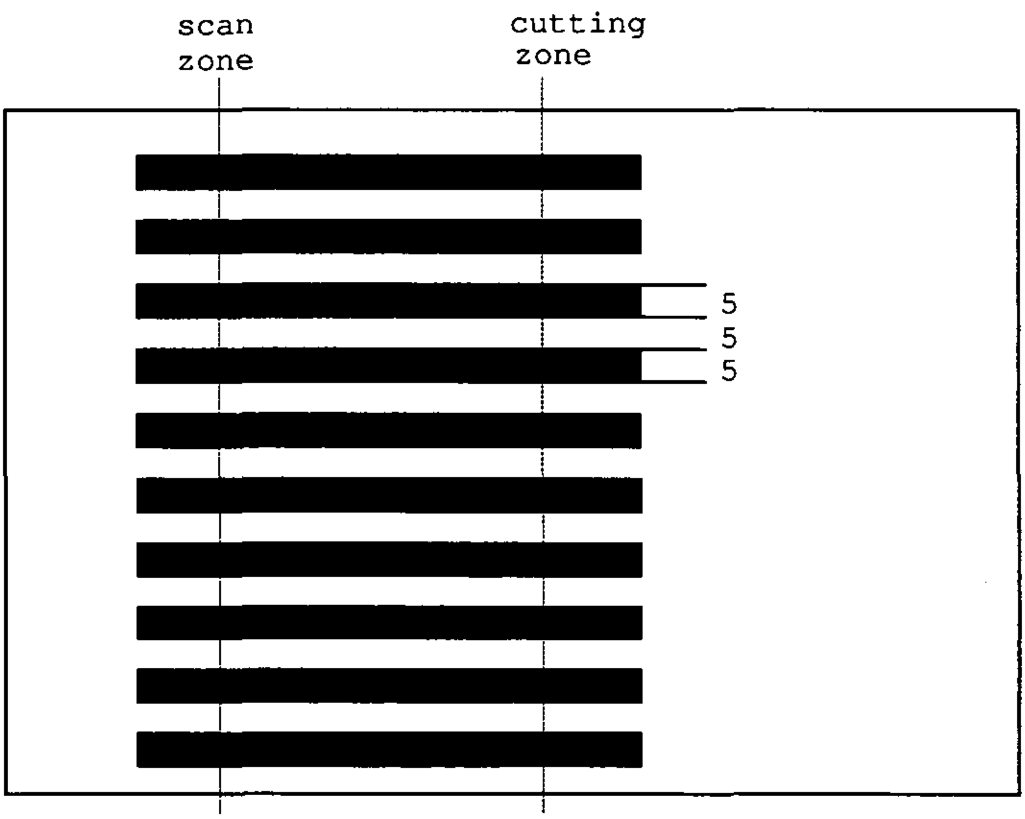


Figure 8.8 Calibration mask

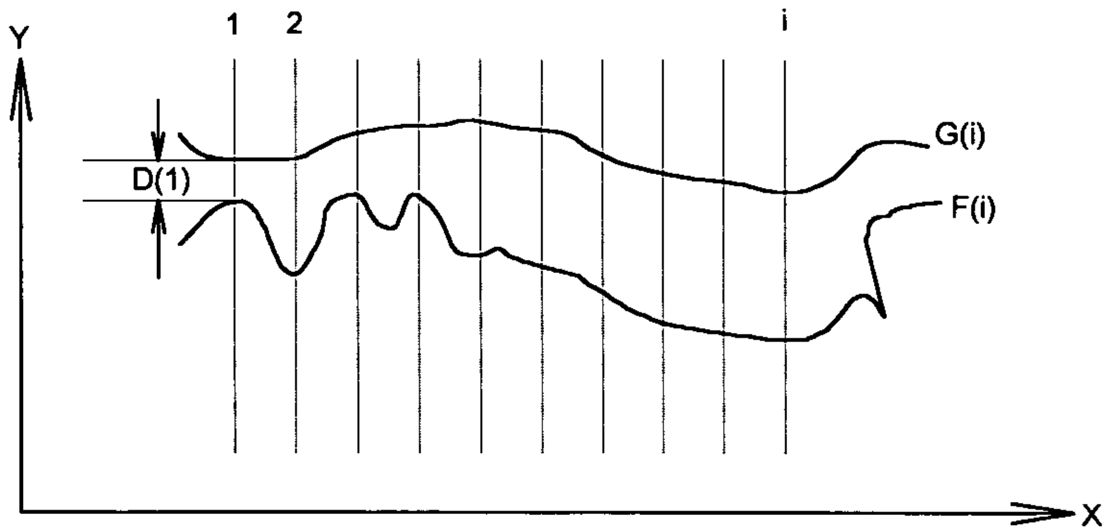


Figure 8.9 *D(x)* definition

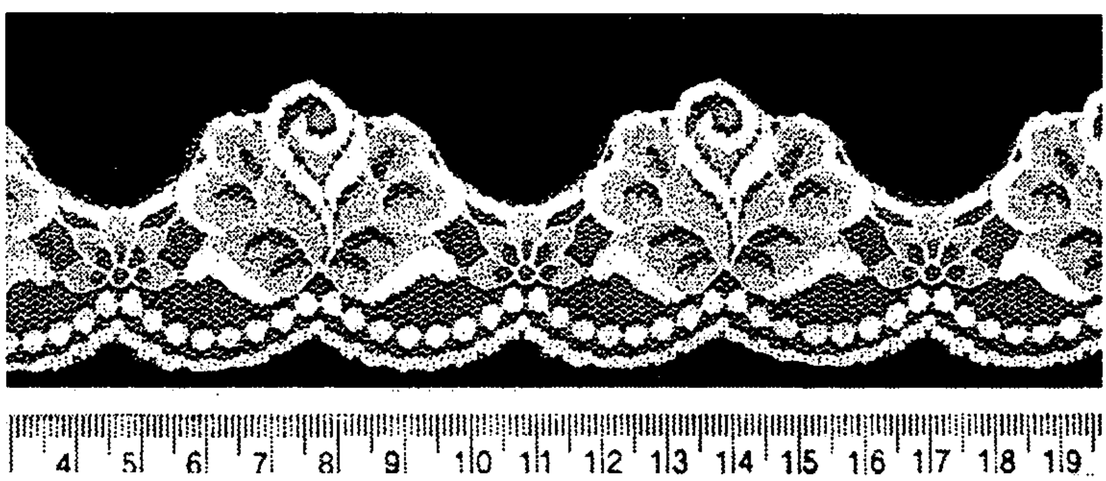


Figure 8.10 Twin beam cut result at 250mm/s

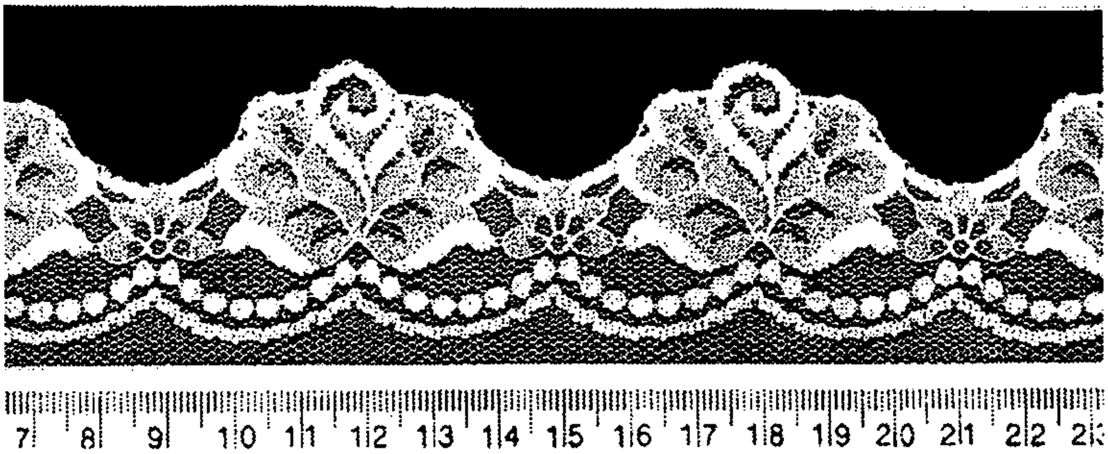


Figure 8.11 Single beam cut result at 1070mm/s

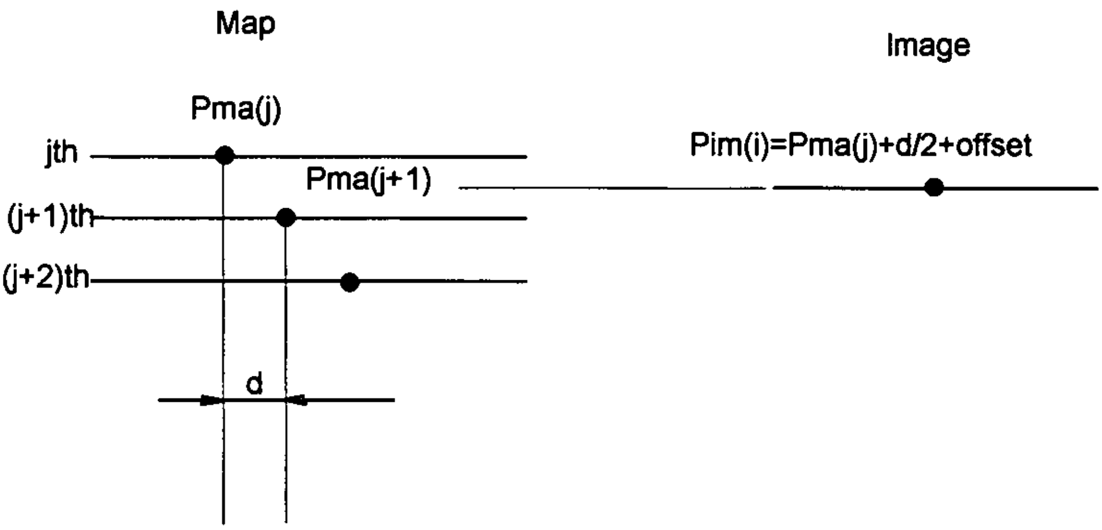


Figure 8.12 Ideal cutting position

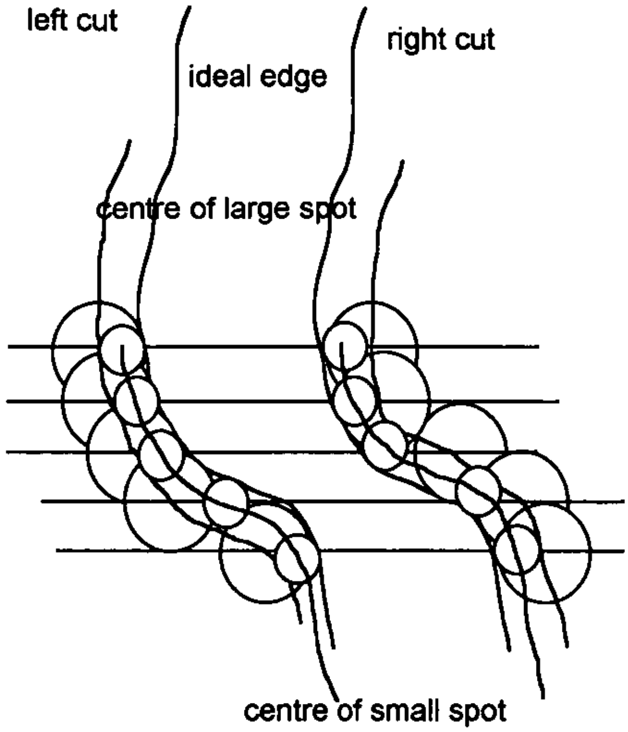


Figure 8.13 Different side of edge using different offset

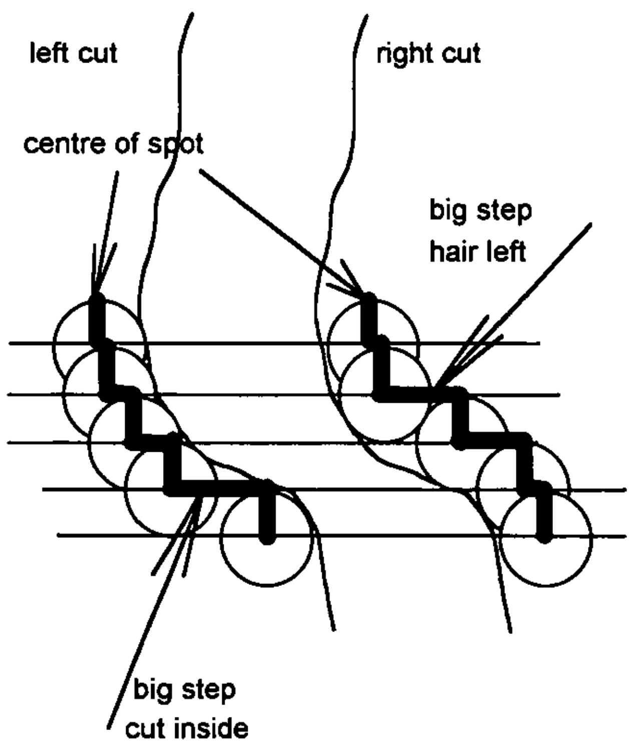


Figure 8.14 Step effect

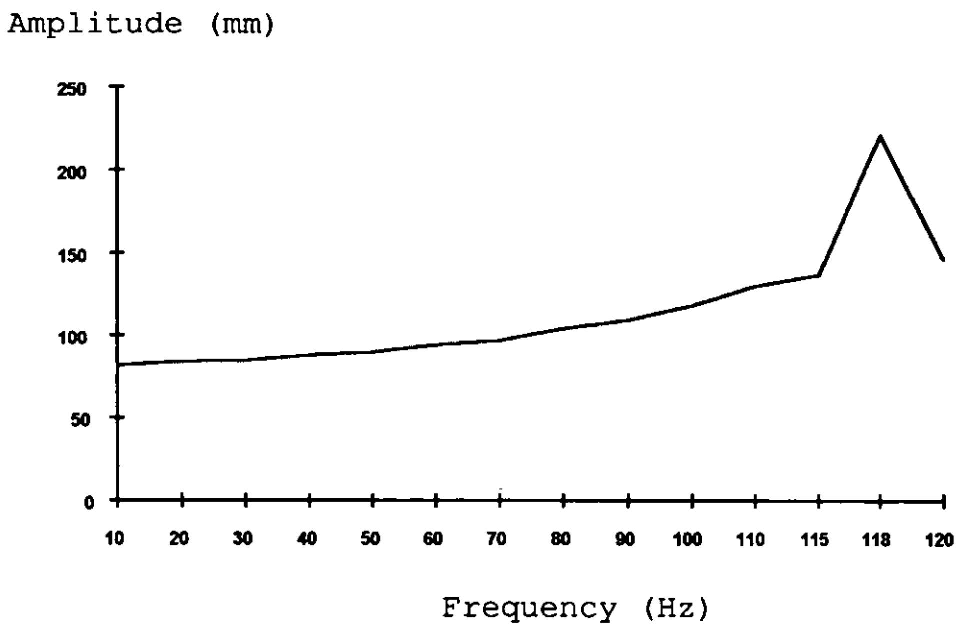


Figure 8.15 Frequency versus amplitude

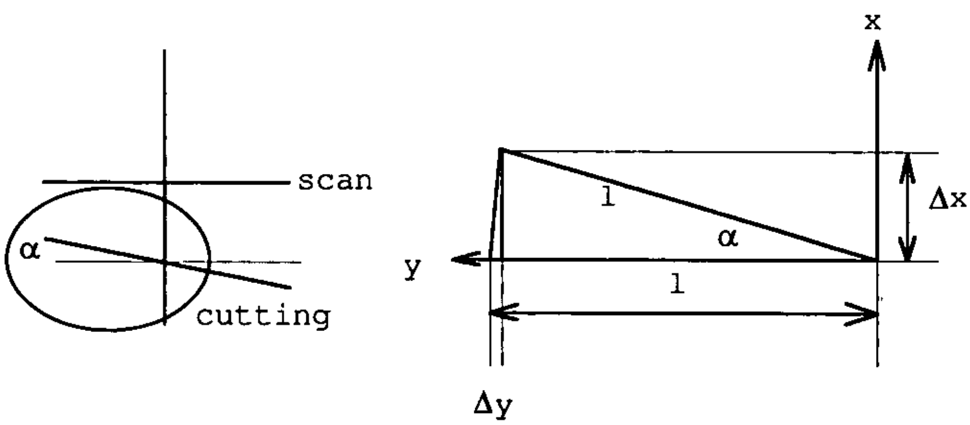


Figure 8.16 Non parallel between scan zone and cutting zone

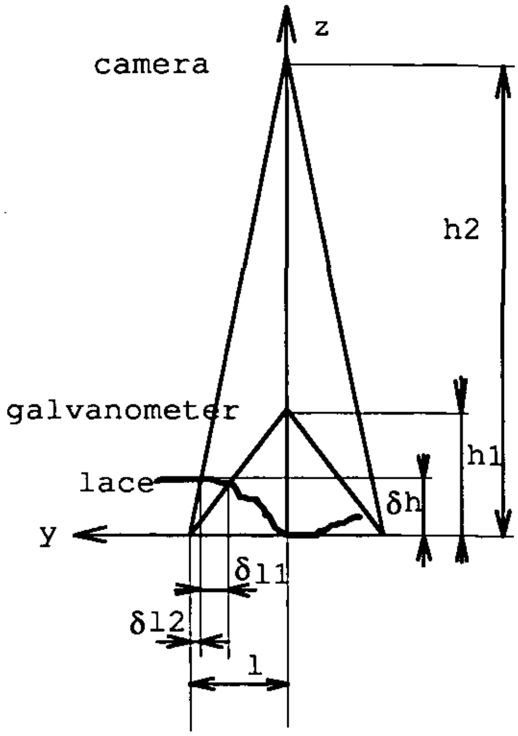


Figure 8.17 Ripple effect

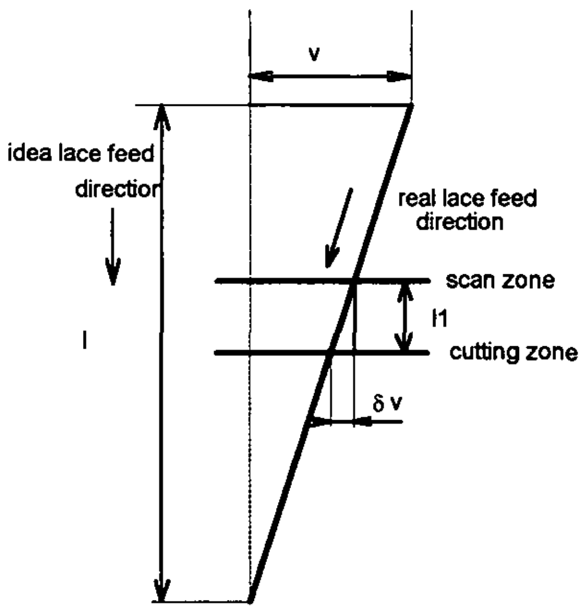


Figure 8.18 Lateral drift

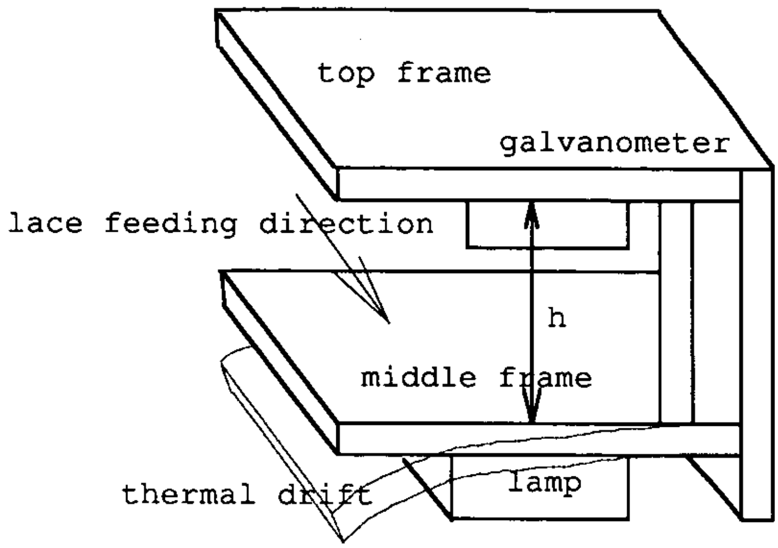
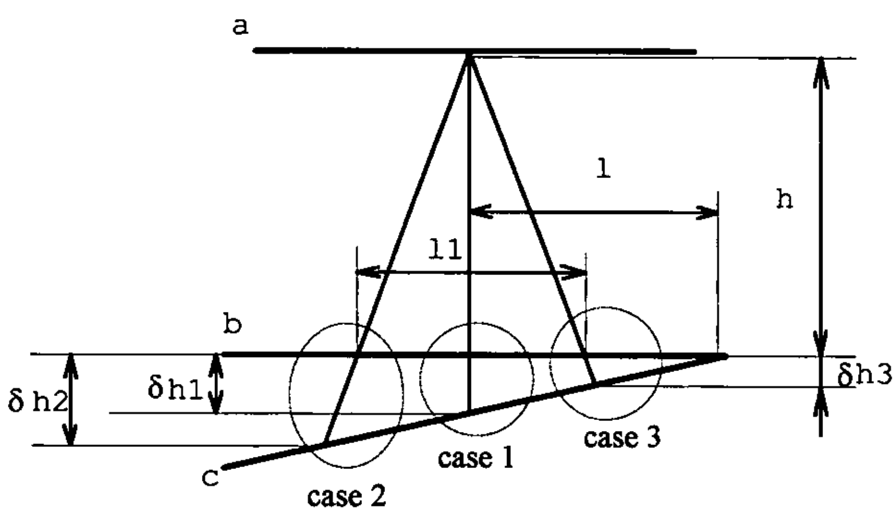


Figure 8.19 Thermal drift of frame and simplified model on Figure 8.20



Note1: a) top frame

b) middle frame before thremal drift

c) middle frame after thremal drift

Note2: The areas labelled by case1, case2 and case3 are presented in Figure 8.21-8.23.

Figure 8.20 Three cases of thermal drift of frame

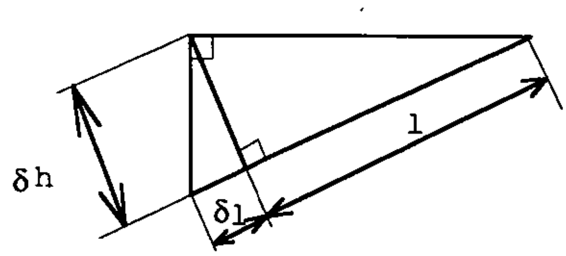


Figure 8.21 Case 1 of thermal drift of frame

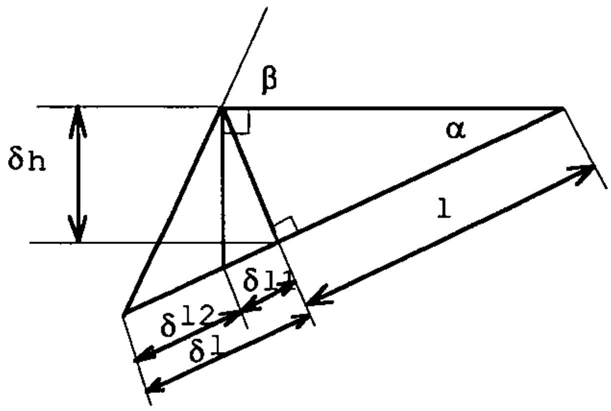


Figure 8.22 Case 2 of thermal drift of frame

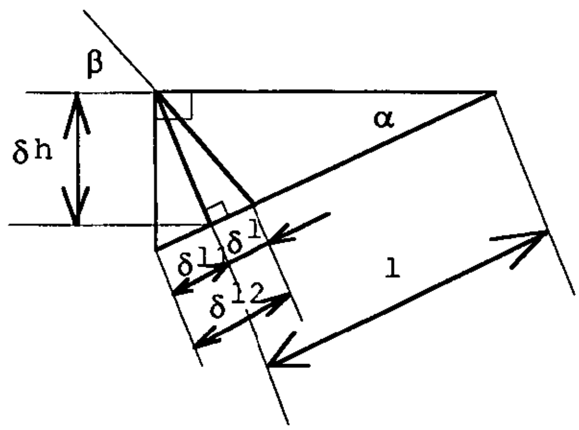


Figure 8.23 Case 3 of thermal drift of frame

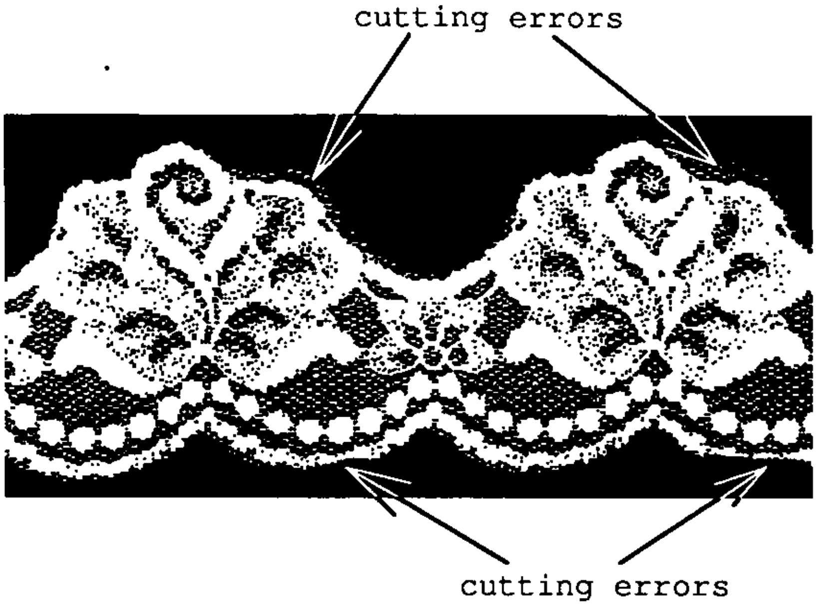


Figure 8.24 Example of cutting errors

Chapter 9

CONCLUSIONS AND FURTHER RESEARCH

9.1 CONCLUSIONS

This thesis has addressed the question of pattern cutting on elastic material. A study of a computer vision system capable of finding a cutting path on a piece of patterned material has been given. Lace separation is an example of pattern cutting in this thesis. During the research, a map-guided approach to enable the vision system to find a predetermined cutting path on the lace has been investigated, developed and implemented. The research has focused on: start point finding; tracking along a desired path; maintaining stable working condition for the vision system.

The methods of finding the start point have enabled the vision system to register the parts of an incoming image from the lace to be cut with a pre-stored map for further tracking along a desired path on the lace. A modified Hough transform can find a desired feature on the image with up to 20% stretch in one direction, with or without a moderate degree of pattern skew. The method to find the start point by using the tracking algorithm has been proved to be faster than the Hough transform and with reduced memory requirement. Furthermore, the latter method has enabled the vision system to quickly switch from finding the start point to tracking a cutting path or vice versa because the two tasks use the same algorithm.

The weighted incremental tracking method can cope with an arbitrary geometric relationship between a patterned surface and the path to be followed across it. The minimised computational requirement of the tracking method has enabled it to track high resolution images in real-time and also made it possible to be implemented in front-end correlators to achieve higher processing speed.

In order to maintain stable working condition for the vision system, the areas of light source, threshold and synchronisation have been studied. A DC powered high frequency fluorescent tube has proved an ideal source for back illumination of the lace cutting process. A threshold selection method for the comparator to produce binary images for pattern cutting has accommodated the illumination variations which are caused by tube ageing and working temperature drift. A distance counter and a double pulse generator have synchronised the scanning action of the line-scan camera with the movement of the lace and have enabled the image to be kept consistent regardless of variation of the lace speed.

The vision system based study in this thesis has successfully guided the laser beam to scallop and centre cut both rigid and stretch lace on test rig 3. Twin beam simultaneously cutting has been achieved at 250mm/s lace feed velocity and high speed single beam cutting using a front end hardware correlator has been achieved at speeds in excess of 1m/s.

9.2 FURTHER RESEARCH

Further research will be in the following areas;

1. Automatic reference map production,
2. Parallelization of the tracking algorithm,
3. Combination of cutting and inspection.

9.2.1 Reference Map Production

The tracking and start point finding algorithm based on a reference map have pushed back the research frontier of the pattern cutting. A manual method for producing a reference map has been developed during the study for the purpose of studying the tracking and start point finding methods. When galvanometer performance is related to the frequency of the guiding line, it can be seen that more study is required to define the guiding line to consider the galvanometer performance. Automation of map production is another requirement.

9.2.2 Parallelization of the Tracking Algorithm

In some applications, higher processing speeds will be required. This means more computational capacity. One way to increase the capacity is parallel computing. The parallel nature of the tracking algorithm will allow this technique to be fully exploited as multiprocessor systems become more powerful. There are at least two paths to make the tracking algorithm work in parallel. One is a parallel cross-correlation (eq.4.4). Another is to match one line of the incoming image in parallel with three consecutive lines of the map. More study of parallel computation is needed.

9.2.3 Combination of Cutting and Inspection

Most patterns on the patterned material in textile industry are produced by printing, knitting or weaving. The faults on the patterns should also be identified before the patterned material is sent to the customers. One way to inspect the patterned material might be to combine the pattern cutting and inspection together because the visual information of the pattern for the inspection has already been stored in the vision system for the pattern cutting. More study of how to share the information in the memory and how to identify the faults from the visual information is needed. The limitation of this approach is that cutting faults cannot be identified. Another approach is to put an inspection station downstream from the cutting station. Then the cutting faults as well as other faults can be identified and the inspection result of cutting faults can be fed back to cutting station for compensation.

REFERENCES

Albanesi, M G & Ferretti, M(1990), 'A Space Saving Approach to the Hough Transform' 10th International Conference on Pattern Recognition, 16-21 June 1990, Atlantic IEEE Computer Society Press (C1990.0818620625) Vol 2 pp472-475.

Ameziane, M., bonnet, P. & Postaire, J.G.(1985): 'Vision applied to a cutting process in the textile industry', Proceedings of the 7th International Conference on Automatic Inspection and Production Control, Birmingham, England, 26-28 March 1985, Publ by IFS(Publ) Ltd, Kempston, England, pp185-193.

Atkinson, A.D.S.(1944), 'Fluorescent Lighting', George Newnes Limited, London.

Batchelor,B.G., Hill,D.A., & Hodgson, D.C.(1985), (Edt.) 'Automated Visual Inspection', IFS Ltd, North-Holland.

Ballard, D. H.(1981), 'Generalizing the Hough Transform to Detect Arbitrary Shapes', *Pattern Recognition,* Vol. 13, pp111-122.

Bretschi, J.(1979), 'Automated Inspection Systems for Industry', IFS Publications Ltd, UK.

Canny, J. F.(1986), 'A Computational Approach to Edge Detection', *IEEE Trans. Pattern Anal. Machine Intell.* Vol. PAMI-8, No.6, pp679-698.

Chapman, K. W., Johnson, W. C., & McLean, T. J.(1990), 'A High Speed Statistical Process Control Application of Machine Vision to Electronics Manufacturing', *Computer Industrial Engineering,* Vol. 19, No. 1-4, pp. 234-238.

Chassaing, R, & Horning D.W.(1990), *Digital Signal Processing with the TMS320C25*, John Wiley and Sons, Inc., Now York.

Cooling, J.E(1991), 'Software Design for Real-Time Systems', Chapman and Hall, London.

Data β (1991), 'DBV56 QUATTRO Hardware Users Guide', Data β, UK.

Dakin, J & Culshaw, B. (ed)(1988), 'Optical Fibre Sensors: Principles and components', Artech House, Boston.

Davies, E. R.(1990), 'Machine Vision theory algorithms practicalities', Academic Press, London.

Davies, E.R.(1984), 'Design of Cost-effective System for the Inspection of Certain Food Products During Manufacture', In: Pugh, A.(ed.) *Proceedings of the 4th International Conference on Robot Vision and Sensory Controls, London(9-11 October),* IFS (Publications)Ltd, Bedford and North-Holland, Amsterdam, pp.437-446.

Duda, R. O. & Hart, P. E.(1972), 'Use of the Hough Transformation to Detect Lines and Curces in Pictures', *Comm.ACM,* Vol.15, pp 11-15.

Edmonds, J. M. & Davies, E. R.(1991), 'High-Speed Processor for Realtime Visual Inspection', *Microprocessors and Microsystems,* Vol.15, No.1, pp.11-19, January/February.

ERA (1993), 'Surface Inspection', ERA Technology News, Winter.

Fairchild (1990), CCD manual.

Gibbons, R.(1992), 'Image Processing for Amorphous Foodstuffs', PhD thesis, Loughborough University of Technology.

Gregory, R.L.(1990), 'Eye and Brain The Psychology of Seeing', Fourth Edition, Weidenfeld and Nicolson, London.

Hall, G. & Matias, A.(1993), 'otation, Scale and Translation Invariant Template Matching on a Transputer Network', *Microprocessors and Microsystems,* Vol. 17, No. 6, pp 333-340.

Hannah, J. & Stephens, R. C.(1984), 'Mechanics of Machines', Edward Arnold, Scotland.

Hanover (1991), 'Intelligent Laser Cutter', Via Combattenti, the 11th International Exhibition of Textile Machinary.

Harris Corporation (1991), 'Data Sheet, HSP45256, file number 2814.1'.

Horn, B. K. P.(1986), 'Robot Vision', The MIT Press.

Hough, P. V. C.(1966), 'Method and Means for Recognising Complex Patterns', *US Patent 3069654.*

Humphreys, G.W. & Riddoch, M. J.(1987), 'To See But Not To See A Case Study of Visual Agnosia', Lawrence Erlbaum Associates Ltd., London.

Inman, D. J. (1994), 'Engineering Vibration', Prentice-Hall International (UK) Limited, London.

Jackson, M.R., Preston, M.E. & King, T.G.(1993), 'Integrated Vision and Laser Cutting System for Fabric Processing', Mechatronics in Textile Industries One-day Seminar, Loughborough, 22 March.

Jain, A. K.(1989), 'Fundamentals of Digital Image Processing', Prentice-Hall International Editions, US.

Johnson, J.R(1989), 'Introduction to Digital Signal Processing', Prentice-Hall International, Inc. London.

Kamemaru, S., Itoh, H., and Yano, J.(1992), 'Character Recognition by Feature Extraction Using Cross-Correlation Signals from a Matched Filter', *Optical Engineering*, Vol. 32, No. 1, pp26-32.

Karkanis, S., Metaxaki-Kossionides, C., & Dimitriadis, B.(1989), 'A Machine-Vision Quality Inspection System for Textile industries supported by Parallel Multitransputer Architecture', *Microprocessing and Microprogramming*, No.28, pp.247-252.

Khoury, J.(1991), 'PC-based Vision in Laser Cutting of Upholstery Fabric', *Photonics Spectra,* August, pp123-124.

Kimoto, I., and Yamafuji, K.(1986), 'Automation of Sealant Painting and Lace Cutting Using Pattern Tracking Techniques', *The International Journal of Advanced Manufacturing Technology,* Vol. 1, No.4, pp101-107.

Lawton, D. T., Levitt, T. S., McConnell, C., & Glicksman, J.(1986), 'Terrain Models for an Autonomous Land Vehicle', In *Proc. 1986 IEEE International Conf. on Robotics and Automation.* San Francisco, California, pp.2043-2051 (IEEE Cat. 86CH2282-2), April 7-10.

Levey, S.M.(1983), 'Lace A History', Victoria & Albert Museum.

Levine, M. D.(1985), 'Vision in Man and Machine', McGraw-Hill Book Company.

Lowe, D., & Richards, J.(1984), 'The lace Heritage', Nottingham Lace Centre Limited, Nottingham.

Lyvers, E.R. & Mitchell, O. R.(1988), 'Precision edge contrast and Orientation estimation'. *IEEE Trans. Pattern Anal. Machine Intell.* Vol. PAMI-10, No. , pp927-937.

Maitre, H and Wu, Y(1989), 'A Dynamic Programming Algorithmm for Elastic Registration of Distorted Pictures Based on Autoregressive Model' IEEE Trans. On Acoustic, Speech and Signal Processing Vol 37 No 2, pp288-296.

Mao, Z. Q. (1992), 'Optoelelctronic High Order Feedback Neural NETwork (HOFNET)', PhD thesis, Department of Electronic and Electrical Engineering, University College London, London, UK.

Merlin, P M & Farber, D J(1975), 'A Parallel Mechanism for Detecting Curves in Pictures', *IEEE Trans. on Computer,* Vol C-24 pp96-98.

Moshfeghi, M.(1991), 'Elastic Matching of Multimodality Medical Images', *Graphical Model and Image Processing,* Vol 53 No 3 pp271-282.

Motorola (1989), 'DSP56000UM/AD, DSP56000/DSP56001 Digital Signal Processor User's Manual'.

Norton-Wayne, L., Machellar, A., & Nicklin C.(1989), 'Measurement of Garment Dimensions Using Machine Vision', *Third International Conference on Image Processing and Its Applicaitons*, 18-20 July 1989, Warwick, UK, IEE conference publication No. 307, pp.197-201.

Norton-Wayne, L.(1991), 'Inspection of Lace Using Machine Vision', *Computer Graphics Forum*, Vol. 10, pp.113-119.

Osram Limited (1991), 'Perfect Lighting with Electronic Control Components'.

Philips Lighting Limited (1991), 'Philips Lighting'.

Preston, M.(1989), 'Lace separation project proposal to SERC'.

Proakis, J G & Manolakis, D G(1988), 'Introduction to digital signal processing', Macmillan Publishing Company, New York.

Reynolds C.(1984), 'Lenses', Focal Press, London.

Rubinstein, J., Segman, J. and Zeevi, Y. (1990), 'Recognition of Distorted Patterns by Invariance Kernals', Proc. *10th ICPR*, pp159-164, IEEE Computer Press.

Ruocco, S. R. (1987), 'Robot Sensors and Transducers', Open University Press, UK.

Russell, R.A., and Wong, P.(1989), 'An Application of Computer Vision to Lace Cutting', *Robotics and Autonomous Systems*, No. 5, pp.91-96.

Ryan, L. & Hardaker, C.(1991), 'The CIMTEX Project: Part Three Vision Cell: A look at The Research Programme', *Knitting International*, pp68-69, June.

Skilton, J(1988), 'Automated Lace Inspection', Leicester Polytechnic, Report No. LRAJCS-1.

Spencer, D.J.(1983), 'Knitting Technology', Pergamon Press.

Stubbs,D.F.(1979), 'Frequency and the Brain', Life Sciences, Vol. 18, pp.345-394.

Thomas, G.B & Ross, L.(1988), 'Calculus and Analytic Geometry', Assison-Wesley Pub. Co., US.

Tao, L.(1991), 'First Technical Report: Vision System', Department of Mechanical Engineering, Loughborough University of Technology, August.

Turk, M. A., Morgenthaler, D. G., Gremban, K. D., & Marra, M.(1988), 'VITS- A Vision System for Autonomous Land Vehicle Navigation', *IEEE Trans. Pattern Anal. Machine Intell.* Vol. PAMI-10, No.3, pp342-361.

Wood, E. J. and Hodgson, R.M.(1989), 'Carpet Texture Measurement Using Image Analysis', *Textile Research Institute,* Jan, pp.1-12.

Yoda, H., Ohuchi, Y., Taniguchi,Y., & Ejiri, M.(1988), 'An Automatic Wafer Inspection System Using Piplined Image Processing Techniques', *IEEE Trans. Pattern Anal. Machine Intell.* Vol. PAMI-10, No.1, pp4-15.

Zheng, Q. & Chellappa, R.(1993), 'A Computational Approach to Image Registration', *IEEE Trans. on Image Processing*, Vol2, No.3, pp311-326.

APPENDIXES

Appendix 1 Published Papers and Patent

King, T.G., Tao, L.G., Jackson, M.R., Preston, M.E. & Yang, S.(1993), "Computer-Vision Controlled High-Speed Laser Cutting of Lace", Proc. 2nd. Int. Conf. on Computer Integrated Manufacturing (ICCIM '93), Sep. 6-10, Singapore, 1993. Vol.2, pp 929-936, World Scientific Publishing, Singapore.

King, T.G., Tao, L.G., Preston, M.E.and Jackson, M.R.(1993), "Real-Time Tracking of Patterns on Deformable Materials Using DSP", IEE SERTA '93, 13-14 Sept. 1993, Cirencester UK.

International Patent Publication No. WO 94/03301, "Automatic Operations on Materials', Filed 5 August 1993.

Preston, M.E., Jackson, M. R., Yang, S., Tao, L.G., and King, T.G, "The Design of a High-Speed Lace Cutting of Lace", Proc. Int. Conf. on Machine Automation (ICMA '94) pp473-486, Tampere, Finland, Feb 1994, ISBN 951-722-107-X.

Jackson, M.R., Preston, M.E., Yang, S. and Tao, L.G., "Laser Cutting Parameters for the High Speed Cutting of Net Fabrics", The European Symposium on Optics for Productivity in Manufacturing, Frankfurt/Main, FR Germany, 20-24 June 1994

Jackson, M.R., Preston, M.E., Tao, L.G., Yang, S. and King, T.G., "High Speed Laser Cutting of Lace", The European Symposium on Optics for Productivity in Manufacturing, Frankfurt/Main, FR Germany, 20-24 June 1994

King, T.G., Tao, L.G.(1994), "An Incremental Real-Time Pattern Tracking Algorithm for Line-Scan Camera Application", Mechatronics, Vol 4, pp503-516.

King, T.G., Tao, L.G., Jackson, M.R., Preston, M.E. & Yang, S.(1994), "A Mechatronic Lace Scalloping Machine using Computer Vision", Proceedings Mechatronics and Machine Vision in Practice, Int. Conf. on Mechatronics and Machine Vision in Practice, Sep. 13-15, Toowoomba, Australia, 1994. pp 22-27, ISBN 0-8186-6300-6.

Appendix 2 Experiment on Tracking (Table 4.3)

Column 1

Source file: lac1.b m	Table file: Table B of Table 4.1	the Width of matching: 128 pixels	Curve file: lac1.gui

Avaerage tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.015	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
y-distortion 2%	0 .04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.01
y-distortion 3%	0.135	0.125	0.115	0.105	0.105	0.095	0.095	0.105	0.085	0.075	0.055
y-distortion 4%	0.105	0.085	0.085	0.085	0.075	0.065	0.065	0.055	0.045	0.045	0.035
y-distortion 5%	0.19	0.17	0.17	0.17	0.15	0.15	0.11	0.09	0.09	0.06	0.06
y-distortion 6%	0.185	0.205	0.185	0.155	0.165	0.135	0.135	0.125	0.125	0.105	0.075
y-distortion 7%	0.215	0.205	0.205	0.175	0.145	0.145	0.105	0.115	0.105	0.075	0.075
y-distortion 8%	0.27	0.24	0.24	0.23	0.22	0.23	0.19	0.17	0.16	0.14	0.09
y-distortion 9%	0.265	0.265	0.235	0.205	0.195	0.185	0.195	0.175	0.145	0.115	0.095
y-distortion 10%	0.37	0.35	0.33	0.31	0.29	0.26	0.26	0.235	0.195	0.145	0.115
y-distortion 11%	0.34	0.3	0.31	0.27	0.25	0.24	0.19	0.16	0.14	0.12	0.11
y-distortion 12%	0.355	0.345	0.335	0.305	0.295	0.265	0.265	0.225	0.185	0.185	0.155
y-distortion 13%	0.34	0.33	0.33	0.31	0.31	0.27	0.26	0.25	0.23	0.2	0.17
y-distortion 14%	0.43	0.4	0.38	0.35	0.35	0.31	0.33	0.30	0.22	0.22	0.16
y-distortion 15%	0.415	0.405	0.415	0.365	0.365	0.345	0.265	0.255	0.225	0.175	0.165

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.025	0.025	0.025	0.025	0.025	0.015	0.005	0.005	0.005	0.005	0.005
y-distortion 2%	0 .06	0.06	0.06	0.05	0.05	0.04	0.03	0.03	0.03	0.03	0.01
y-distortion 3%	0.145	0.145	0.145	0.145	0.145	0.145	0.095	0.105	0.085	0.065	0.045
y-distortion 4%	0.145	0.135	0.135	0.115	0.105	0.095	0.075	0.055	0.045	0.035	0.035
y-distortion 5%	0.21	0.21	0.2	0.2	0.18	0.17	0.15	0.12	0.09	0.08	0.06
y-distortion 6%	0.225	0.175	0.155	0.165	0.165	0.145	0.125	0.125	0.105	0.105	0.06
y-distortion 7%	0.245	0.245	0.225	0.175	0.185	0.155	0.135	0.145	0.095	0.075	0.065
y-distortion 8%	0.29	0.28	0.25	0.25	0.26	0.25	0.23	0.20	0.15	0.14	0.09
y-distortion 9%	0.295	0.265	0.255	0.235	0.225	0.205	0.205	0.195	0.175	0.115	0.095
y-distortion 10%	0.46	0.40	0.34	0.33	0.33	0.3	0.265	0.225	0.195	0.135	0.125
y-distortion 11%	0.34	0.35	0.3	0.31	0.26	0.22	0.22	0.19	0.14	0.11	0.09
y-distortion 12%	0.405	0.365	0.345	0.355	0.345	0.305	0.275	0.265	0.215	0.165	0.125
y-distortion 13%	0.41	0.41	0.41	0.38	0.34	0.32	0.30	0.28	0.25	0.22	0.15
y-distortion 14%	0.45	0.44	0.43	0.4	0.39	0.36	0.34	0.30	0.28	0.24	0.15
y-distortion 15%	0.485	0.455	0.475	0.465	0.425	0.345	0.335	0.305	0.255	0.225	0.145

Maxmum tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	1	1	1	1	1	1	1	1	1	1	1
y-distortion 2%	1	1	1	1	1	1	1	1	1	1	1
y-distortion 3%	1	1	1	1	1	1	1	1	1	1	1
y-distortion 4%	1	1	1	1	1	1	1	1	1	1	1
y-distortion 5%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 9%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 10%	3	3	3	3	3	3	2	2	2	2	2

y-distortion 11%	3	3	3	3	3	3	2	2	2	2	2
y-distortion 12%	3	3	3	3	3	3	3	2	2	3	2
y-distortion 13%	3	3	3	3	3	3	3	2	2	2	2
y-distortion 14%	4	4	4	4	4	4	3	3	3	3	2
y-distortion 15%	4	4	4	4	3	3	3	3	2	2	2

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 2%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 3%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 4%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 5%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 9%	3	3	3	3	3	3	2	2	2	2	2
y-distortion 10%	3	3	3	3	2	3	2	2	2	2	2
y-distortion 11%	3	3	3	3	3	2	2	2	2	2	2
y-distortion 12%	3	3	3	3	2	2	3	2	2	3	2
y-distortion 13%	3	3	3	3	3	3	3	3	2	2	2
y-distortion 14%	4	4	4	4	3	3	3	3	2	2	2
y-distortion 15%	4	4	4	3	3	3	3	3	3	2	2

Column 2

Source file: lac2.b_m	Table file: Table B of Table 4.1	the Width of matching: 96 pixels	Curve file: lac2.gui

Avaerage tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.085	0.085	0.065	0.065	0.065	0.065	0.045	0.025	0.025	0.025	0.005
y-distortion 2%	0 .08	0.08	0.06	0.06	0.06	0.05	0.05	0.07	0.05	0.03	0.03
y-distortion 3%	0.11	0.11	0.11	0.11	0.09	0.09	0.1	0.11	0.09	0.07	0.05
y-distortion 4%	0.135	0.125	0.125	0.125	0.095	0.085	0.085	0.095	0.075	0.055	0.035
y-distortion 5%	0.155	0.145	0.135	0.135	0.135	0.105	0.105	0.125	0.095	0.085	0.075
y-distortion 6%	0.18	0.18	0.15	0.13	0.1	0.1	0.1	0.1	0.12	0.1	0.07
y-distortion 7%	0.19	0.18	0.18	0.17	0.14	0.17	0.175	0.17	0.16	0.12	0.1
y-distortion 8%	0.29	0.29	0.29	0.28	0.25	0.24	0.24	0.25	0.19	0.135	0.06
y-distortion 9%	0.195	0.195	0.195	0.195	0.195	0.185	0.185	0.2	0.145	0.115	0.075
y-distortion 10%	0.255	0.265	0.255	0.255	0.235	0.235	0.235	0.235	0.145	0.125	0.075
y-distortion 11%	0.32	0.3	0.27	0.27	0.28	0.25	0.2	0.195	0.15	0.16	0.1
y-distortion 12%	0.24	0.25	0.25	0.24	0.22	0.23	0.23	0.21	0.18	0.16	0.09
y-distortion 13%	0.475	0.465	0.435	0.415	0.415	0.395	0.395	0.315	0.335	0.215	0.185
y-distortion 14%	0.265	0.195	0.205	0.195	0.185	0.185	0.205	0.185	0.175	0.135	0.095
y-distortion 15%	0.37	0.33	0.33	0.33	0.33	0.29	0.28	0.27	0.22	0.25	0.15

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.115	0.115	0.115	0.115	0.115	0.125	0.025	0.025	0.025	0.025	0.005
y-distortion 2%	0.14	0.14	0.12	0.125	0.115	0.105	0.06	0.08	0.05	0.03	0.03
y-distortion 3%	0.14	0.14	0.14	0.14	0.14	0.14	0.11	0.12	0.1	0.06	0.05
y-distortion 4%	0.145	0.155	0.155	0.165	0.15	0.13	0.12	0.14	0.115	0.06	0.035
y-distortion 5%	0.185	0.195	0.185	0.17	0.18	0.175	0.155	0.165	0.145	0.125	0.095

y-distortion 6%	0.24	0.19	0.19	0.185	0.175	0.175	0.13	0.13	0.14	0.12	0.1
y-distortion 7%	0.26	0.25	0.25	0.21	0.185	0.225	0.21	0.2	0.2	0.12	0.06
y-distortion 8%	0.35	0.34	0.33	0.29	0.26	0.23	0.29	0.3	0.22	0.15	0.08
y-distortion 9%	0.29	0.295	0.295	0.295	0.295	0.235	0.245	0.24	0.18	0.11	0.075
y-distortion 10%	0.355	0.35	0.37	0.345	0.33	0.33	0.255	0.255	0.135	0.115	0.105
y-distortion 11%	0.36	0.35	0.35	0.31	0.36	0.33	0.31	0.26	0.17	0.12	0.085
y-distortion 12%	0.42	0.40	0.36	0.35	0.35	0.27	0.27	0.23	0.23	0.16	0.1
y-distortion 13%	0.575	0.515	0.465	0.465	0.455	0.365	0.415	0.385	0.305	0.255	0.235
y-distortion 14%	0.385	0.375	0.325	0.325	0.255	0.245	0.245	0.225	0.175	0.165	0.145
y-distortion 15%	0.54	0.51	0.49	0.46	0.43	0.41	0.33	0.29	0.32	0.26	0.18

Maxmum tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 2%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 3%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 4%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 5%	3	3	3	3	3	3	3	2	2	2	2
y-distortion 6%	3	3	3	3	3	3	3	3	3	3	2
y-distortion 7%	3	3	3	3	3	3	3	3	3	3	3
y-distortion 8%	3	3	3	3	3	3	3	3	3	3	2
y-distortion 9%	3	3	3	3	3	3	3	3	3	3	2
y-distortion 10%	3	3	3	3	3	3	3	3	3	3	2
y-distortion 11%	4	4	4	4	4	4	4	4	3	3	2
y-distortion 12%	4	4	4	4	4	4	4	4	4	4	2
y-distortion 13%	5	5	5	5	5	5	5	4	4	3	3
y-distortion 14%	5	5	5	5	5	5	5	4	3	2	2
y-distortion 15%	4	3	3	3	3	3	3	3	3	3	3

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 2%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 3%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 4%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 5%	3	3	3	2	2	2	2	2	2	2	2
y-distortion 6%	3	3	3	3	3	3	3	3	3	2	2
y-distortion 7%	3	3	3	3	3	3	3	3	3	3	1
y-distortion 8%	3	3	3	3	3	3	3	3	3	2	2
y-distortion 9%	3	3	3	3	4	3	3	3	3	2	1
y-distortion 10%	3	4	4	4	3	3	3	3	3	2	2
y-distortion 11%	4	4	4	4	4	4	4	4	3	2	1
y-distortion 12%	4	4	4	4	4	4	4	3	4	2	2
y-distortion 13%	5	5	5	5	5	5	3	3	3	3	3
y-distortion 14%	5	5	5	5	5	5	3	3	3	2	2
y-distortion 15%	4	4	4	3	3	3	3	3	3	3	3

Column 3

Source file: lac3.b_m	Table file: Table B of Table 4.1	the Width of matching: 96 pixels	Curve file: lac3.gui

Avaerage tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.165	0.165	0.125	0.085	0.085	0.085	0.085	0.065	0.005	0.005	0.005
y-distortion 2%	0.165	0.165	0.125	0.085	0.085	0.085	0.085	0.065	0.005	0.005	0.005
y-distortion 3%	0.12	0.12	0.1	0.1	0.07	0.03	0.01	0.01	0.01	0.01	0.01
y-distortion 4%	0.17	0.14	0.14	0.12	0.1	0.09	0.08	0.04	0.01	0.01	0.01
y-distortion 5%	0.17	0.15	0.1	0.09	0.07	0.07	0.04	0.03	0.01	0.01	0.01
y-distortion 6%	0.225	0.195	0.145	0.145	0.115	0.095	0.075	0.085	0.055	0.045	0.03
y-distortion 7%	0.245	0.225	0.215	0.205	0.145	0.125	0.105	0.075	0.045	0.045	0.045
y-distortion 8%	0.355	0.245	0.255	0.215	0.175	0.105	0.105	0.065	0.025	0.025	0.025
y-distortion 9%	0.41	0.35	0.26	0.22	0.22	0.19	0.14	0.1	0.07	0.04	0.045
y-distortion 10%	0.48	0.5	0.46	0.31	0.29	0.28	0.16	0.16	0.12	0.07	0.06
y-distortion 11%	0.5	0.49	0.47	0.46	0.31	0.29	0.27	0.27	0.15	0.09	0.05
y-distortion 12%	0.795	0.755	0.655	0.645	0.535	0.515	0.445	0.425	0.285	0.115	0.06
y-distortion 13%	0.935	0.755	0.715	0.655	0.455	0.345	0.395	0.395	0.315	0.145	0.115
y-distortion 14%	0.89	0.85	0.73	0.73	0.67	0.5	0.31	0.3	0.3	0.3	0.245
y-distortion 15%	0.8	0.7	0.79	0.66	0.55	0.51	0.42	0.39	0.37	0.26	0.245

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.145	0.125	0.105	0.085	0.065	0.065	0.105	0.065	0.045	0.025	0.005
y-distortion 2%	0.145	0.125	0.105	0.085	0.065	0.065	0.105	0.065	0.045	0.025	0.005
y-distortion 3%	0.11	0.1	0.09	0.05	0.02	0.02	0.05	0.03	0.01	0.01	0.01
y-distortion 4%	0.14	0.12	0.1	0.08	0.08	0.05	0.06	0.06	0.05	0.01	0.01
y-distortion 5%	0.12	0.12	0.08	0.05	0.03	0.03	0.07	0.06	0.04	0.02	0.02
y-distortion 6%	0.225	0.215	0.165	0.145	0.105	0.095	0.095	0.105	0.075	0.045	0.03
y-distortion 7%	0.245	0.215	0.185	0.175	0.105	0.095	0.125	0.095	0.065	0.045	0.065
y-distortion 8%	0.315	0.215	0.185	0.165	0.105	0.075	0.135	0.075	0.065	0.025	0.025
y-distortion 9%	0.42	0.37	0.22	0.14	0.13	0.13	0.13	0.14	0.12	0.08	0.045
y-distortion 10%	0.56	0.47	0.41	0.27	0.26	0.23	0.27	0.14	0.11	0.1	0.05
y-distortion 11%	0.59	0.59	0.55	0.46	0.83	0.29	0.31	0.22	0.2	0.11	0.08
y-distortion 12%	0.775	0.925	0.745	0.735	0.675	0.515	0.345	0.325	0.395	0.145	0.07
y-distortion 13%	0.915	0.985	0.775	0.675	0.665	0.495	0.385	0.365	0.335	0.275	0.105
y-distortion 14%	0.92	0.75	0.75	0.73	0.57	0.43	0.41	0.33	0.38	0.365	0.185
y-distortion 15%	0.96	0.72	0.66	0.72	0.67	0.63	0.6	0.48	0.36	0.425	0.205

Maxmum tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	2	2	1	1	1
y-distortion 2%	2	2	2	2	2	2	2	2	1	1	1
y-distortion 3%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 4%	2	2	2	2	2	2	2	2	1	1	1
y-distortion 5%	2	2	2	2	2	2	2	2	1	1	1
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	8	2	2	2	2	2	2	2	2	2	2
y-distortion 9%	8	8	2	2	2	2	2	2	2	2	2
y-distortion 10%	11	11	11	8	8	8	2	2	2	2	2
y-distortion 11%	10	10	10	11	8	8	8	8	2	2	2
y-distortion 12%	16	9	10	9	8	8	8	8	8	3	3
y-distortion 13%	16	16	9	9	8	8	8	8	8	3	3
y-distortion 14%	16	16	16	16	16	16	8	8	8	8	8
y-distortion 15%	9	9	16	9	9	9	8	8	8	8	8

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	2	2	2	2	1

y-distortion 2%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 3%	2	2	2	2	1	1	2	2	1	1	1
y-distortion 4%	2	2	2	2	2	2	2	2	2	1	1
y-distortion 5%	2	2	2	2	1	1	2	2	2	2	2
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	8	2	2	2	2	2	2	2	2	2	2
y-distortion 9%	8	8	2	2	2	2	2	2	2	2	2
y-distortion 10%	11	10	10	8	8	8	8	2	2	2	2
y-distortion 11%	11	11	11	10	10	8	10	8	8	2	2
y-distortion 12%	16	16	16	16	16	16	8	8	8	3	3
y-distortion 13%	16	16	16	16	16	16	8	8	8	8	3
y-distortion 14%	16	16	16	16	16	16	16	8	8	8	8
y-distortion 15%	16	16	16	16	16	16	10	8	8	8	8

Column 4

Source file: lac4.b_m	Table file: Table B of Table 4.1	the Width of matching: 64 pixels	Curve file: lac4.gui

Avaerage tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.045	0.045	0.045	0.045	0.045	0.045	0.025	0.025	0.025	0.025	0.025
y-distortion 2%	0.075	0.065	0.065	0.065	0.065	0.065	0.045	0.045	0.035	0.035	0.045
y-distortion 3%	0.07	0.07	0.07	0.06	0.06	0.06	0.04	0.04	0.04	0.04	0.04
y-distortion 4%	0.1	0.08	0.09	0.06	0.06	0.06	0.04	0.04	0.04	0.04	0.04
y-distortion 5%	0.12	0.12	0.12	0.12	0.12	0.12	0.1	0.07	0.08	0.05	0.02
y-distortion 6%	0.125	0.095	0.085	0.075	0.075	0.065	0.075	0.055	0.055	0.045	0.025
y-distortion 7%	0.165	0.165	0.165	0.155	0.145	0.125	0.095	0.085	0.075	0.085	0.065
y-distortion 8%	0.185	0.175	0.175	0.155	0.135	0.115	0.075	0.085	0.075	0.065	0.06
y-distortion 9%	0.18	0.18	0.18	0.16	0.18	0.15	0.13	0.11	0.1	0.09	0.07
y-distortion 10%	0.2	0.2	0.18	0.18	0.16	0.15	0.12	0.08	0.11	0.08	0.05
y-distortion 11%	0.195	0.205	0.185	0.185	0.175	0.175	0.135	0.115	0.125	0.105	0.045
y-distortion 12%	0.235	0.215	0.215	0.195	0.185	0.165	0.115	0.115	0.115	0.115	0.055
y-distortion 13%	0.25	0.225	0.215	0.215	0.155	0.145	0.135	0.135	0.145	0.115	0.08
y-distortion 14%	0.28	0.28	0.27	0.24	0.18	0.17	0.15	0.15	0.14	0.14	0.13
y-distortion 15%	0.26	0.25	0.2	0.2	0.2	0.17	0.17	0.16	0.14	0.13	0.07

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.085	0.085	0.085	0.085	0.075	0.045	0.035	0.055	0.045	0.025	0.025
y-distortion 2%	0.115	0.115	0.105	0.105	0.095	0.065	0.045	0.065	0.055	0.035	0.035
y-distortion 3%	0.11	0.11	0.11	0.1	0.09	0.06	0.05	0.07	0.06	0.04	0.04
y-distortion 4%	0.13	0.11	0.1	0.1	0.08	0.08	0.05	0.07	0.06	0.04	0.04
y-distortion 5%	0.17	0.16	0.16	0.15	0.13	0.12	0.09	0.1	0.08	0.05	0.04
y-distortion 6%	0.135	0.125	0.125	0.115	0.115	0.105	0.065	0.065	0.065	0.065	0.055
y-distortion 7%	0.235	0.215	0.215	0.215	0.185	0.175	0.155	0.105	0.095	0.105	0.085
y-distortion 8%	0.235	0.235	0.225	0.215	0.185	0.175	0.145	0.125	0.095	0.095	0.055
y-distortion 9%	0.24	0.22	0.21	0.2	0.21	0.2	0.17	0.14	0.13	0.09	0.08
y-distortion 10%	0.27	0.26	0.25	0.25	0.24	0.17	0.13	0.12	0.09	0.08	0.06
y-distortion 11%	0.275	0.255	0.245	0.245	0.245	0.205	0.185	0.135	0.135	0.095	0.055
y-distortion 12%	0.315	0.305	0.295	0.295	0.285	0.225	0.145	0.125	0.125	0.115	0.095
y-distortion 13%	0.345	0.335	0.315	0.305	0.245	0.215	0.145	0.145	0.135	0.115	0.095
y-distortion 14%	0.36	0.37	0.31	0.28	0.25	0.21	0.18	0.17	0.18	0.14	0.1
y-distortion 15%	0.39	0.36	0.3	0.28	0.27	0.25	0.2	0.17	0.13	0.11	0.1

Maxmum tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 2%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 3%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 4%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 5%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 9%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 10%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 11%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 12%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 13%	3	3	3	3	3	3	3	2	2	2	2
y-distortion 14%	3	3	3	3	3	3	3	2	2	2	2
y-distortion 15%	3	3	3	3	3	3	2	2	2	2	1

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 2%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 3%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 4%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 5%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 9%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 10%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 11%	2	2	2	2	2	2	2	2	2	2	1
y-distortion 12%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 13%	3	3	3	3	2	2	2	2	2	2	2
y-distortion 14%	4	4	3	3	2	3	2	2	2	2	2
y-distortion 15%	4	3	3	3	3	2	2	2	2	2	2

Column 5

Source file: lac3.b_m	Table file: Table B of Table 4.1	the Width of matching: 128 pixels	Curve file: lac3.gui

Avaerage tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.045	0.025	0.025	0.025	0.025	0.005	0.005	0.005	0.005	0.005	0.005
y-distortion 2%	0.035	0.025	0.025	0.025	0.025	0.005	0.005	0.005	0.005	0.005	0.005
y-distortion 3%	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
y-distortion 4%	0.1	0.09	0.08	0.06	0.05	0.05	0.05	0.03	0.03	0.03	0.01
y-distortion 5%	0.095	0.065	0.065	0.065	0.055	0.055	0.045	0.045	0.035	0.02	0.02
y-distortion 6%	0.125	0.095	0.095	0.085	0.085	0.075	0.075	0.075	0.075	0.065	0.055
y-distortion 7%	0.2	0.18	0.17	0.16	0.145	0.135	0.115	0.085	0.065	0.045	0.035
y-distortion 8%	0.305	0.255	0.235	0.205	0.115	0.125	0.105	0.075	0.065	0.045	0.035
y-distortion 9%	0.44	0.38	0.33	0.32	0.27	0.18	0.16	0.15	0.04	0.03	0.03
y-distortion 10%	0.465	0.42	0.39	0.33	0.31	0.3	0.21	0.19	0.07	0.07	0.07
y-distortion 11%	0.7	0.61	0.51	0.5	0.41	0.39	0.35	0.32	0.16	0.07	0.06
y-distortion 12%	0.775	0.735	0.635	0.625	0.535	0.435	0.305	0.195	0.185	0.085	0.055
y-distortion 13%	0.815	0.72	0.725	0.665	0.645	0.365	0.355	0.315	0.175	0.095	0.065

y-distortion 14%	1.01	0.92	0.8	0.8	0.77	0.63	0.52	0.49	0.48	0.36	0.19
y-distortion 15%	0.92	0.91	0.89	0.72	0.69	0.69	0.49	0.48	0.45	0.41	0.24

coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	0.065	0.065	0.065	0.045	0.025	0.025	0.005	0.005	0.005	0.005	0.005
y-distortion 2%	0.075	0.065	0.045	0.045	0.025	0.025	0.005	0.005	0.005	0.005	0.005
y-distortion 3%	0.05	0.05	0.01	0.01	0.01	0.01	0.03	0.03	0.02	0.01	0.01
y-distortion 4%	0.14	0.11	0.11	0.11	0.11	0.07	0.05	0.03	0.03	0.03	0.03
y-distortion 5%	0.155	0.125	0.105	0.085	0.075	0.055	0.055	0.045	0.045	0.02	0.02
y-distortion 6%	0.195	0.155	0.115	0.115	0.115	0.095	0.115	0.095	0.085	0.085	0.075
y-distortion 7%	0.285	0.265	0.235	0.215	0.225	0.185	0.145	0.125	0.125	0.085	0.065
y-distortion 8%	0.515	0.425	0.335	0.315	0.205	0.165	0.085	0.065	0.055	0.055	0.035
y-distortion 9%	0.52	0.5	0.47	0.39	0.38	0.24	0.19	0.17	0.08	0.03	0.03
y-distortion 10%	0.585	0.545	0.5	0.41	0.4	0.39	0.23	0.23	0.12	0.09	0.06
y-distortion 11%	0.73	0.69	0.68	0.67	0.56	0.34	0.42	0.27	0.18	0.05	0.05
y-distortion 12%	0.995	0.955	0.935	0.995	0.925	0.705	0.455	0.315	0.215	0.095	0.075
y-distortion 13%	0.94	0.92	0.885	0.965	0.865	0.685	0.335	0.305	0.185	0.105	0.085
y-distortion 14%	1.18	1.01	0.98	0.87	0.74	0.61	0.64	0.58	0.58	0.55	0.265
y-distortion 15%	1.17	1	0.88	0.66	0.72	0.68	0.67	0.66	0.63	0.51	0.295

Maxmum tracking error

coefficient β, no noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	1	1	1	1	1	1
y-distortion 2%	2	2	2	2	2	1	1	1	1	1	1
y-distortion 3%	1	1	1	1	1	1	1	1	1	1	1
y-distortion 4%	2	2	2	1	1	1	1	1	1	1	1
y-distortion 5%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	8	8	8	8	2	2	2	2	2	2	2
y-distortion 9%	8	8	8	8	8	8	8	8	2	2	2
y-distortion 10%	8	8	8	8	8	8	8	8	2	2	2
y-distortion 11%	9	9	9	9	9	9	9	9	8	2	2
y-distortion 12%	10	10	9	9	9	9	9	8	8	3	3
y-distortion 13%	10	10	10	10	10	9	9	8	8	3	3
y-distortion 14%	10	10	10	9	9	8	8	8	8	8	8
y-distortion 15%	9	9	9	9	8	8	8	8	8	8	8

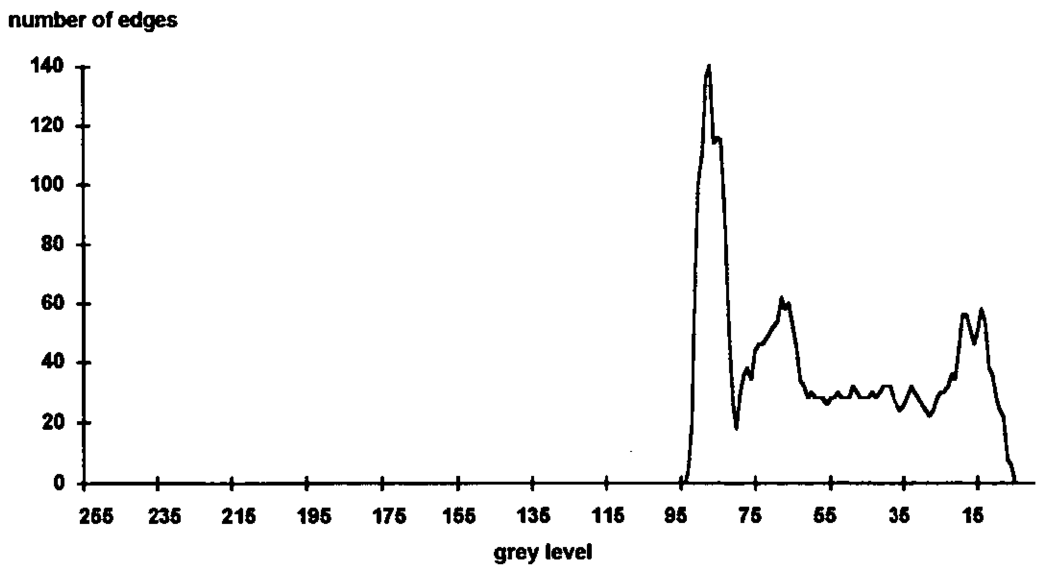
coefficient β, 2% noise	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
y-distortion 1%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 2%	2	2	2	2	2	2	1	1	1	1	1
y-distortion 3%	2	2	1	1	1	1	1	1	1	1	1
y-distortion 4%	2	2	2	2	2	1	1	1	1	1	1
y-distortion 5%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 6%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 7%	2	2	2	2	2	2	2	2	2	2	2
y-distortion 8%	10	10	8	8	2	2	2	2	2	2	2
y-distortion 9%	10	10	10	8	8	8	8	8	2	2	2
y-distortion 10%	11	11	10	8	8	8	8	8	2	2	2
y-distortion 11%	10	10	10	10	11	9	8	8	8	2	2
y-distortion 12%	16	16	10	10	10	10	9	8	8	3	3
y-distortion 13%	16	16	10	10	10	10	8	8	9	3	3
y-distortion 14%	16	10	10	10	10	10	8	8	8	8	8
y-distortion 15%	16	10	10	9	10	10	9	9	9	8	8

Appendix 3 Experiment on Threshold Decision

The grey level is set from 255 to 0 during the test, so the data and curve should be red from the highest grey level to the lowest grey level.

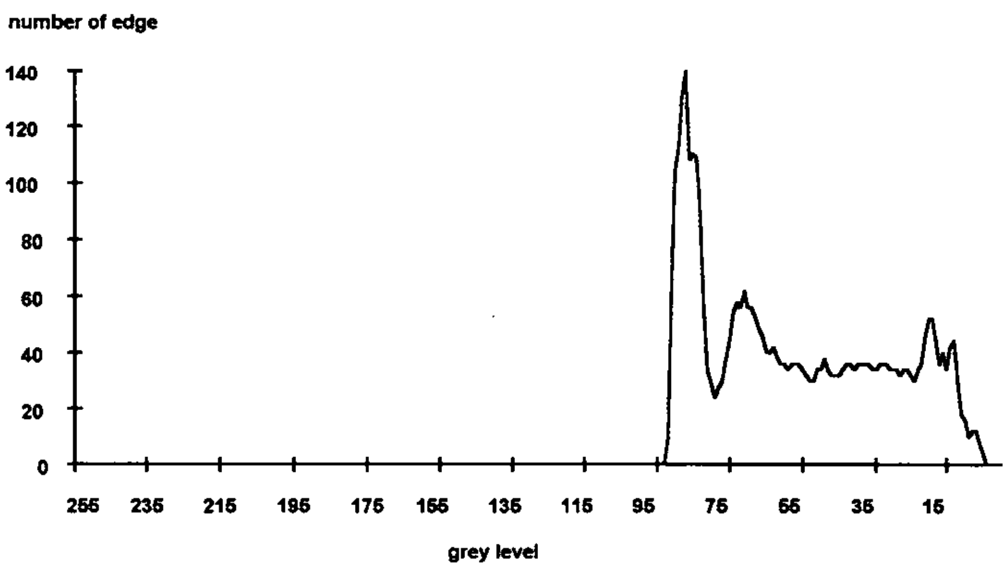
170v position1

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 004 020 070 100 112 136 140 114 116
115 089 051 026 018 030 036 038 034 044 046 046 048 050 052 054 062 058 060
052 046 034 032 028 030 028 028 028 026 028 028 030 028 028 028 032 030 028
028 028 030 028 030 032 032 032 028 026 024 026 028 032 030 028 026 024 022
024 028 030 030 032 036 034 044 056 056 052 046 050 058 054 038 036 028 024
022 008 006 000 000 000 000 000 000



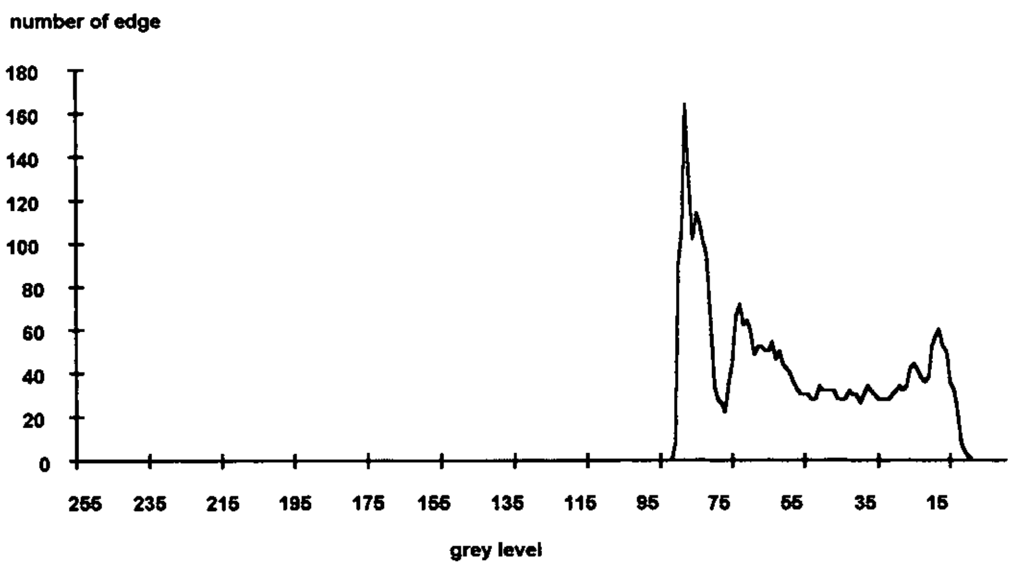
170v position2

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 010 066 104 112 130 140 108 110
109 093 059 034 030 024 028 030 038 044 054 058 056 062 056 056 052 048 046
040 040 042 038 036 036 034 036 036 036 034 032 030 030 034 034 038 034 032
032 032 034 036 036 034 036 036 036 036 034 034 036 036 036 034 034 034 032
034 034 032 030 034 036 046 052 052 044 036 040 034 042 044 028 018 016 010
012 012 008 004 000 000 000 000 000



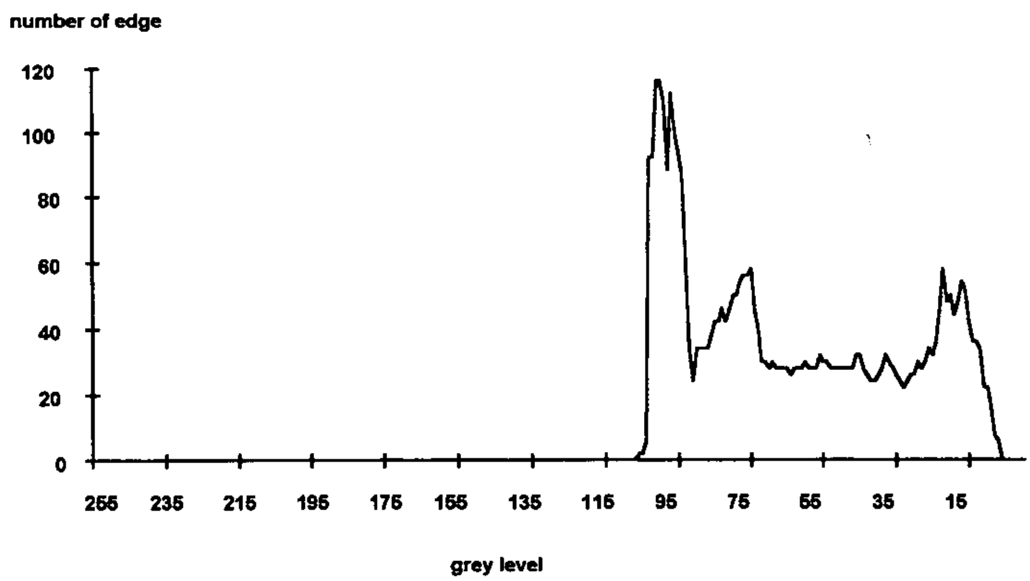
170v position3

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 008 090 106 164 128 102 114
110 102 095 065 034 028 026 022 036 044 066 072 062 064 060 048 052 052 050
050 054 046 050 044 042 040 036 032 030 030 030 028 028 034 032 032 032 032
028 028 028 032 030 030 026 030 034 032 030 028 028 028 028 030 032 034 032
034 042 044 042 038 036 038 052 056 060 052 050 036 032 024 008 004 002 000
000 000 000 000 000 000 000 000 000



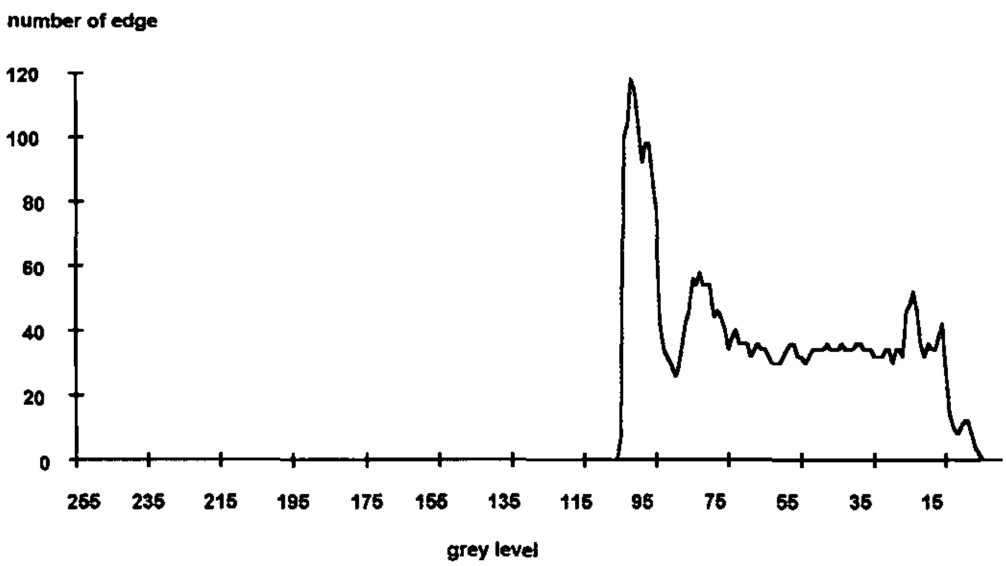
185v position1

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 002 002 006
092 092 116 116 110 088 112 100 093 087 059 034 024 034 034 034 034 038 042
042 046 042 046 050 050 054 056 056 058 046 040 030 030 028 030 028 028 028
028 026 028 028 028 030 028 028 028 032 030 030 028 028 028 028 028 028 028
032 032 028 026 024 024 026 028 032 030 028 026 024 022 024 026 026 030 028
030 034 032 036 046 058 048 050 044 048 054 052 042 036 036 034 022 022 016
008 006 000 000 000 000 000 000 000



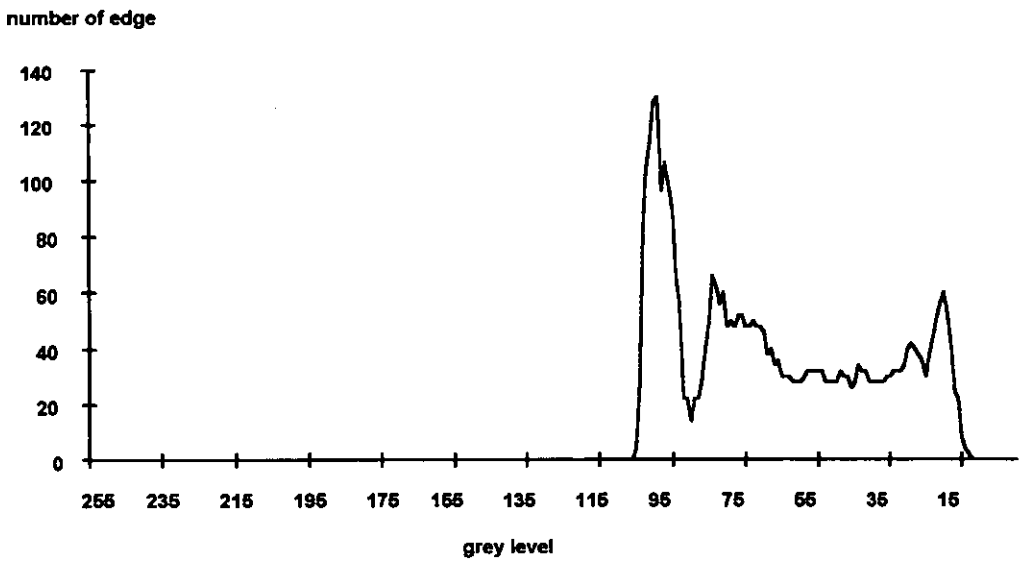
185v position2

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 008 100
104 118 114 104 092 098 098 085 077 042 034 032 030 026 028 034 042 046 056
054 058 054 054 054 044 046 044 040 034 038 040 036 036 036 032 034 036 034
034 032 030 030 030 032 034 036 036 032 032 030 032 034 034 034 034 036 034
034 034 036 034 034 034 036 036 034 034 034 032 032 032 034 034 030 034 034
032 046 048 052 046 036 032 036 034 034 038 042 028 014 010 008 010 012 012
008 004 002 000 000 000 000 000 000



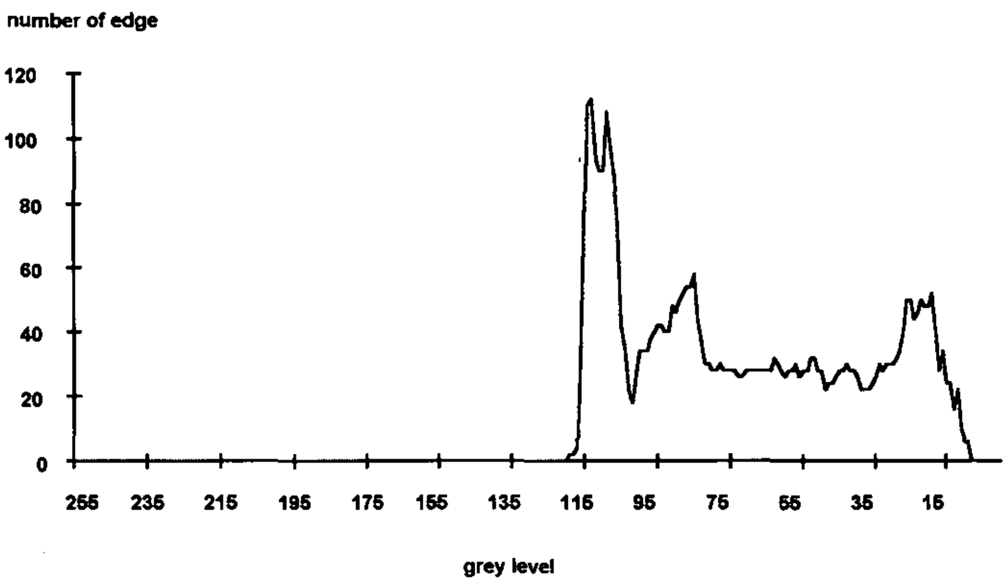
185v position3

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 004 028
082 104 114 128 130 096 106 098 089 065 056 022 022 014 022 022 028 040 050
066 062 056 060 048 050 048 052 052 048 048 050 048 048 046 038 040 034 036
030 030 030 028 028 028 030 032 032 032 032 032 028 028 028 028 032 030 030
026 028 034 032 032 028 028 028 028 028 030 030 032 032 032 034 040 042 040
038 036 030 038 044 050 056 060 054 042 024 022 008 004 002 000 000 000 000
000 000 000 000 000 000 000 000 000



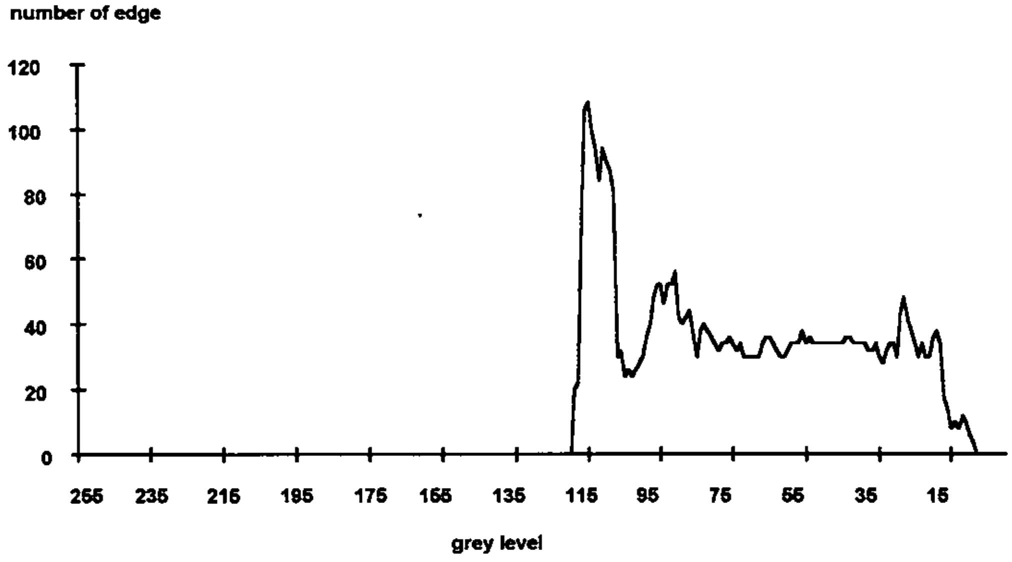
200v position1

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 002 002 004 032 074 110 112 094 090 000 108 098 088 077 042 034
022 018 026 034 034 034 038 040 042 042 040 040 048 046 050 052 054 054 058
044 038 030 030 028 028 030 028 028 028 028 026 026 028 028 028 028 028 028
028 028 032 030 028 026 028 028 030 026 028 028 032 032 028 028 022 024 024
026 028 028 030 028 028 026 022 022 022 024 026 030 028 030 030 030 032 034
040 050 050 044 046 050 048 048 052 040 028 034 024 024 016 022 010 006 006
000 000 000 000 000 000 000 000 000



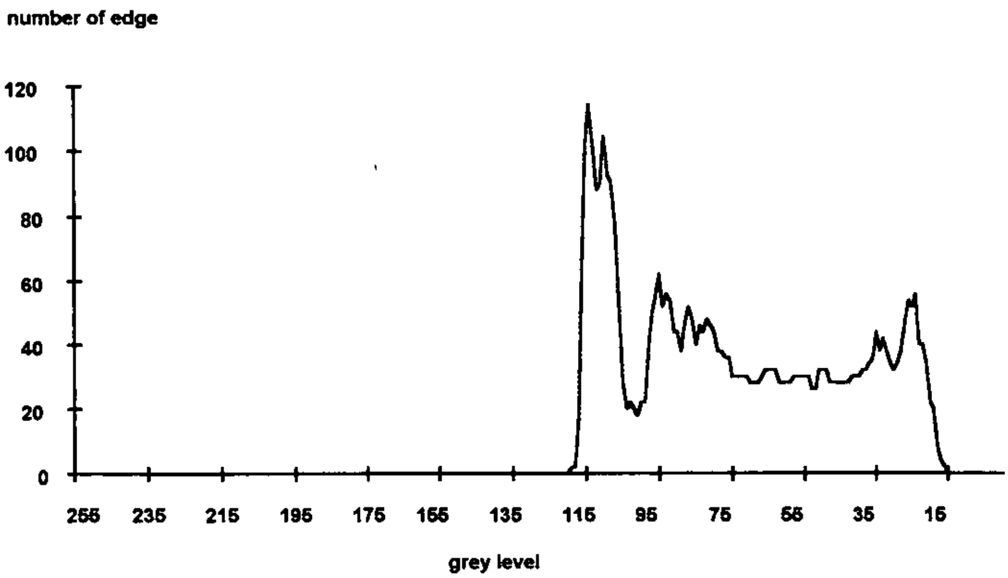
200v position2

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 020 022 072 106 108 100 094 084 094 000 087 081 030 032 024 026
024 026 028 030 036 040 048 052 052 046 052 052 056 042 040 042 044 036 030
038 040 038 036 034 032 034 034 036 034 032 034 030 030 030 030 030 034 036
036 034 032 030 030 032 034 034 034 038 034 036 034 034 034 034 034 034 034
034 034 036 036 034 034 034 034 032 032 034 030 028 032 034 034 030 044 048
042 038 034 030 034 030 030 036 038 034 018 014 008 010 008 012 010 006 004
000 000 000 000 000 000 000 000 000

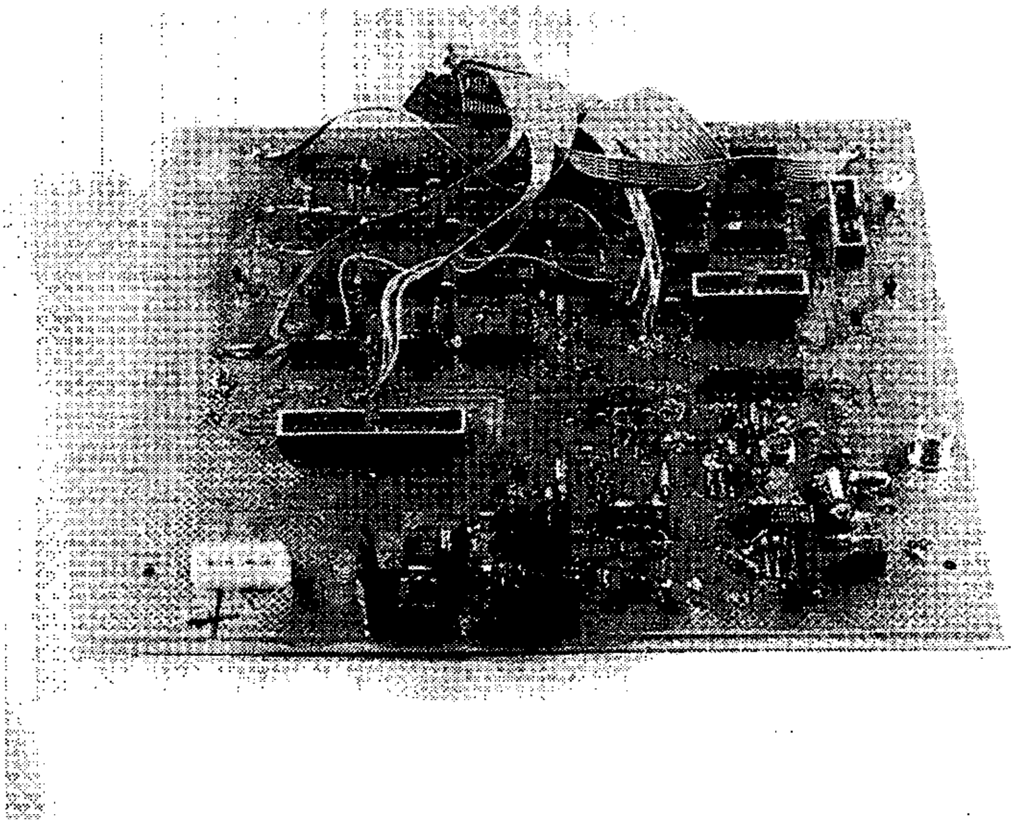


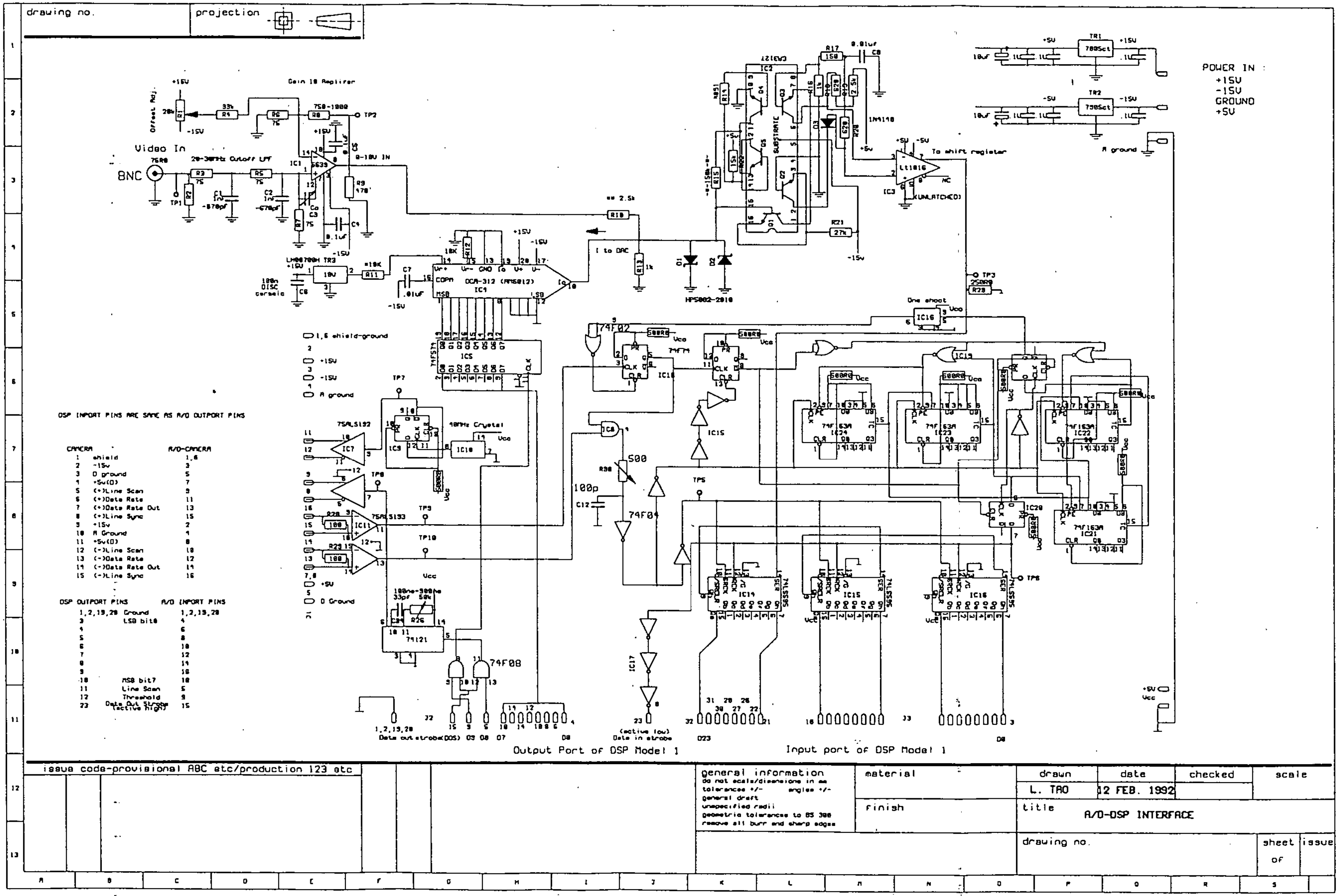
200v position3

000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 002 000 018 064 102 114 102 088 000 104 092 090 077 054 028 020
022 020 018 022 022 040 050 056 062 052 056 054 044 044 038 046 052 048 040
046 044 048 046 044 038 038 036 036 030 030 030 030 030 028 028 028 030 032
032 032 032 028 028 028 028 030 030 030 030 030 026 026 032 032 032 028 028
028 028 028 028 030 030 030 032 032 034 036 044 038 042 038 034 032 034 038
046 054 052 056 040 040 034 022 020 008 004 002 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000

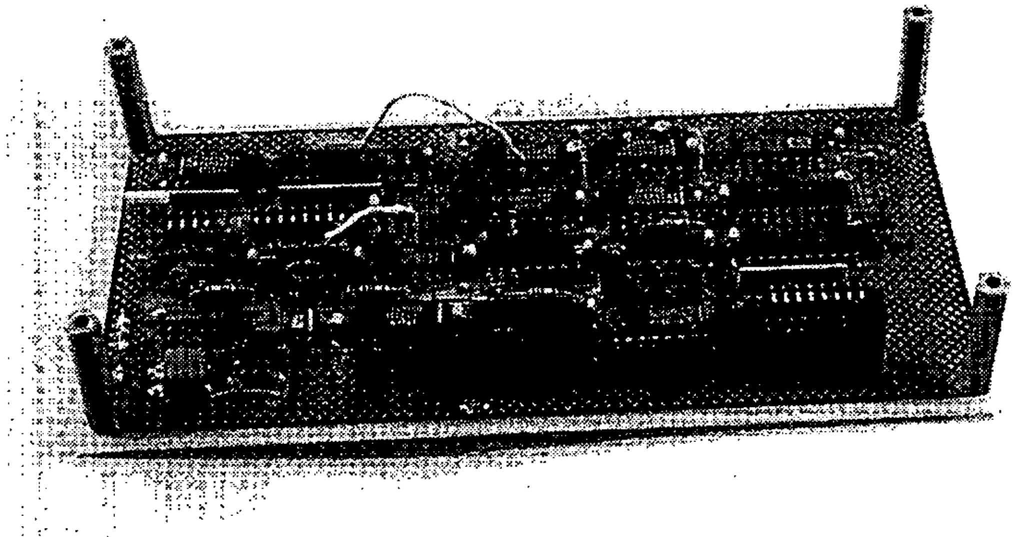


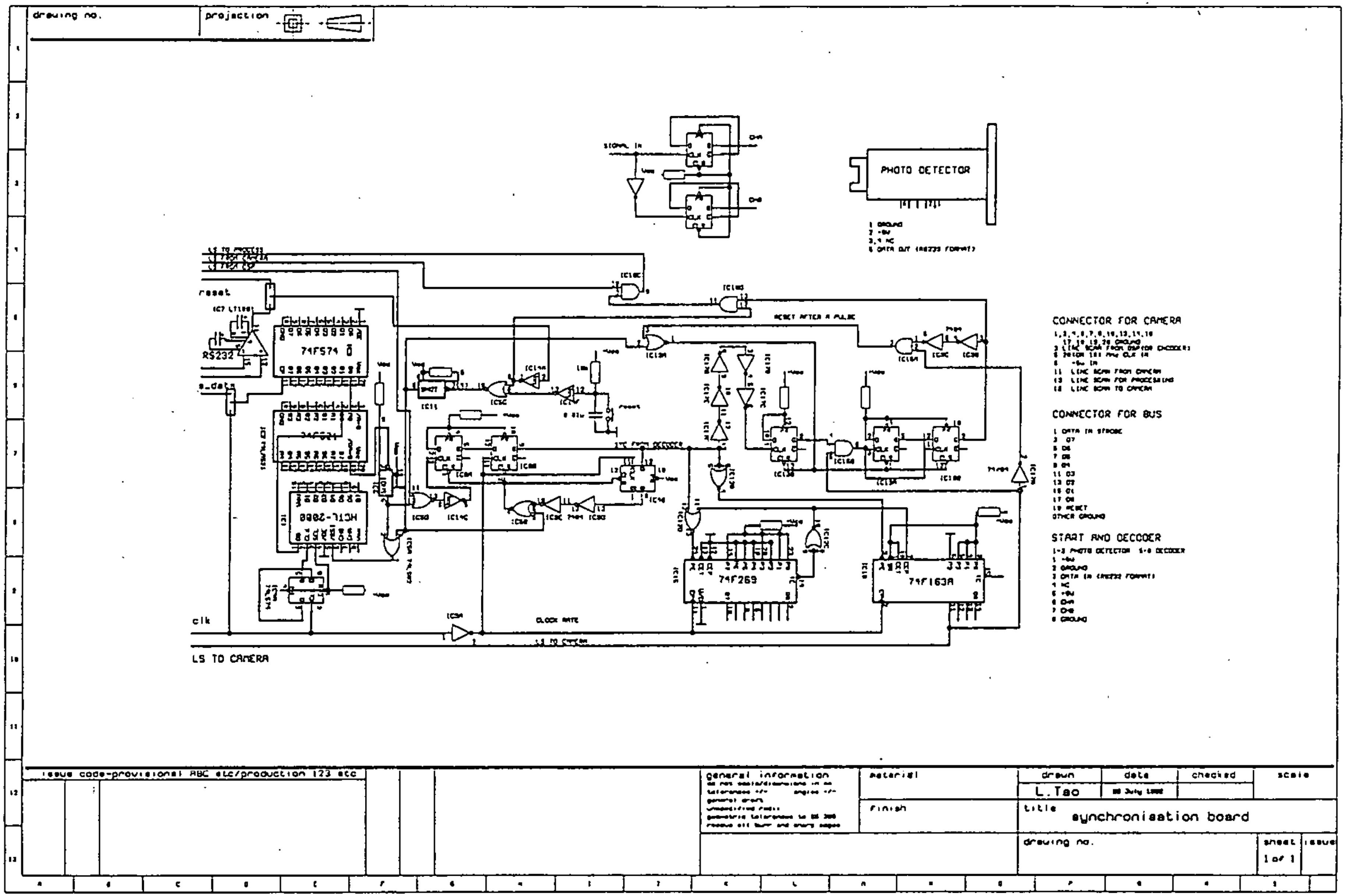
Appendix 4 Interface between Camera and DSP Board




drawing no.
projection
Video In
BNC
Gain 10 Amplifier
Offset Adj.
20-30MHz Cutoff LPF
I to DAC
To shift register
POWER IN :
+15V
-15V
GROUND
+5V
One shoot
40MHz Crystal
DSP INPORT PINS ARE SAME AS A/D OUTPORT PINS
CAMERA
A/D-CAMERA
shield
Line Scan
Data Rate
Data Rate Out
Line Sync
A Ground
DSP OUTPORT PINS
A/D INPORT PINS
LSB bit0
MSB bit7
Line Scan
Threshold
Data Out Strobe (active high)
Data out strobe(DOS)
(active low) Data in strobe
Output Port of DSP Model 1
Input port of DSP Model 1
issue code-provisional ABC etc/production 123 etc
general information
do not scale/dimensions in mm
tolerances +/- angles +/-
general draft
unspecified radii
geometric tolerances to BS 308
remove all burr and sharp edges
material
finish
drawn
L. TAO
date
12 FEB. 1992
checked
scale
title
A/D-DSP INTERFACE
drawing no.
sheet
of
issue

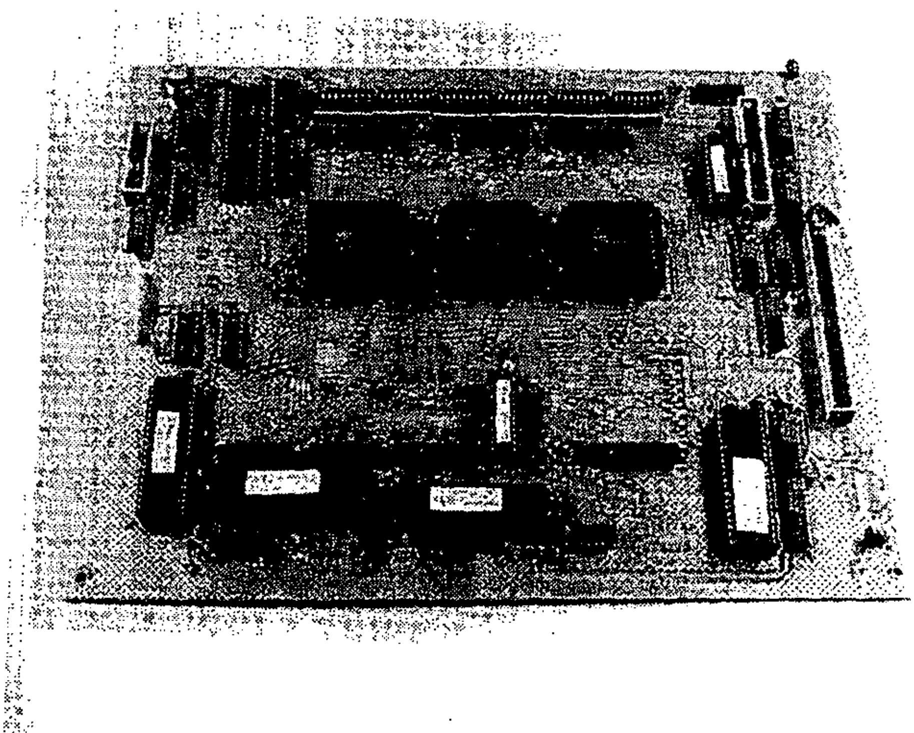
Appendix 5 Synchronisation Board

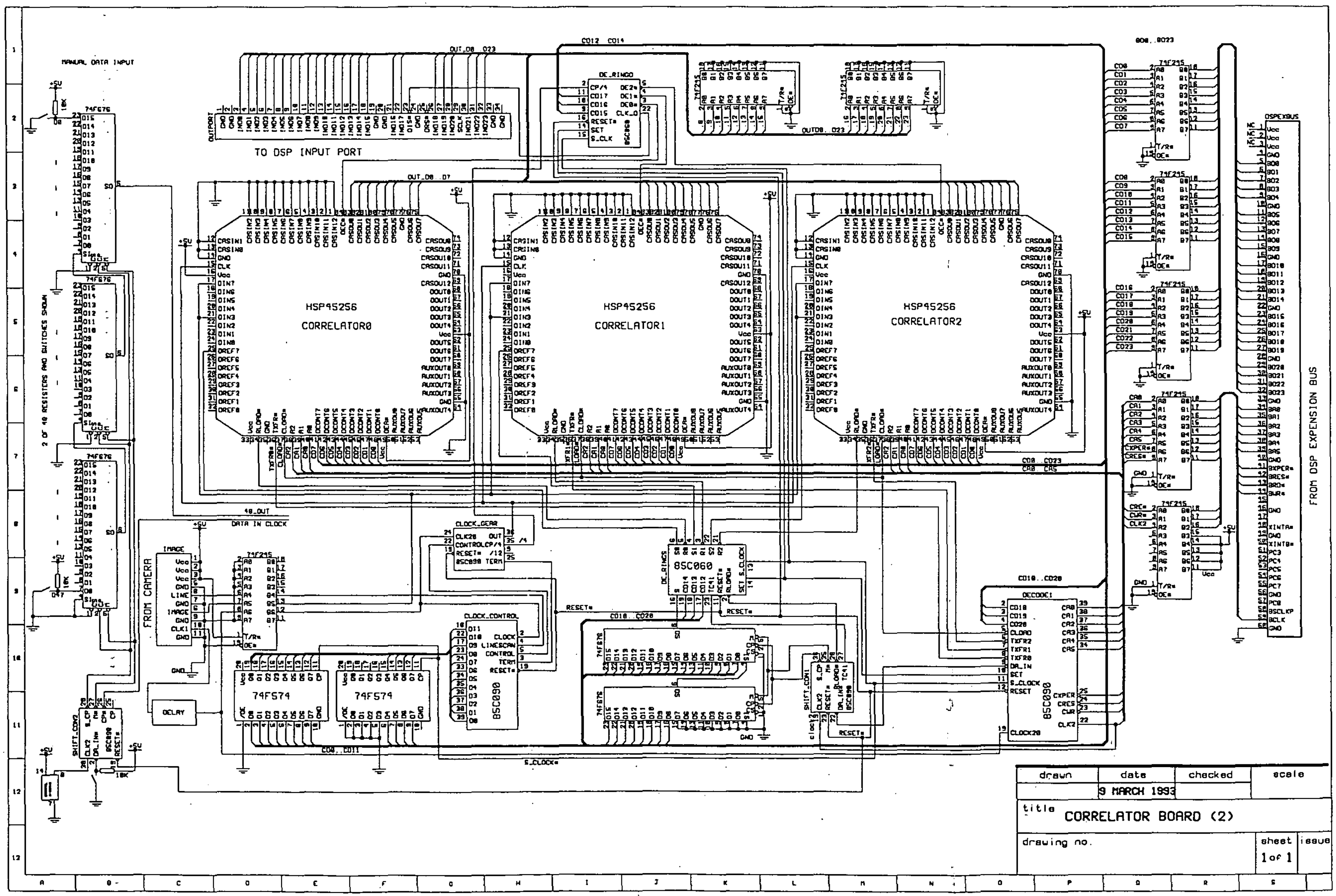




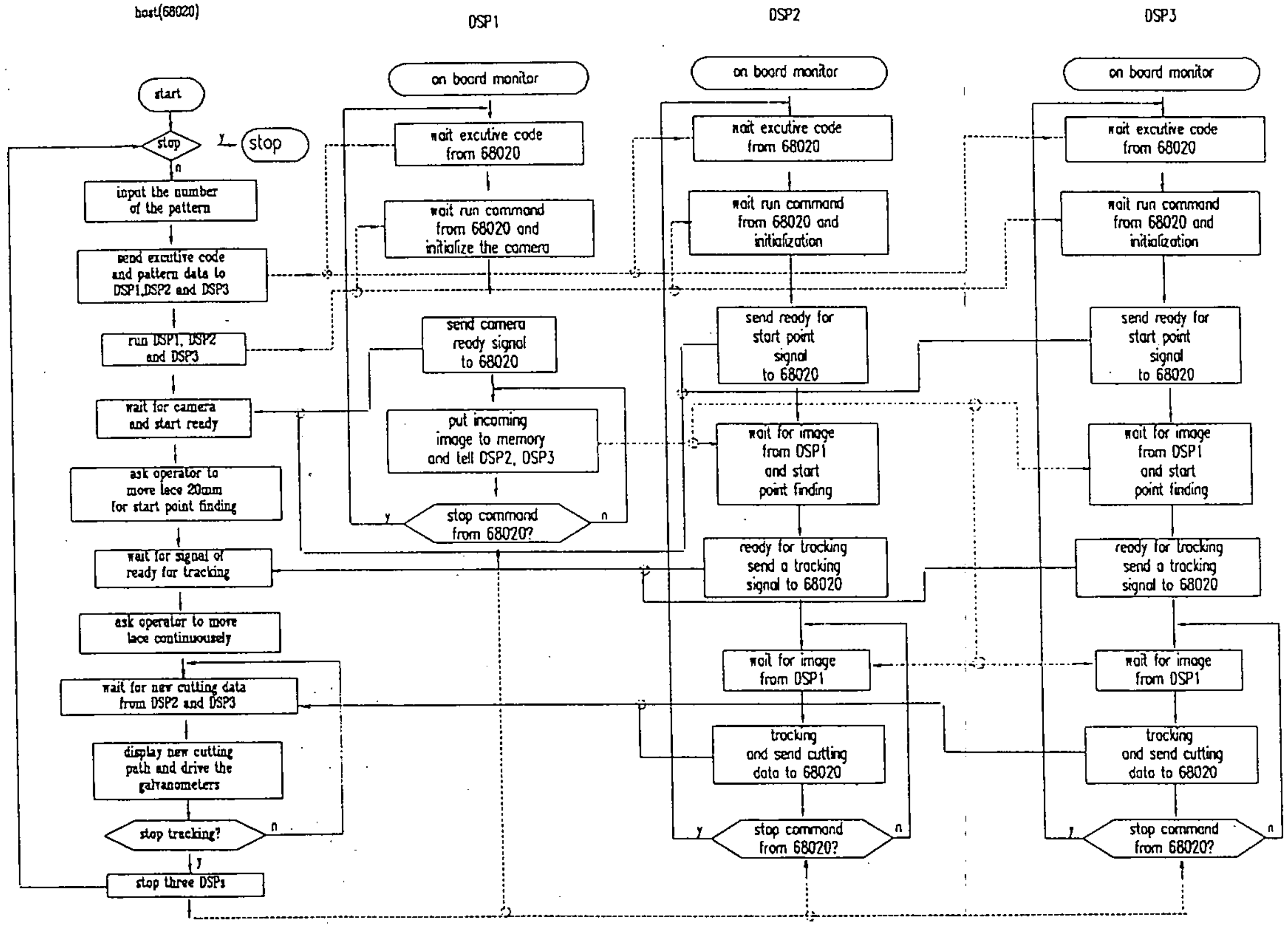
drawing no.
projection
SIGNAL IN
PHOTO DETECTOR
reset
RS232
74F574
74F521
HCTL-2000
s_data
clk
CLOCK RATE
LS TO CAMERA
RESET AFTER A PULSE
74F269
74F163A
CONNECTOR FOR CAMERA
CONNECTOR FOR BUS
START AND DECODER
general information
material
finish
drawn
date
checked
scale
L.Tao
title
synchronisation board
drawing no.
sheet
issue
1 of 1

Appendix 6 Correlator Board




MANUAL DATA INPUT
2 OF 48 RESISTERS AND SWITCHES SHOWN
TO DSP INPUT PORT
HSP45256
CORRELATOR0
HSP45256
CORRELATOR1
HSP45256
CORRELATOR2
FROM CAMERA
DATA IN CLOCK
CLOCK_GEAR
CLOCK_CONTROL
DELAY
74F574
74F574
85C060
DE_RINGS
DECODE1
RESET*
S_CLOCK*
FROM DSP EXPENSION BUS
drawn
date
checked
scale
9 MARCH 1993
title
CORRELATOR BOARD (2)
drawing no.
sheet
issue
1 of 1

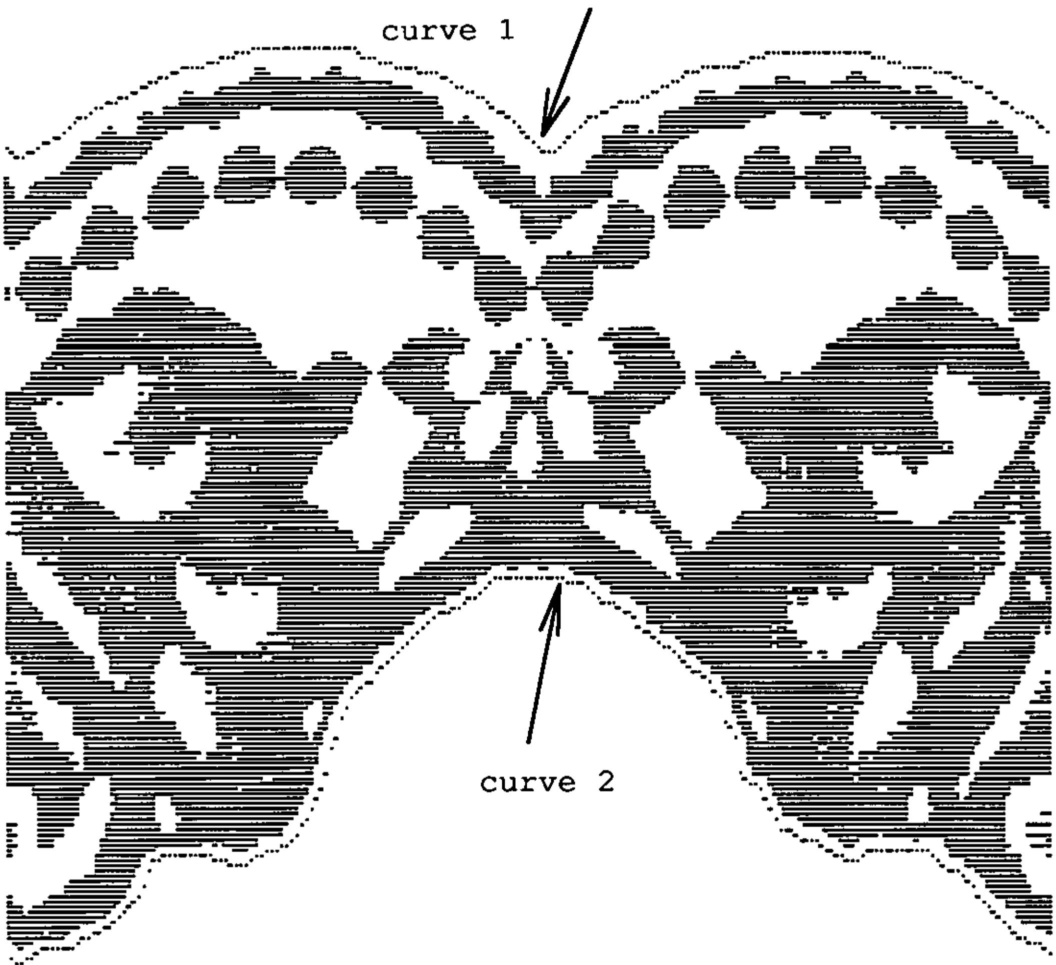
Appendix 7 Flowchart for Vision System



inside chip instruction flowchart
host to DSP flowchart
DSP to host flowchart
inter-DSP flowchart

Appendix 8 Tracking Accuracy

Map:



curve 1

(x,y)	D(x)	map	Amap	σ^2
0000 0381	5	ffffff ffffff ffffff fc000f f1ffff 01d005	5.000000	0.000000
0001 0382	6	ffffff ffffff ffffff fe000f fffffe 02002a	5.500000	0.250000
0002 0383	8	ffffff ffffff ffffff ff803f cffffc 000055	6.333333	1.555555
0003 0383	8	ffffff ffffff ffffff ff807e 07fff8 000014	6.750000	1.687500
0004 0382	8	ffffff ffffff ffffff ff803c 00fff8 008208	7.000000	1.599998
0005 0382	6	ffffff ffffff ffffff fe00f8 00ffe0 202800	6.833333	1.472218
0006 0381	6	ffffff ffffff ffffff fe00fc 001ff0 5e0054	6.714286	1.346935
0007 0380	7	ffffff ffffff ffffff ff00fe 000fe0 0f8000	6.750000	1.187500
0008 0380	6	ffffff ffffff ffffff fe00fe 000fc0 1fe000	6.666667	1.111115
0009 0379	7	ffffff ffffff ffffff ff00ff 0007c0 1ffc00	6.700000	1.010004
0010 0379	7	ffffff ffffff ffffff ff01ff 000f04 17ff00	6.727273	0.925624
0011 0378	7	ffffff ffffff ffffff ff00ff e00e00 0fffe0	6.750000	0.854168
0012 0377	7	ffffff ffffff ffffff ff007f f01f00 07fffc	6.769231	0.792901
0013 0376	7	ffffff ffffff ffffff ff00ff e3ff80 03ffff	6.785714	0.739801
0014 0375	7	ffffff ffffff ffffff ff007f c3ff00 01ffff	6.800000	0.693338
0015 0375	4	ffffff ffffff ffffff f8007c 03ff00 01ffff	6.625000	1.109381

0016 0374	5	ffffff ffffff ffffff fc00f8 00ff80 00ffff	6.529411	1.190316
0017 0373	6	ffffff ffffff ffffff fe01f0 007fc0 01ffff	6.500000	1.138896
0018 0373	5	ffffff ffffff ffffff fc03f0 007fc0 01ffff	6.421052	1.191146
0019 0372	5	ffffff ffffff ffffff fc03f8 003fe0 03fffb	6.349999	1.227510
0020 0372	4	ffffff ffffff ffffff f803f8 003fe0 03fffb	6.238095	1.419509
0021 0372	5	ffffff ffffff ffffff fc07fc 003f80 03ffeb	6.181818	1.421494
0022 0371	7	ffffff ffffff ffffff ff03ff 007f80 01fff7	6.217391	1.387533
0023 0371	7	ffffff ffffff ffffff ff03ff 007f00 07ffd7	6.250000	1.354178
0024 0370	8	ffffff ffffff ffffff ff81ff 803f80 03ff82	6.320000	1.417608
0025 0370	8	ffffff ffffff ffffff ff81ff ebfe00 03fe80	6.384615	1.467466
0026 0369	5	ffffff ffffff ffffff fd01ff ffff00 405400	6.333333	1.481491
0027 0369	4	ffffff ffffff ffffff f803ff dfff80 005c00	6.250000	1.616083
0028 0368	5	ffffff ffffff ffffff fc03fe 07ffe0 002820	6.206896	1.612374
0029 0368	6	ffffff ffffff ffffff fe0ff8 03ffc0 002000	6.200000	1.560006
0030 0367	7	ffffff ffffff ffffff ff07f0 01ffc0 000000	6.225806	1.529662
0031 0367	5	ffffff ffffff ffffff fc0fe0 01ffc0 014000	6.187500	1.527348
0032 0366	6	ffffff ffffff ffffff fe0fe0 00fff0 000a00	6.181818	1.482098
0033 0366	6	ffffff ffffff ffffff fe0fe0 00fff0 002020	6.176470	1.439453
0034 0365	5	ffffff ffffff ffffff fc07f0 007ffc 000014	6.142857	1.436739
0035 0365	4	ffffff ffffff ffffff f80ff0 00fffc 000000	6.083333	1.520836
0036 0365	4	ffffff ffffff ffffff f80ffc 01fffe 000000	6.027027	1.593866
0037 0365	5	ffffff ffffff ffffff fc0ffc 01ffff 000000	6.000000	1.578949
0038 0365	5	ffffff ffffff ffffff fc0ffe 07ffff 000001	5.974359	1.563447
0039 0364	6	ffffff ffffff ffffff fe07ff efffff 800000	5.975000	1.524378
0040 0364	6	ffffff ffffff ffffff fe03ff ffffff 800000	5.975610	1.487213
0041 0364	5	ffffff ffffff ffffff fc03ff 8fffff c00000	5.952381	1.473925
0042 0363	5	ffffff ffffff ffffff fc07fc 07ffff e00000	5.930233	1.460253
0043 0363	5	ffffff ffffff ffffff fc07f0 03ffff f00000	5.909091	1.446283
0044 0363	5	ffffff ffffff ffffff fc07f0 03ffff f80000	5.888889	1.432103
0045 0362	6	ffffff ffffff ffffff fe03f0 01ffff fc0000	5.891304	1.401231
0046 0362	5	ffffff ffffff ffffff fc0fe0 01ffff fe0000	5.872341	1.387960
0047 0362	5	ffffff ffffff ffffff fc0fe0 01ffff fe0000	5.854167	1.374565
0048 0361	5	ffffff ffffff ffffff fc07f0 00ffff ff8000	5.836735	1.361101
0049 0361	3	ffffff ffffff ffffff f007f0 01ffff ffe000	5.780000	1.491599
0050 0361	3	ffffff ffffff ffffff f00ffc 01ffff fff800	5.725491	1.610916
0051 0361	4	ffffff ffffff ffffff f80ffc 01ffff fffc00	5.692308	1.636094
0052 0361	5	ffffff ffffff ffffff fc07fc 07ffff fffe00	5.679245	1.614098
0053 0361	5	ffffff ffffff ffffff fc07ff dfffff ffff00	5.666667	1.592590
0054 0361	5	ffffff ffffff ffffff fc07ff dfffff ffff00	5.654546	1.571568
0055 0361	5	ffffff ffffff ffffff fc07ff 07ffff ffff00	5.642858	1.551019
0056 0361	5	ffffff ffffff ffffff fc07fe 01ffff ffff00	5.631579	1.530930
0057 0361	5	ffffff ffffff ffffff fc07fc 01ffff ffffc0	5.620690	1.511296
0058 0361	6	ffffff ffffff ffffff fe07f8 01ffff ffff80	5.627119	1.488077
0059 0361	6	ffffff ffffff ffffff fe07f8 00ffff ffff00	5.633334	1.465556
0060 0361	5	ffffff ffffff ffffff fc07f0 00ffff fffe00	5.622951	1.448000
0061 0361	5	ffffff ffffff ffffff fc07f0 01ffff fffc00	5.612904	1.430803
0062 0361	5	ffffff ffffff ffffff fc07f0 01ffff fffc00	5.603175	1.413956
0063 0361	4	ffffff ffffff ffffff f807f0 01ffff fffc00	5.578125	1.431395
0064 0361	3	ffffff ffffff ffffff f007f8 01ffff fff000	5.538462	1.510057
0065 0361	3	ffffff ffffff ffffff f007fc 03ffff fff000	5.500000	1.583333
0066 0361	5	ffffff ffffff ffffff fc07ff 07ffff fff000	5.492538	1.563375
0067 0361	5	ffffff ffffff ffffff fc07ff ffffff fff800	5.485295	1.543899
0068 0361	5	ffffff ffffff ffffff fc07ff f7ffff fffc01	5.478261	1.524889
0069 0362	4	ffffff ffffff ffffff f807ff c0ffff fff80f	5.457143	1.533878

0070 0363	4	ffffff ffffff ffffff f807ff 007fff fff87f	5.436620	1.541757
0071 0363	4	ffffff ffffff ffffff f801fe 007fff fffdff	5.416667	1.548612
0072 0363	4	ffffff ffffff ffffff f803fe 007fff ffffff	5.397261	1.554514
0073 0364	3	ffffff ffffff ffffff f007f8 007fff fffbff	5.364865	1.610117
0074 0364	3	ffffff ffffff ffffff f003f8 007fff fff0ff	5.333334	1.662221
0075 0364	3	ffffff ffffff ffffff f003f8 00ffff ffe07f	5.302632	1.711043
0076 0365	3	ffffff ffffff ffffff f007f0 01ffff ff007f	5.272728	1.756787
0077 0365	3	ffffff ffffff ffffff f007fc 01ffff ff003f	5.243590	1.799636
0078 0365	3	ffffff ffffff ffffff f007fc 01ffff fe001f	5.215190	1.839768
0079 0365	3	ffffff ffffff ffffff f007fc 07ffff fc001f	5.187500	1.877343
0080 0365	4	ffffff ffffff ffffff f807ff d7ffff fc0007	5.172840	1.871358
0081 0366	3	ffffff ffffff ffffff f007ff fbbfff f8000f	5.146342	1.905409
0082 0366	4	ffffff ffffff ffffff f803ff f803ff f8000f	5.132531	1.898095
0083 0366	4	ffffff ffffff ffffff f8003f f003ff e00002	5.119048	1.890588
0084 0366	4	ffffff ffffff ffffff f8001f f003ff e00000	5.105883	1.882906
0085 0367	5	ffffff ffffff ffffff fc003f e003ff c00000	5.104651	1.861141
0086 0367	8	ffffff ffffff ffffff ff803f c001ff e05400	5.137931	1.934997
0087 0368	7	ffffff ffffff ffffff ff007f 8003ff 83f802	5.159091	1.951964
0088 0369	7	ffffff ffffff ffffff ff00ff 000fff 87fc00	5.179775	1.967683
0089 0369	6	ffffff ffffff ffffff fe00ff 801fff c7fc1d	5.188889	1.953210
0090 0370	4	ffffff ffffff ffffff f800ff 803fff bff9ff	5.175824	1.947109
0091 0370	5	ffffff ffffff ffffff fc00ff e83fff fffbff	5.173913	1.926278
0092 0371	5	ffffff ffffff ffffff fc007f fc007f fff7ff	5.172043	1.905889
0093 0371	6	ffffff ffffff ffffff fe001f ff001f ffdfff	5.180851	1.892828
0094 0372	5	ffffff ffffff ffffff fc001f fe000f fe3ffe	5.178947	1.873242
0095 0373	5	ffffff ffffff ffffff fd001f fc001f 707ff0	5.177083	1.854058
0096 0374	8	ffffff ffffff ffffff ff803f f8001e 01ffc0	5.206186	1.916250
0097 0374	8	ffffff ffffff ffffff ff801f f8003f 00fe00	5.234694	1.975531
0098 0375	9	ffffff ffffff ffffff ffc03f f8007c 01c004	5.272727	2.097335
0099 0376	8	ffffff ffffff ffffff ff803f f803f8 f800f8	5.300000	2.150000
0100 0377	7	ffffff ffffff ffffff ff007f f007e7 f03ffc	5.316832	2.157042
0101 0378	8	ffffff ffffff ffffff ff800f f80fff e0fffa	5.343138	2.205783
0102 0379	7	ffffff ffffff ffffff ff0007 fdff7f e3fff0	5.359224	2.210761
0103 0380	6	ffffff ffffff ffffff fe0007 fbffff e3ffa0	5.365385	2.193415
0104 0381	9	ffffff ffffff ffffff ffc01f c7ffff 805d00	5.400001	2.297139
0105 0382	9	ffffff ffffff ffffff ffc03e 03fbfe 000000	5.433963	2.396580
0106 0382	8	ffffff ffffff ffffff ff8038 00fc28 f80000	5.457944	2.435147
0107 0382	4	ffffff ffffff ffffff f80078 003e01 ff8000	5.444445	2.432095
0108 0381	5	ffffff ffffff ffffff fc007c 001f81 fff000	5.440368	2.411577
0109 0380	6	ffffff ffffff ffffff fe00fe 0007e0 7ffe00	5.445455	2.392479
0110 0379	7	ffffff ffffff ffffff ff007e 0003df 1fff00	5.459460	2.392503
0111 0378	8	ffffff ffffff ffffff ff807f 8003ff efff80	5.482143	2.428251
0112 0377	9	ffffff ffffff ffffff ffc07f c001ff f7ffc0	5.513275	2.515313
0113 0376	9	ffffff ffffff ffffff ffc03f e003ff fbffe0	5.543860	2.598956
0114 0375	8	ffffff ffffff ffffff ff801f f007ff fe7ff0	5.565217	2.628360
0115 0375	7	ffffff ffffff ffffff ff003f c1ffff fe0150	5.577586	2.623298
0116 0374	8	ffffff ffffff ffffff ff803f 81fffb ff0080	5.598290	2.650603
0117 0374	4	ffffff ffffff ffffff f8003e 00fff3 fe0000	5.584745	2.649608
0118 0373	5	ffffff ffffff ffffff fc0010 007ff1 7f0000	5.579832	2.630193
0119 0373	5	ffffff ffffff ffffff fc01f0 007ff8 5c0000	5.575000	2.611050
0120 0372	6	ffffff ffffff ffffff fe03f0 003ff8 080000	5.578512	2.590952
0121 0372	4	ffffff ffffff ffffff f803f0 003ff8 000000	5.565574	2.589972
0122 0372	4	ffffff ffffff ffffff f803f8 003ff8 0000e0	5.552845	2.588679
0123 0372	4	ffffff ffffff ffffff f803f8 007ff8 0000f8	5.540322	2.587094

0124 0371	7	ffffff ffffff ffffff ff01fe 007ffc 0001fe	5.552000	2.583306
0125 0371	7	ffffff ffffff ffffff ff01ff 007ffc 0001ff	5.563492	2.579309
0126 0370	7	ffffff ffffff ffffff ff00ff 83fffe 0001ff	5.574803	2.575118
0127 0370	8	ffffff ffffff ffffff ff80ff ffffff 0003ff	5.593750	2.600595
0128 0369	5	ffffff ffffff ffffff fc00ff ffffff c001ff	5.589147	2.583149
0129 0369	4	ffffff ffffff ffffff f803fd 1fffff c003ff	5.576922	2.582556
0130 0368	5	ffffff ffffff ffffff fc03f8 03ffff f803ff	5.572518	2.565364
0131 0368	5	ffffff ffffff ffffff fc07e0 03ffff fe07ff	5.568181	2.548396
0132 0367	5	ffffff ffffff ffffff fc07c0 01ffff ffc7ff	5.563909	2.531644
0133 0366	6	ffffff ffffff ffffff fe03e0 00ffff ffffff	5.567163	2.514160
0134 0366	6	ffffff ffffff ffffff fe07e0 00ffff ffffff	5.570370	2.496913
0135 0366	4	ffffff ffffff ffffff f807e0 00ffff ffe3ff	5.558823	2.496550
0136 0365	4	ffffff ffffff ffffff f807f0 007fff fff07f	5.547445	2.495933
0137 0365	4	ffffff ffffff ffffff f807f0 00ffff fff07f	5.536232	2.495072
0138 0365	4	ffffff ffffff ffffff f807f8 01ffff ffe01f	5.525179	2.493980
0139 0365	5	ffffff ffffff ffffff fc07fc 01ffff ffc001	5.521428	2.478122
0140 0365	5	ffffff ffffff ffffff fc07ff 07ffff ffc000	5.517730	2.462459
0141 0365	5	ffffff ffffff ffffff fc07ff dfffff ffc000	5.514084	2.446993
0142 0365	5	ffffff ffffff ffffff fc07ff ffffff ff8000	5.510489	2.431717
0143 0365	4	ffffff ffffff ffffff f807fc 1fffff ff0000	5.500000	2.430565
0144 0364	4	ffffff ffffff ffffff f803f8 07ffff ffc000	5.489655	2.429213
0145 0364	4	ffffff ffffff ffffff f807e0 03ffff ffe000	5.479452	2.427665
0146 0363	5	ffffff ffffff ffffff fc07f0 01ffff fff000	5.476191	2.412704
0147 0363	5	ffffff ffffff ffffff fc07e0 01ffff fffc00	5.472973	2.397924
0148 0363	5	ffffff ffffff ffffff fc07c0 01ffff fffe00	5.469799	2.383321
0149 0362	6	ffffff ffffff ffffff fe07e0 00ffff ffff80	5.473333	2.369293
0150 0362	6	ffffff ffffff ffffff fe07e0 00ffff ffff80	5.476821	2.355425
0151 0362	4	ffffff ffffff ffffff f80fe0 03ffff ffff80	5.467105	2.354186
0152 0362	4	ffffff ffffff ffffff f80ff8 03ffff fffe00	5.457517	2.352772
0153 0362	4	ffffff ffffff ffffff f80ff8 03ffff fffe00	5.448052	2.351199
0154 0362	5	ffffff ffffff ffffff fc0ffe 0fffff ffff00	5.445162	2.337317
0155 0362	5	ffffff ffffff ffffff fc0fff bfffff ffff00	5.442308	2.323595
0156 0362	5	ffffff ffffff ffffff fc07ff bfffff fffe00	5.439491	2.310033
0157 0362	6	ffffff ffffff ffffff fe07fe 0fffff fffe00	5.443039	2.297386
0158 0362	6	ffffff ffffff ffffff fe07fe 03ffff fffc00	5.446542	2.284876
0159 0362	6	ffffff ffffff ffffff fe07f8 03ffff fff800	5.450001	2.272500
0160 0362	6	ffffff ffffff ffffff fe03f0 03ffff ffe000	5.453417	2.260252
0161 0362	6	ffffff ffffff ffffff fe07f0 01ffff ffc000	5.456791	2.248133
0162 0362	5	ffffff ffffff ffffff fc0fe0 01ffff ffc000	5.453989	2.235612
0163 0362	5	ffffff ffffff ffffff fc0fe0 01ffff ff8000	5.451221	2.223228
0164 0362	5	ffffff ffffff ffffff fc0fe0 03ffff ff8000	5.448486	2.210980
0165 0362	4	ffffff ffffff ffffff f80fe0 03ffff fe0000	5.439760	2.210223
0166 0362	3	ffffff ffffff ffffff f00ff8 03ffff fe0000	5.425151	2.232419
0167 0362	3	ffffff ffffff ffffff f00ff8 03ffff fc0000	5.410716	2.253929
0168 0362	4	ffffff ffffff ffffff f80ffe 0fffff f80000	5.402368	2.252299
0169 0362	5	ffffff ffffff ffffff fc0fff afffff f80000	5.400001	2.239998
0170 0363	4	ffffff ffffff ffffff f80fff ffffff f80000	5.391814	2.238294
0171 0363	4	ffffff ffffff ffffff f80fff c7ffff f00000	5.383722	2.236477
0172 0363	5	ffffff ffffff ffffff fc07ff c07fff f00000	5.381504	2.224398
0173 0364	4	ffffff ffffff ffffff f80ffe 00ffff 800000	5.373564	2.222520
0174 0364	4	ffffff ffffff ffffff f803fe 00ffff 800000	5.365715	2.220537
0175 0364	4	ffffff ffffff ffffff f807fe 007ffe 000000	5.357955	2.218455
0176 0365	4	ffffff ffffff ffffff f80ff0 007ffc 000000	5.350284	2.216279
0177 0365	4	ffffff ffffff ffffff f807f0 00fffe 000000	5.342698	2.214015

0178 0365	5	ffffff ffffff ffffff fc07f0 00fffe 000000	5.340783	2.202299
0179 0365	3	ffffff ffffff ffffff f00ffc 01fffc 000000	5.327779	2.220335
0180 0365	3	ffffff ffffff ffffff f007fc 01fff8 000000	5.314918	2.237841
0181 0365	3	ffffff ffffff ffffff f007fe 01fff0 000140	5.302199	2.254826
0182 0366	3	ffffff ffffff ffffff f00ffe 0ffff0 000ff8	5.289618	2.271310
0183 0366	4	ffffff ffffff ffffff f80fff affff8 003fff	5.282609	2.267957
0184 0366	4	ffffff ffffff ffffff f80fff fffff8 003fff	5.275676	2.264544
0185 0366	4	ffffff ffffff ffffff f803ff f83ff8 000fff	5.268818	2.261069
0186 0367	4	ffffff ffffff ffffff f801ff f007f0 001fff	5.262033	2.257538
0187 0367	4	ffffff ffffff ffffff f8007f f001f8 001fff	5.255320	2.253956
0188 0368	4	ffffff ffffff ffffff f800ff e003f8 003fff	5.248678	2.250324
0189 0368	8	ffffff ffffff ffffff ff80ff c003f8 002fff	5.263159	2.278112
0190 0369	7	ffffff ffffff ffffff ff00ff 0007f8 001fff	5.272252	2.281897
0191 0369	7	ffffff ffffff ffffff ff00ff 0007fc 0007ff	5.281251	2.285476
0192 0370	6	ffffff ffffff ffffff fe03ff 000ff8 000fff	5.284975	2.276297
0193 0370	5	ffffff ffffff ffffff fc03ff 801ff8 000fff	5.283506	2.264981
0194 0371	4	ffffff ffffff ffffff f803ff c07ff8 001fff	5.276924	2.261773
0195 0372	4	ffffff ffffff ffffff f803ff e3fff8 003fff	5.270409	2.258512
0196 0372	5	ffffff ffffff ffffff fc00ff f82ff8 003fff	5.269036	2.247418
0197 0373	4	ffffff ffffff ffffff f8007f fc0050 007ffc	5.262627	2.244159
0198 0374	4	ffffff ffffff ffffff f800ff f80038 00ffe0	5.256282	2.240851
0199 0374	8	ffffff ffffff ffffff ff80ff f8003e 007f80	5.270000	2.267098
0200 0375	7	ffffff ffffff ffffff ff007f f0007f 01fc00	5.278607	2.270635
0201 0375	7	ffffff ffffff ffffff ff007f f0007f c1f050	5.287129	2.273994
0202 0376	8	ffffff ffffff ffffff ff80ff e000ff 8283f8	5.300493	2.298869
0203 0377	7	ffffff ffffff ffffff ff00ff f007ff c01fff	5.308824	2.301690
0204 0378	6	ffffff ffffff ffffff fe00ff e00fff 80ffff	5.312195	2.292780

curve 1: average offset is 5.312195 pixels from edge.
maximum offset is 9
curve 2

0000 0541	-2	0007d4 4001fc 07f007 ffffff ffffff ffffff	-2.000000	0.000000
0001 0542	-2	0002a2 8003fa bff807 ffffff ffffff ffffff	-2.000000	0.000000
0002 0542	-1	000be0 8003ff fff803 ffffff ffffff ffffff	-1.666667	0.222222
0003 0541	-1	00007d 0001ff ffff03 ffffff ffffff ffffff	-1.500000	0.250000
0004 0541	-1	c0007d 1001ff fffe03 ffffff ffffff ffffff	-1.400000	0.240000
0005 0540	-1	f0003f a8003f fffe03 ffffff ffffff ffffff	-1.333333	0.222222
0006 0539	0	fc0007 fc0000 07ff01 ffffff ffffff ffffff	-1.142857	0.408163
0007 0539	0	ff0001 ff0000 07fe05 ffffff ffffff ffffff	-1.000000	0.500000
0008 0539	0	ff0000 ff0000 07fc01 ffffff ffffff ffffff	-0.888889	0.543210
0009 0539	0	ff0000 ff0000 1ffc01 ffffff ffffff ffffff	-0.800000	0.560000
0010 0539	0	7f8000 7fc000 1ffc01 ffffff ffffff ffffff	-0.727273	0.561984
0011 0539	-1	7fc000 7fc000 7ff003 ffffff ffffff ffffff	-0.750000	0.520833
0012 0539	-2	1ff000 7fd000 7fe007 ffffff ffffff ffffff	-0.846154	0.591716
0013 0538	-1	03fc00 00fa00 fff003 ffffff ffffff ffffff	-0.857143	0.551021
0014 0537	-1	01ff00 007f07 fff803 ffffff ffffff ffffff	-0.866667	0.515556
0015 0537	-1	407f80 00001f fff003 ffffff ffffff ffffff	-0.875000	0.484375
0016 0536	0	003fc0 00001f fff801 ffffff ffffff ffffff	-0.823529	0.498270
0017 0536	-1	003ff8 00003f ff8003 ffffff ffffff ffffff	-0.833333	0.472222
0018 0535	0	01407f 00001f fe0001 ffffff ffffff ffffff	-0.789474	0.481995
0019 0535	-1	04001f 00001f f00003 ffffff ffffff ffffff	-0.800000	0.460000
0020 0534	-1	080000 800000 a00003 ffffff ffffff ffffff	-0.809524	0.439909
0021 0533	-1	000000 700000 000003 ffffff ffffff ffffff	-0.818182	0.421488
0022 0532	-1	002020 380000 000003 ffffff ffffff ffffff	-0.826087	0.404537
0023 0531	-2	005400 100000 000007 ffffff ffffff ffffff	-0.875000	0.442708

0024 0530	-3	002a00	000000	00000f	ffffff	ffffff	ffffff	-0.960000	0.598400
0025 0529	-4	000000	000000	00001f	ffffff	ffffff	ffffff	-1.076923	0.917160
0026 0527	-4	000000	000100	00001f	ffffff	ffffff	ffffff	-1.185185	1.187929
0027 0524	-2	e00002	000000	000007	ffffff	ffffff	ffffff	-1.214286	1.168368
0028 0522	0	f80080	000a80	000001	ffffff	ffffff	ffffff	-1.172414	1.177170
0029 0522	-1	f80000	003fe0	000f83	ffffff	ffffff	ffffff	-1.166667	1.138889
0030 0521	0	f00000	007ff4	0007f1	ffffff	ffffff	ffffff	-1.129032	1.144641
0031 0521	0	f40000	01fffc	0007f1	ffffff	ffffff	ffffff	-1.093750	1.147461
0032 0521	0	f00000	01fffc	0001f1	ffffff	ffffff	ffffff	-1.060606	1.147842
0033 0521	0	f003fc	01ffff	000041	ffffff	ffffff	ffffff	-1.029412	1.146194
0034 0521	0	f005ff	005fff	c00001	ffffff	ffffff	ffffff	-1.000000	1.142857
0035 0521	0	d001ff	e01fff	f00001	ffffff	ffffff	ffffff	-0.972222	1.138117
0036 0521	0	4001ff	f017ff	f00001	ffffff	ffffff	ffffff	-0.945946	1.132213
0037 0521	0	00007f	fc07ff	f40001	ffffff	ffffff	ffffff	-0.921053	1.125346
0038 0521	-1	00007f	fc07ff	fc0003	ffffff	ffffff	ffffff	-0.923077	1.096647
0039 0521	-1	00005f	fe015f	f00003	ffffff	ffffff	ffffff	-0.925000	1.069375
0040 0522	-1	00003f	fe03af	e00003	ffffff	ffffff	ffffff	-0.926829	1.043427
0041 0522	-2	00000f	fe0000	800007	ffffff	ffffff	ffffff	-0.952381	1.045351
0042 0522	-1	00002f	fe0000	000003	ffffff	ffffff	ffffff	-0.953488	1.021093
0043 0523	-2	00007f	fc0000	000007	ffffff	ffffff	ffffff	-0.977273	1.022211
0044 0523	-1	00007f	fc0001	500003	ffffff	ffffff	ffffff	-0.977778	0.999506
0045 0523	-1	0001ff	fd0000	404003	ffffff	ffffff	ffffff	-0.978261	0.977788
0046 0523	-1	00007f	fc0000	000003	ffffff	ffffff	ffffff	-0.978723	0.956994
0047 0522	-1	00023f	fe0000	000003	ffffff	ffffff	ffffff	-0.979167	0.937066
0048 0521	0	000017	ff0000	000001	ffffff	ffffff	ffffff	-0.959184	0.937110
0049 0521	-2	00001f	fd0000	000007	ffffff	ffffff	ffffff	-0.980000	0.939600
0050 0520	-1	00000f	fe2000	002003	ffffff	ffffff	ffffff	-0.980392	0.921184
0051 0520	-1	00802f	faa000	002003	ffffff	ffffff	ffffff	-0.980769	0.903477
0052 0520	-1	00002f	fa8000	000003	ffffff	ffffff	ffffff	-0.981132	0.886437
0053 0519	0	000010	014000	000001	ffffff	ffffff	ffffff	-0.962963	0.887517
0054 0519	-2	000000	014000	000007	ffffff	ffffff	ffffff	-0.981818	0.890579
0055 0518	-2	000000	002000	000007	ffffff	ffffff	ffffff	-1.000000	0.892857
0056 0517	-2	000000	000000	000007	ffffff	ffffff	ffffff	-1.017544	0.894429
0057 0515	-2	000000	000000	000007	ffffff	ffffff	ffffff	-1.034483	0.895363
0058 0514	-3	038000	000000	1e000f	ffffff	ffffff	ffffff	-1.067797	0.944556
0059 0512	-3	03f000	000000	0fe00f	ffffff	ffffff	ffffff	-1.100000	0.990000
0060 0510	-5	03fc00	020000	07e03f	ffffff	ffffff	ffffff	-1.163934	1.219027
0061 0508	-7	03ff80	000000	01e0ff	ffffff	ffffff	ffffff	-1.258065	1.739854
0062 0505	-7	01fff0	000000	0000ff	ffffff	ffffff	ffffff	-1.349206	2.227261
0063 0500	-5	000fff	e00000	00003f	ffffff	ffffff	ffffff	-1.406250	2.397461
0064 0498	-5	0007ff	f80000	00003f	ffffff	ffffff	ffffff	-1.461538	2.556213
0065 0494	-3	0000ff	ffe000	00000f	ffffff	ffffff	ffffff	-1.484848	2.552800
0066 0492	-3	80003f	fff800	00000f	ffffff	ffffff	ffffff	-1.507463	2.548451
0067 0490	-2	f0003f	fffe00	000007	ffffff	ffffff	ffffff	-1.514706	2.514489
0068 0489	-2	fc01ff	ffffc0	000007	ffffff	ffffff	ffffff	-1.521739	2.481411
0069 0488	-1	fe03ff	fffff8	000003	ffffff	ffffff	ffffff	-1.514286	2.449795
0070 0487	-2	ff87ff	fffffc	000007	ffffff	ffffff	ffffff	-1.521127	2.418567
0071 0487	-2	ffdfff	fffffc	000007	ffffff	ffffff	ffffff	-1.527778	2.388116
0072 0485	-2	ffffff	fffffc	000007	ffffff	ffffff	ffffff	-1.534247	2.358415
0073 0484	-3	fffbff	fffffe	0f800f	ffffff	ffffff	ffffff	-1.554054	2.355186
0074 0483	-2	fff87f	fffffc	3fc007	ffffff	ffffff	ffffff	-1.560000	2.326400
0075 0482	-1	fff81f	fffff8	3fe003	ffffff	ffffff	ffffff	-1.552631	2.299861
0076 0481	-2	fff007	fffffc	7fc007	ffffff	ffffff	ffffff	-1.558442	2.272558
0077 0480	-1	fff801	ffffe8	3fc003	ffffff	ffffff	ffffff	-1.551282	2.247369

0078	0479	-2	fff800 7ffc00 7fe007 ffffff ffffff ffffff	-1.556962	2.221438
0079	0478	-1	fff800 3ff800 7ff003 ffffff ffffff ffffff	-1.550000	2.197500
0080	0477	-2	fffc00 07f800 7ff007 ffffff ffffff ffffff	-1.555555	2.172839
0081	0476	-1	fffe00 03f800 7ff003 ffffff ffffff ffffff	-1.548780	2.150059
0082	0475	-1	7fff00 01f000 3ff003 ffffff ffffff ffffff	-1.542169	2.127739
0083	0474	-1	3ffe00 002000 1ff803 ffffff ffffff ffffff	-1.535714	2.105867
0084	0473	0	1fff00 000000 0ffc01 ffffff ffffff ffffff	-1.517647	2.108511
0085	0473	0	0fff00 000000 0ff801 ffffff ffffff ffffff	-1.500000	2.110464
0086	0473	-2	07ff81 500100 0ff007 ffffff ffffff ffffff	-1.505747	2.089046
0087	0472	-1	03ff83 f80280 0ff003 ffffff ffffff ffffff	-1.500000	2.068181
0088	0472	-1	07ffc3 fe0000 0fe003 ffffff ffffff ffffff	-1.494382	2.047720
0089	0471	-1	07fff1 ff0740 07e003 ffffff ffffff ffffff	-1.488889	2.027654
0090	0470	-1	03fffb ff9ff8 03e003 ffffff ffffff ffffff	-1.483516	2.007970
0091	0470	-1	83ffff ffbff8 008003 ffffff ffffff ffffff	-1.478261	1.988657
0092	0469	-2	f001ff ffdffc 000007 ffffff ffffff ffffff	-1.483871	1.970169
0093	0468	-1	fe003f ffbffe 000003 ffffff ffffff ffffff	-1.478723	1.951674
0094	0467	-1	ff0007 ff1fff 000003 ffffff ffffff ffffff	-1.473684	1.933517
0095	0466	-1	ff8003 ee0ffe 000003 ffffff ffffff ffffff	-1.468750	1.915689
0096	0466	-1	ff8001 e01ffc 000003 ffffff ffffff ffffff	-1.463917	1.898182
0097	0466	-1	ff8003 f00fe0 0a0003 ffffff ffffff ffffff	-1.459184	1.880986
0098	0466	0	ffc003 e00e00 200001 ffffff ffffff ffffff	-1.444444	1.883277
0099	0466	0	ffe00f e3e003 e00001 ffffff ffffff ffffff	-1.430000	1.885100
0100	0466	-1	ffe00f cfe07f f80003 ffffff ffffff ffffff	-1.425743	1.868248
0101	0466	-1	0ff80f ffe0ff fa0003 ffffff ffffff ffffff	-1.421569	1.851691
0102	0466	-1	03feff bff1ff f80003 ffffff ffffff ffffff	-1.417476	1.835423
0103	0466	-1	01feff fff8ff e80003 ffffff ffffff ffffff	-1.413461	1.819434
0104	0466	0	03f8ff fff00b a00001 ffffff ffffff ffffff	-1.400000	1.820952
0105	0466	0	03e03f bfe000 000001 ffffff ffffff ffffff	-1.386792	1.822090
0106	0466	-1	03800f c28f80 000003 ffffff ffffff ffffff	-1.383177	1.806446
0107	0466	-1	078003 e01ff8 000003 ffffff ffffff ffffff	-1.379629	1.791066
0108	0466	-1	0f8003 f03ffe 000003 ffffff ffffff ffffff	-1.376147	1.775944
0109	0467	-2	7f0003 f03fff 000007 ffffff ffffff ffffff	-1.381818	1.763305
0110	0467	-1	7e0003 df1fff 000003 ffffff ffffff ffffff	-1.378378	1.748721
0111	0467	0	ff0007 ffdfff 000001 ffffff ffffff ffffff	-1.366071	1.749920
0112	0467	0	ff0007 ffdfff 000001 ffffff ffffff ffffff	-1.353982	1.750802
0113	0468	-1	fe003f ffbffe 00f003 ffffff ffffff ffffff	-1.350877	1.736534
0114	0469	-2	fc01ff ff9ffc 01f007 ffffff ffffff ffffff	-1.356522	1.725066
0115	0470	-2	e0ffff ff00a8 03e007 ffffff ffffff ffffff	-1.362069	1.713733
0116	0471	-2	03fff7 fe0100 03e007 ffffff ffffff ffffff	-1.367521	1.702535
0117	0471	-2	01ffe7 fc0000 03f007 ffffff ffffff ffffff	-1.372881	1.691468
0118	0471	0	01ffc5 fc0000 03f001 ffffff ffffff ffffff	-1.361344	1.692959
0119	0472	-1	03ffc2 e00000 07f003 ffffff ffffff ffffff	-1.358333	1.679930
0120	0473	-2	07ff01 000000 07f007 ffffff ffffff ffffff	-1.363636	1.669421
0121	0473	-2	07ff00 000000 0ff007 ffffff ffffff ffffff	-1.368852	1.659029
0122	0474	-2	0ffe00 003800 1fe007 ffffff ffffff ffffff	-1.373984	1.648754
0123	0476	-3	7ff800 00f800 3fc00f ffffff ffffff ffffff	-1.387097	1.656607
0124	0477	-4	fff000 07f800 7fc01f ffffff ffffff ffffff	-1.408000	1.697536
0125	0477	0	fff000 07ff00 7fc001 ffffff ffffff ffffff	-1.396825	1.699672
0126	0478	-1	ffe000 1ffff8 3fc003 ffffff ffffff ffffff	-1.393701	1.687519
0127	0479	-2	ffe000 7ffffc 3fc007 ffffff ffffff ffffff	-1.398437	1.677185
0128	0480	-1	ffe000 fffffc 3f8003 ffffff ffffff ffffff	-1.395349	1.665405
0129	0480	-1	ffe001 fffffe 0fc003 ffffff ffffff ffffff	-1.392308	1.653787
0130	0481	-2	fff007 ffffff 07c047 ffffff ffffff ffffff	-1.396946	1.643960
0131	0482	-2	fff81f fffffe 028007 ffffff ffffff ffffff	-1.401515	1.634240

0132 0483	-2	fffc7f ffffff 000007 ffffff ffffff ffffff	-1.406015	1.624625
0133 0484	-1	ffffff fffffe 000003 ffffff ffffff ffffff	-1.402985	1.613722
0134 0485	-2	ffffff ffffff 000007 ffffff ffffff ffffff	-1.407407	1.604389
0135 0485	-1	fff1ff ffffff 000003 ffffff ffffff ffffff	-1.404412	1.593804
0136 0486	-1	ffe0ff fffffe 000003 ffffff ffffff ffffff	-1.401460	1.583355
0137 0487	-2	ffc1ff fffff0 000007 ffffff ffffff ffffff	-1.405797	1.574459
0138 0489	-2	fe01ff ffff80 000007 ffffff ffffff ffffff	-1.410072	1.565654
0139 0490	-3	f8003f fffe00 00000f ffffff ffffff ffffff	-1.421428	1.572398
0140 0495	-6	0003ff ffc000 00007f ffffff ffffff ffffff	-1.453901	1.708868
0141 0496	-5	0003ff ff8000 00003f ffffff ffffff ffffff	-1.478873	1.784765
0142 0498	-3	000fff f80000 00000f ffffff ffffff ffffff	-1.489510	1.788352
0143 0505	-8	03fff8 000000 0001ff ffffff ffffff ffffff	-1.534722	2.068238
0144 0506	-7	03ffe0 000000 00a0ff ffffff ffffff ffffff	-1.572414	2.258549
0145 0507	-2	007fc0 000000 01f007 ffffff ffffff ffffff	-1.575342	2.244323
0146 0507	0	001fc0 000000 007c01 ffffff ffffff ffffff	-1.564626	2.245823
0147 0508	-1	001f80 000000 003e03 ffffff ffffff ffffff	-1.560811	2.232788
0148 0510	-3	003c00 000000 00f80f ffffff ffffff ffffff	-1.570470	2.231611
0149 0512	-3	000000 000000 00000f ffffff ffffff ffffff	-1.580000	2.230267
0150 0514	-3	000000 002a00 00000f ffffff ffffff ffffff	-1.589404	2.228762
0151 0516	-3	000000 000000 00000f ffffff ffffff ffffff	-1.598684	2.227104
0152 0516	-1	a00000 000000 000003 ffffff ffffff ffffff	-1.594771	2.214875
0153 0516	-1	000000 ae8000 000003 ffffff ffffff ffffff	-1.590909	2.202775
0154 0517	-2	000000 7f0000 000007 ffffff ffffff ffffff	-1.593548	2.189636
0155 0518	-2	000000 3f8000 000007 ffffff ffffff ffffff	-1.596154	2.176652
0156 0518	-1	800000 3fa000 000003 ffffff ffffff ffffff	-1.592357	2.165037
0157 0518	-1	000000 ffe800 000003 ffffff ffffff ffffff	-1.588607	2.153541
0158 0518	-1	000000 fff800 080003 ffffff ffffff ffffff	-1.584906	2.142162
0159 0519	-2	000007 ffd000 050007 ffffff ffffff ffffff	-1.587500	2.129844
0160 0520	-2	00000f ff8000 080007 ffffff ffffff ffffff	-1.590062	2.117665
0161 0520	-1	000003 ffe00a 000003 ffffff ffffff ffffff	-1.586420	2.106729
0162 0521	-2	000003 ffc010 400007 ffffff ffffff ffffff	-1.588957	2.094848
0163 0521	-1	000007 ffc000 000003 ffffff ffffff ffffff	-1.585366	2.084176
0164 0522	-1	00002f fe0000 000003 ffffff ffffff ffffff	-1.581818	2.073609
0165 0522	-1	00003f fe0002 aa0003 ffffff ffffff ffffff	-1.578313	2.063144
0166 0522	-2	00003f fe002f fa8007 ffffff ffffff ffffff	-1.580838	2.051848
0167 0521	-2	00001f fe005f f00007 ffffff ffffff ffffff	-1.583333	2.040674
0168 0520	-1	80000f fe00ff f80003 ffffff ffffff ffffff	-1.579882	2.030601
0169 0520	-1	80003f fe03ff fa0003 ffffff ffffff ffffff	-1.576471	2.020623
0170 0520	-1	e800ff f80fff f80003 ffffff ffffff ffffff	-1.573099	2.010738
0171 0520	-2	fa00ff 803fff e00807 ffffff ffffff ffffff	-1.575581	2.000101
0172 0520	-1	fa00fe 00ffff e00803 ffffff ffffff ffffff	-1.572254	1.990444
0173 0520	-1	fa00bc 01ffff a00003 ffffff ffffff ffffff	-1.568966	1.980876
0174 0520	-1	fa0000 01ffff 800003 ffffff ffffff ffffff	-1.565714	1.971396
0175 0520	-1	fe0000 00fffe 800003 ffffff ffffff ffffff	-1.562500	1.962003
0176 0520	-1	fe0000 003ffe 800003 ffffff ffffff ffffff	-1.559322	1.952696
0177 0520	-1	fe0000 003ffa 0003e3 ffffff ffffff ffffff	-1.556180	1.943473
0178 0520	0	ff0000 0003e0 000fe1 ffffff ffffff ffffff	-1.547486	1.946069
0179 0520	-1	ff0000 000000 000fe3 ffffff ffffff ffffff	-1.544445	1.936913
0180 0520	-1	ff0020 020000 000fe3 ffffff ffffff ffffff	-1.541437	1.927841
0181 0521	-2	fe0001 400000 000107 ffffff ffffff ffffff	-1.543956	1.918397
0182 0522	-3	fc0200 03e000 00000f ffffff ffffff ffffff	-1.551913	1.919436
0183 0523	-4	f00010 1fc000 00001f ffffff ffffff ffffff	-1.565217	1.941399
0184 0524	-3	e00000 ff8000 00000f ffffff ffffff ffffff	-1.572973	1.941972
0185 0524	-1	c08003 ff0000 000003 ffffff ffffff ffffff	-1.569893	1.933287

0186 0524	-1	808003 fc0000 008003 ffffff ffffff ffffff	-1.566845	1.924676
0187 0525	-2	00001f f00000 07c007 ffffff ffffff ffffff	-1.569149	1.915431
0188 0526	-1	00003f 800000 0f0003 ffffff ffffff ffffff	-1.566138	1.907001
0189 0527	-2	00007f 000001 f00007 ffffff ffffff ffffff	-1.568421	1.897950
0190 0529	-3	0507fc 00001f 00000f ffffff ffffff ffffff	-1.575916	1.898687
0191 0530	-3	003ff8 0000f8 00000f ffffff ffffff ffffff	-1.583333	1.899306
0192 0531	-2	01ffc0 0003f0 000007 ffffff ffffff ffffff	-1.585492	1.890360
0193 0532	-2	03ff00 000fc0 000007 ffffff ffffff ffffff	-1.587629	1.881497
0194 0533	-2	1ffc00 007f00 000007 ffffff ffffff ffffff	-1.589744	1.872716
0195 0534	-3	bfc000 03f800 00000f ffffff ffffff ffffff	-1.596939	1.873256
0196 0535	-2	ff0000 07f001 ff4007 ffffff ffffff ffffff	-1.598985	1.864568
0197 0536	-3	800000 3fe00f ff800f ffffff ffffff ffffff	-1.606061	1.865014
0198 0537	-2	000001 ffc01f ff0007 ffffff ffffff ffffff	-1.608040	1.856418
0199 0537	-1	000007 fd001f fff003 ffffff ffffff ffffff	-1.605000	1.848976
0200 0538	-2	a0003f e8003f aff807 ffffff ffffff ffffff	-1.606965	1.840549
0201 0538	-2	f800ff a000ff abfe07 ffffff ffffff ffffff	-1.608911	1.832199
0202 0538	-1	f80ffe a000ff abfe03 ffffff ffffff ffffff	-1.605911	1.824990
0203 0539	-2	707ff4 4001fd 57fc07 ffffff ffffff ffffff	-1.607843	1.816802
0204 0540	-2	003fa0 8003fa 8bf807 ffffff ffffff ffffff	-1.609756	1.808686

curve 2: average offset is -1.609756 pixels from edge.
maximum offset is 2

Image:

curve 1

(x,y)	D(x)	Image	Aim	σ^2
0000 0344	5	ffffff ffffff ffffff fc0fc0 01ffff ffffff	5.000000	0.000000
0001 0343	3	ffffff ffffff ffffff f007e0 01ffff ffc1ff	4.000000	1.000000

0002 0343	3	ffffff ffffff ffffff f007f0 01ffff ffc07f	3.666667	0.888888
0003 0342	5	ffffff ffffff ffffff fc07fc 03ffff ffc003	4.000000	1.000000
0004 0342	6	ffffff ffffff ffffff fe07fe 0fffff ffc000	4.400000	1.439998
0005 0343	5	ffffff ffffff ffffff fc07ff ffffff ff0001	4.500000	1.250000
0006 0343	4	ffffff ffffff ffffff f807fc 7fffff ff0000	4.428571	1.102042
0007 0341	5	ffffff ffffff ffffff fc07f0 07ffff ffe000	4.500000	1.000000
0008 0340	6	ffffff ffffff ffffff fe03e0 03ffff fff800	4.666667	1.111113
0009 0339	7	ffffff ffffff ffffff ff03f0 01ffff ffff00	4.900000	1.490004
0010 0339	7	ffffff ffffff ffffff ff03e0 01ffff ffff80	5.090909	1.719014
0011 0339	4	ffffff ffffff ffffff f807f0 01ffff ffff00	5.000000	1.666673
0012 0339	4	ffffff ffffff ffffff f807f8 03ffff ffff00	4.923077	1.609472
0013 0340	5	ffffff ffffff ffffff fc0ff8 3fffff fffe00	4.928571	1.494901
0014 0339	5	ffffff ffffff ffffff fc07ff 7fffff ffff00	4.933333	1.395560
0015 0339	6	ffffff ffffff ffffff fe07ff 1fffff fffc00	5.000000	1.375005
0016 0339	6	ffffff ffffff ffffff fe07ff 07ffff fffc00	5.058823	1.349486
0017 0339	6	ffffff ffffff ffffff fe07f8 01ffff fff000	5.111111	1.320992
0018 0339	5	ffffff ffffff ffffff fc07f0 01ffff ffe000	5.105263	1.252082
0019 0339	5	ffffff ffffff ffffff fc07e0 01ffff ffc000	5.099999	1.190007
0020 0339	5	ffffff ffffff ffffff fc07e0 01ffff ff8000	5.095238	1.133792
0021 0339	3	ffffff ffffff ffffff f00ff0 07ffff ff0000	5.000000	1.272733
0022 0339	5	ffffff ffffff ffffff fc0ffc 07ffff fc0000	5.000000	1.217397
0023 0339	5	ffffff ffffff ffffff fc0fff 07ffff fc0000	5.000000	1.166673
0024 0340	4	ffffff ffffff ffffff f80fff ffffff f80000	4.960000	1.158406
0025 0340	5	ffffff ffffff ffffff fc07ff e3ffff f00000	4.961538	1.113914
0026 0340	5	ffffff ffffff ffffff fc03ff 80ffff e00000	4.962962	1.072713
0027 0341	5	ffffff ffffff ffffff fc07fe 007fff 000000	4.964285	1.034449
0028 0342	4	ffffff ffffff ffffff f80ff8 00fffe 000280	4.931034	1.029737
0029 0342	4	ffffff ffffff ffffff f80ff8 00fffe 00008a	4.899999	1.023343
0030 0342	3	ffffff ffffff ffffff f00ff8 00fffe 00000a	4.838709	1.103027
0031 0342	4	ffffff ffffff ffffff f80ffe 03fff8 0003e8	4.812499	1.089855
0032 0343	3	ffffff ffffff ffffff f01ffc 07fff8 001ffd	4.757575	1.153360
0033 0343	4	ffffff ffffff ffffff f807ff fffffc 041fff	4.735293	1.135821
0034 0343	5	ffffff ffffff ffffff fc07ff fc7ffc 011fff	4.742857	1.105313
0035 0344	4	ffffff ffffff ffffff f800ff f803fc 003fff	4.722221	1.089515
0036 0344	4	ffffff ffffff ffffff f8007f f003fe 003fff	4.702702	1.073783
0037 0344	9	ffffff ffffff ffffff ffc03f e001fe 0003ff	4.815789	1.518706
0038 0345	7	ffffff ffffff ffffff ff007f c003fc 0005ff	4.871794	1.598955
0039 0347	5	ffffff ffffff ffffff fc01ff 801ffc 001fff	4.875000	1.559380
0040 0348	4	ffffff ffffff ffffff f803ff 80fff8 003fff	4.853658	1.539563
0041 0349	4	ffffff ffffff ffffff f8007f fc07f4 007ffe	4.833333	1.519845
0042 0350	3	ffffff ffffff ffffff f000ff fe0038 00fff8	4.790698	1.560846
0043 0350	8	ffffff ffffff ffffff ff803f fe001f 80ff80	4.863636	1.754132
0044 0351	7	ffffff ffffff ffffff ff003f fc003f c1fc10	4.911111	1.814320
0045 0354	6	ffffff ffffff ffffff fe01ff e003ff 800fff	4.934783	1.800091
0046 0355	5	ffffff ffffff ffffff fc01ff e00fff c07fff	4.936171	1.761879
0047 0355	7	ffffff ffffff ffffff ff001f fc7fff f07c05	4.979167	1.812062
0048 0354	8	ffffff ffffff ffffff ff8003 ffffff f80a80	5.040817	1.957512
0049 0353	15	ffffff ffffff ffffff ffff00 ff1fff f00001	5.240000	3.862397
0050 0353	15	ffffff ffffff ffffff ffff01 fc07ff e00155	5.431373	5.617836
0051 0354	10	ffffff ffffff ffffff ffe00f e007ff abe3ae	5.519231	5.903474
0052 0357	7	ffffff ffffff ffffff ff007f 001ff1 7f8055	5.547170	5.832680
0053 0356	8	ffffff ffffff ffffff ff80ff 000fe0 3ffe00	5.592592	5.834019
0054 0356	8	ffffff ffffff ffffff ff80ff 800f82 3fff00	5.636364	5.831401
0055 0354	9	ffffff ffffff ffffff ffc0ff e00f80 2ffff8	5.696428	5.925701

0056 0353	9	ffffff ffffff ffffff ffc07f f01f00 47ffff	5.754385	6.009851
0057 0352	8	ffffff ffffff ffffff ff807f c3ff80 03ffff	5.793103	5.991680
0058 0352	6	ffffff ffffff ffffff fe007e 03ff80 03ffff	5.796609	5.890840
0059 0350	6	ffffff ffffff ffffff fe00f8 003f80 03ffff	5.799999	5.793335
0060 0349	7	ffffff ffffff ffffff ff01fc 001fe0 07ffff	5.819671	5.721584
0061 0350	4	ffffff ffffff ffffff f803f8 007f8a afffff	5.790322	5.681844
0062 0349	5	ffffff ffffff ffffff fc07fe 007f85 57fff7	5.777777	5.601410
0063 0348	8	ffffff ffffff ffffff ff81ff 803f80 8fffeb	5.812500	5.589842
0064 0348	8	ffffff ffffff ffffff ff83ff 80ff00 0fffab	5.846153	5.576330
0065 0347	8	ffffff ffffff ffffff ff81ff ffff00 501540	5.878788	5.561062
0066 0346	5	ffffff ffffff ffffff fc03ff afff80 000800	5.865671	5.489417
0067 0345	5	ffffff ffffff ffffff fc07f0 03ffc0 010000	5.852941	5.419551
0068 0345	5	ffffff ffffff ffffff fc07f0 03ffc0 014050	5.840579	5.351397
0069 0343	7	ffffff ffffff ffffff ff03f0 007ff8 000051	5.857142	5.293879
0070 0342	8	ffffff ffffff ffffff ff81f8 007ffe 000080	5.887323	5.283082
0071 0342	6	ffffff ffffff ffffff fe03fc 00fffe 000000	5.888888	5.209881
0072 0342	6	ffffff ffffff ffffff fe03fe 00ffff 800002	5.890410	5.138680
0073 0342	8	ffffff ffffff ffffff ff83ff 8fffff c0000a	5.918918	5.128563
0074 0342	7	ffffff ffffff ffffff ff03ff ffffff c00002	5.933332	5.075561
0075 0342	6	ffffff ffffff ffffff fe03ff bfffff e00000	5.934210	5.008832
0076 0342	4	ffffff ffffff ffffff f807f8 0fffff e00000	5.909090	4.991739
0077 0342	5	ffffff ffffff ffffff fc0fe0 0fffff f80000	5.897435	4.938200
0078 0341	5	ffffff ffffff ffffff fc07f0 03ffff fc0000	5.886075	4.885760
0079 0340	6	ffffff ffffff ffffff fe03e0 01ffff ff8000	5.887499	4.824852
0080 0340	6	ffffff ffffff ffffff fe07e0 03ffff ffe000	5.888888	4.765439
0081 0340	5	ffffff ffffff ffffff fc03f8 03ffff fffc00	5.878048	4.716840
0082 0340	6	ffffff ffffff ffffff fe07fc 07ffff fffe00	5.879517	4.660191
0083 0340	8	ffffff ffffff ffffff ff83ff ffffff fffe00	5.904761	4.657607
0084 0340	8	ffffff ffffff ffffff ff83ff bfffff fffe00	5.929411	4.653850
0085 0341	6	ffffff ffffff ffffff fe07fc 07ffff ffff00	5.930232	4.599794
0086 0341	6	ffffff ffffff ffffff fe07f0 07ffff fffc00	5.931034	4.546977
0087 0341	5	ffffff ffffff ffffff fc07e0 07ffff fff000	5.920454	4.505046
0088 0341	5	ffffff ffffff ffffff fc07c0 07ffff fff000	5.910111	4.463840
0089 0340	4	ffffff ffffff ffffff f807f0 03ffff fff000	5.888888	4.454331
0090 0340	4	ffffff ffffff ffffff f807f8 03ffff ffe000	5.868131	4.444162
0091 0340	6	ffffff ffffff ffffff fe07fe 0fffff fff000	5.869564	4.396042
0092 0340	6	ffffff ffffff ffffff fe07ff ffffff fff803	5.870966	4.348957
0093 0341	5	ffffff ffffff ffffff fc07ff c1ffff fff0ff	5.861701	4.310673
0094 0341	4	ffffff ffffff ffffff f807ff 007fff fffdff	5.842104	4.301401
0095 0343	4	ffffff ffffff ffffff f807f8 01ffff ffffff	5.822916	4.291573
0096 0343	4	ffffff ffffff ffffff f807f0 01ffff ffc3ff	5.804122	4.281237
0097 0343	4	ffffff ffffff ffffff f807f0 01ffff ff01ff	5.785713	4.270422
0098 0343	3	ffffff ffffff ffffff f007f0 01ffff ff007f	5.757575	4.304883
0099 0343	3	ffffff ffffff ffffff f007fc 03ffff fc001f	5.729999	4.337118
0100 0343	5	ffffff ffffff ffffff fc07ff 07ffff fc000f	5.722771	4.299403
0101 0345	3	ffffff ffffff ffffff f00fff ffffff f0001f	5.696077	4.329222
0102 0345	3	ffffff ffffff ffffff f007ff f007ff f4141c	5.669901	4.357075
0103 0345	4	ffffff ffffff ffffff f8007f f007ff c00100	5.653845	4.341732
0104 0345	5	ffffff ffffff ffffff fc007f e007ff c07401	5.647618	4.304414
0105 0346	8	ffffff ffffff ffffff ff807f 8007ff effe0e	5.669810	4.315516
0106 0346	8	ffffff ffffff ffffff ff807f 800fff effe3f	5.691588	4.325454
0107 0347	5	ffffff ffffff ffffff fc00ff c07fff fffdff	5.685184	4.289788
0108 0348	4	ffffff ffffff ffffff f800ff e8ffff ffffff	5.669724	4.276246
0109 0349	5	ffffff ffffff ffffff fc001f ff0007 ffdfff	5.663636	4.241410

0110 0350	4	ffffff ffffff ffffff f8003f fe000f fe7ff8	5.648648	4.227910
0111 0352	8	ffffff ffffff ffffff ff803f f8003f 03fe02	5.669642	4.239086
0112 0353	7	ffffff ffffff ffffff ff007f f8007e 11f004	5.681415	4.217098
0113 0355	7	ffffff ffffff ffffff ff007f f007f7 f01ffc	5.692982	4.195222
0114 0356	6	ffffff ffffff ffffff fe00ff e00fff e0fffa	5.695652	4.159552
0115 0357	7	ffffff ffffff ffffff ff0007 ffffff f7ffd0	5.706896	4.138234
0116 0358	6	ffffff ffffff ffffff fe0007 ffffff e3ffa8	5.709401	4.103590
0117 0361	9	ffffff ffffff ffffff ffc01e 07fdfc 000000	5.737288	4.159799
0118 0362	8	ffffff ffffff ffffff ff8020 01f803 fe0028	5.756302	4.167506
0119 0360	5	ffffff ffffff ffffff fc00fc 003f83 fff800	5.750000	4.137502
0120 0359	6	ffffff ffffff ffffff fe01fc 000fd1 fffc00	5.752066	4.103819
0121 0357	7	ffffff ffffff ffffff ff007f 0007ff ffff00	5.762295	4.082839
0122 0356	8	ffffff ffffff ffffff ff80ff c00fff ffffc0	5.780488	4.090023
0123 0354	8	ffffff ffffff ffffff ff803f f83fff feffe0	5.798387	4.096443
0124 0353	8	ffffff ffffff ffffff ff801f f1ffff fe0750	5.816000	4.102138
0125 0352	5	ffffff ffffff ffffff fc001e 00fff3 fe8008	5.809524	4.074826
0126 0351	6	ffffff ffffff ffffff fe003c 003ff9 7d0000	5.811023	4.043025
0127 0351	5	ffffff ffffff ffffff fc01f0 003ff0 017000	5.804687	4.016539
0128 0351	4	ffffff ffffff ffffff f801f0 003ff1 4551f0	5.790697	4.010457
0129 0350	6	ffffff ffffff ffffff fe01fe 003ff8 0001fe	5.792307	3.979943
0130 0350	7	ffffff ffffff ffffff ff01fe 003ff8 0003ff	5.801526	3.960608
0131 0349	8	ffffff ffffff ffffff ff80ff dffffe 0007ff	5.818181	3.966942
0132 0348	8	ffffff ffffff ffffff ff80ff ffffff 8003ff	5.834586	3.972640
0133 0346	6	ffffff ffffff ffffff fe00fe 03ffff e003ff	5.835820	3.943195
0134 0346	6	ffffff ffffff ffffff fe01f8 03ffff f80fff	5.837036	3.914185
0135 0345	7	ffffff ffffff ffffff ff01f0 00ffff ffffff	5.845587	3.895276
0136 0345	7	ffffff ffffff ffffff ff01f0 00ffff ffffff	5.854013	3.876500
0137 0345	6	ffffff ffffff ffffff fe03e0 00ffff ffc1ff	5.855072	3.848562
0138 0345	4	ffffff ffffff ffffff f807f0 01ffff ffc1ff	5.841726	3.845454
0139 0346	4	ffffff ffffff ffffff f807f0 07ffff fe000f	5.828571	3.842040
0140 0346	6	ffffff ffffff ffffff fe07f8 3fffff fe0007	5.829787	3.814997
0141 0346	6	ffffff ffffff ffffff fe07fe ffffff fe0003	5.830986	3.788331
0142 0346	4	ffffff ffffff ffffff f807f8 3fffff fc0001	5.818182	3.785116
0143 0345	5	ffffff ffffff ffffff fc07c0 0fffff ff0000	5.812500	3.763448
0144 0344	6	ffffff ffffff ffffff fe03e0 07ffff ffc000	5.813793	3.737732
0145 0343	6	ffffff ffffff ffffff fe07c0 03ffff fffc00	5.815068	3.712367
0146 0342	7	ffffff ffffff ffffff ff03e0 00ffff fffe00	5.823129	3.696597
0147 0342	5	ffffff ffffff ffffff fc03e0 01ffff fffe00	5.817567	3.676170
0148 0342	5	ffffff ffffff ffffff fc03f8 03ffff fffe00	5.812080	3.655951
0149 0342	8	ffffff ffffff ffffff ff83fe 0fffff fffe00	5.826666	3.663282
0150 0342	8	ffffff ffffff ffffff ff81ff bfffff fffe00	5.841059	3.670098
0151 0343	7	ffffff ffffff ffffff ff03ff 0fffff fff000	5.848684	3.654729
0152 0343	7	ffffff ffffff ffffff ff01fc 07ffff fff000	5.856209	3.639450
0153 0344	6	ffffff ffffff ffffff fe03f0 03ffff ff8000	5.857142	3.615952
0154 0344	6	ffffff ffffff ffffff fe03e0 03ffff ff0000	5.858064	3.592752
0155 0344	6	ffffff ffffff ffffff fe03e0 03ffff fe0000	5.858974	3.569850
0156 0344	6	ffffff ffffff ffffff fe07e0 03ffff fe0000	5.859872	3.547238
0157 0344	4	ffffff ffffff ffffff f807f8 07ffff f80000	5.848101	3.546539
0158 0344	5	ffffff ffffff ffffff fc07f8 0fffff e00002	5.842767	3.528728
0159 0344	6	ffffff ffffff ffffff fe03ff ffffff e00000	5.843750	3.506824
0160 0344	6	ffffff ffffff ffffff fe03ff ffffff e00000	5.844720	3.485194
0161 0346	4	ffffff ffffff ffffff f807ff 00ffff 800000	5.833333	3.484559
0162 0346	4	ffffff ffffff ffffff f803fe 00fffe 000000	5.822085	3.483678
0163 0346	5	ffffff ffffff ffffff fc03fc 007ff8 000008	5.817073	3.466530

0164 0346	4	ffffff ffffff ffffff f803f8 007ff8 000000	5.806060	3.465410
0165 0346	4	ffffff ffffff ffffff f803fc 00fff8 0002a8	5.795180	3.464068
0166 0346	6	ffffff ffffff ffffff fe03fe 00fff8 00002a	5.796407	3.443577
0167 0347	5	ffffff ffffff ffffff fc07ff 07ffc0 001fd0	5.791666	3.426835
0168 0347	6	ffffff ffffff ffffff fe03ff f7ffe0 007ffd	5.792899	3.406814
0169 0347	6	ffffff ffffff ffffff fe007f ffffe0 007fff	5.794117	3.387025
0170 0347	6	ffffff ffffff ffffff fe003f fc1ff0 001fff	5.795321	3.367464
0171 0348	10	ffffff ffffff ffffff ffe03f f003f0 003fff	5.819767	3.450072
0172 0349	9	ffffff ffffff ffffff ffc07f c007f0 007fff	5.838150	3.488251
0173 0350	7	ffffff ffffff ffffff ff00ff 000ff8 001fff	5.844827	3.475919
0174 0350	6	ffffff ffffff ffffff fe00ff 800ff8 000fff	5.845714	3.456190
0175 0352	4	ffffff ffffff ffffff f803ff 007fe0 00ffff	5.835227	3.455798
0176 0353	5	ffffff ffffff ffffff fc07ff 01ffe0 00ffff	5.830508	3.440194
0177 0354	4	ffffff ffffff ffffff f800ff f82fe0 00fff0	5.820224	3.439587
0178 0355	7	ffffff ffffff ffffff ff01ff f00140 01ffc1	5.826815	3.428104
0179 0357	5	ffffff ffffff ffffff fc03ff c001f0 07f000	5.822222	3.412834
0180 0357	7	ffffff ffffff ffffff ff01ff c001fc 1fe100	5.828729	3.401601
0181 0360	4	ffffff ffffff ffffff f80fff 001ff8 00ffff	5.818681	3.401186
0182 0360	4	ffffff ffffff ffffff f807ff 803ffe 03ffff	5.808743	3.400577
0183 0359	6	ffffff ffffff ffffff fe003f f07fff c05017	5.809782	3.382293
0184 0359	7	ffffff ffffff ffffff ff001f ffffff c05417	5.816216	3.371626
0185 0360	11	ffffff ffffff ffffff fff01f e3fffe 00e028	5.844085	3.497196
0186 0360	11	ffffff ffffff ffffff fff01f 80fffe 00faa8	5.871657	3.619894
0187 0360	6	ffffff ffffff ffffff fe00fe 003ffa bf02aa	5.872340	3.600724
0188 0359	7	ffffff ffffff ffffff ff00ff 001ff5 5fc001	5.878306	3.588363
0189 0358	7	ffffff ffffff ffffff ff00ff 0007c0 0ffe00	5.884210	3.576065
0190 0357	8	ffffff ffffff ffffff ff80ff c007c0 1fffc0	5.895288	3.580657
0191 0355	9	ffffff ffffff ffffff ffc07f f00700 01ffff	5.911458	3.611949
0192 0354	8	ffffff ffffff ffffff ff803f f83f80 00ffff	5.922279	3.615721
0193 0354	4	ffffff ffffff ffffff f800fe 03ff80 03ffff	5.912371	3.616032
0194 0353	5	ffffff ffffff ffffff fc007c 01ff80 01ffff	5.907692	3.601735
0195 0352	6	ffffff ffffff ffffff fe03f8 003fc0 03ffff	5.908163	3.583401
0196 0351	5	ffffff ffffff ffffff fc03f8 003fe0 01ffff	5.903553	3.569376
0197 0350	6	ffffff ffffff ffffff fe03fe 003fc0 02fffb	5.904040	3.551392
0198 0349	7	ffffff ffffff ffffff ff01ff 801fc0 01fff1	5.909548	3.539548
0199 0349	7	ffffff ffffff ffffff ff01ff c07f80 07ffd1	5.915000	3.527767
0200 0349	7	ffffff ffffff ffffff ff01ff f7ff00 01ffc0	5.920398	3.516048
0201 0349	3	ffffff ffffff ffffff f007ff 7fff00 000040	5.905941	3.540650
0202 0348	4	ffffff ffffff ffffff f80ff8 0fffc0 000200	5.896552	3.541016
0203 0347	5	ffffff ffffff ffffff fc0fc0 07ffc0 000000	5.892157	3.527581
0204 0346	5	ffffff ffffff ffffff fc0fe0 03ffe0 000000	5.887805	3.514239
0205 0345	5	ffffff ffffff ffffff fc07e0 01fff8 000100	5.883495	3.500985
0206 0345	4	ffffff ffffff ffffff f80ff0 01fffc 000000	5.874396	3.501128
0207 0345	3	ffffff ffffff ffffff f01ff8 03fffe 000000	5.860577	3.523827
0208 0344	4	ffffff ffffff ffffff f80ffc 03ffff 800000	5.851675	3.523450
0209 0343	5	ffffff ffffff ffffff fc07ff dfffff c00001	5.847619	3.510109
0210 0343	5	ffffff ffffff ffffff fc07ff ffffff c00000	5.843602	3.496866
0211 0343	4	ffffff ffffff ffffff f807f0 1fffff f00000	5.834906	3.496326
0212 0343	4	ffffff ffffff ffffff f80fc0 0fffff f80000	5.826291	3.495643
0213 0343	3	ffffff ffffff ffffff f01fc0 0fffff fc0000	5.813084	3.516460
0214 0342	4	ffffff ffffff ffffff f80fc0 07ffff ff0000	5.804651	3.515321
0215 0340	4	ffffff ffffff ffffff f807f0 03ffff fff800	5.796297	3.514053
0216 0340	4	ffffff ffffff ffffff f80ff8 03ffff fffe00	5.788019	3.512660
0217 0340	6	ffffff ffffff ffffff fe0ffc 0fffff ffff00	5.788991	3.496754

0218 0340	6	ffffff ffffff ffffff fe0ffe 3fffff ffff80	5.789955	3.480987
0219 0340	6	ffffff ffffff ffffff fe07ff bfffff ffff80	5.790909	3.465366
0220 0340	6	ffffff ffffff ffffff fe07fe 0fffff ffffc0	5.791855	3.449884
0221 0340	6	ffffff ffffff ffffff fe07f8 03ffff ffff80	5.792793	3.434541
0222 0340	6	ffffff ffffff ffffff fe07f8 03ffff fffe00	5.793722	3.419329
0223 0340	6	ffffff ffffff ffffff fe0fe0 03ffff fffe00	5.794643	3.404253
0224 0340	6	ffffff ffffff ffffff fe0fe0 03ffff fffc00	5.795556	3.389311
0225 0340	4	ffffff ffffff ffffff f80ff8 03ffff fff800	5.787611	3.388517
0226 0340	4	ffffff ffffff ffffff f80ff8 07ffff fff800	5.779736	.387606
0227 0340	5	ffffff ffffff ffffff fc07ff 8fffff fffc00	5.776316	3.375403
0228 0340	5	ffffff ffffff ffffff fc03ff ffffff fffc07	5.772926	3.363280
0229 0341	4	ffffff ffffff ffffff f807ff c0ffff fffdff	5.765217	3.362264
0230 0342	4	ffffff ffffff ffffff f803fe 00ffff ffffff	5.757576	3.361143
0231 0343	5	ffffff ffffff ffffff fc07f8 01ffff fff5ff	5.754310	3.349119
0232 0343	4	ffffff ffffff ffffff f807f0 00ffff ffe0ff	5.746781	3.347896
0233 0343	4	ffffff ffffff ffffff f807f0 01ffff ff803f	5.739316	3.346572
0234 0343	3	ffffff ffffff ffffff f007f8 01ffff ff001f	5.727659	3.364128
0235 0343	3	ffffff ffffff ffffff f007fc 03ffff fe0007	5.716101	3.381268
0236 0343	4	ffffff ffffff ffffff f807ff 47ffff fe0003	5.708860	3.379372
0237 0344	2	ffffff ffffff ffffff e007ff f80fff f80002	5.693277	3.422727
0238 0344	4	ffffff ffffff ffffff f800ff f803ff f80000	5.686192	3.420352
0239 0344	5	ffffff ffffff ffffff fc003f f801ff f82a00	5.683333	3.408055
0240 0344	8	ffffff ffffff ffffff ff803f f800ff f8ff80	5.692945	3.416094
0241 0345	8	ffffff ffffff ffffff ff803f e001ff f7ff07	5.702479	3.423877
0242 0346	7	ffffff ffffff ffffff ff007f c00fff fffe3f	5.707819	3.416688
0243 0348	4	ffffff ffffff ffffff f800ff e07fff fffbff	5.700819	3.414592
0244 0348	5	ffffff ffffff ffffff fc00ff fe3fff ffffff	5.697958	3.402650
0245 0350	4	ffffff ffffff ffffff f8003f fe000f be3ffc	5.691056	3.400491
0246 0350	4	ffffff ffffff ffffff f8003f fe0007 e07ff8	5.684210	3.398255
0247 0351	9	ffffff ffffff ffffff ffc01f fc001f c07c00	5.697580	3.428709
0248 0352	8	ffffff ffffff ffffff ff803f fc003e 3e003e	5.706827	3.436142
0249 0355	7	ffffff ffffff ffffff ff007f f007f7 f07ffc	5.711999	3.429059
0250 0355	7	ffffff ffffff ffffff ff007f f007ff f8fffc	5.717131	3.421984
0251 0357	7	ffffff ffffff ffffff ff0007 ffffff f1ffc0	5.722221	3.414908
0252 0358	6	ffffff ffffff ffffff fe000f fbffff c00080	5.723320	3.401714
0253 0360	9	ffffff ffffff ffffff ffc03e 00fe38 fe8000	5.736220	3.430426
0254 0361	8	ffffff ffffff ffffff ff8070 00fc07 ffc000	5.745098	3.436987
0255 0360	4	ffffff ffffff ffffff f803f8 003fa3 fff800	5.738281	3.435411
0256 0358	6	ffffff ffffff ffffff fe00fe 000fff fffe00	5.739299	3.422306
0257 0356	8	ffffff ffffff ffffff ff80ff 8007ff efffc0	5.748062	3.428774
0258 0355	9	ffffff ffffff ffffff ffc07f c007ff fdfff0	5.760617	3.456211
0259 0353	9	ffffff ffffff ffffff ffc03f f07fff fe0551	5.773077	3.483122
0260 0352	9	ffffff ffffff ffffff ffc03f faffff ff0000	5.785440	3.509518
0261 0352	6	ffffff ffffff ffffff fe003e 00ffeb f8002a	5.786259	3.496301
0262 0351	5	ffffff ffffff ffffff fc007c 007ffc 7d4005	5.783269	3.485352
0263 0350	6	ffffff ffffff ffffff fe03f0 003ff8 a2a880	5.784091	3.472325
0264 0350	4	ffffff ffffff ffffff f803f8 003ff8 0283f8	5.777358	3.471190
0265 0349	5	ffffff ffffff ffffff fc01fc 007ffc 0011ff	5.774436	3.460407
0266 0348	7	ffffff ffffff ffffff ff00ff 003ffe 0001ff	5.779026	3.453053
0267 0347	9	ffffff ffffff ffffff ffc0ff c1ffff c013ff	5.791044	3.478734
0268 0347	9	ffffff ffffff ffffff ffc0ff ffffff c007ff	5.802973	3.503941
0269 0345	5	ffffff ffffff ffffff fc007f c7ffff f807ff	5.799999	3.493344
0270 0345	5	ffffff ffffff ffffff fc01fc 07ffff ff07ff	5.797047	3.482806
0271 0344	6	ffffff ffffff ffffff fe03e0 01ffff ffffff	5.797793	3.470152

0272 0344	6	ffffff ffffff ffffff fe03e0 01ffff ffffff	5.798534	3.457594
0273 0344	5	ffffff ffffff ffffff fc07c0 01ffff ffe3ff	5.795619	3.447294
0274 0344	4	ffffff ffffff ffffff f80fc0 03ffff ffe0ff	5.789090	3.446438
0275 0343	4	ffffff ffffff ffffff f807f0 01ffff ffc007	5.782608	3.445504
0276 0343	5	ffffff ffffff ffffff fc07f8 07ffff ffc001	5.779783	3.435267
0277 0343	6	ffffff ffffff ffffff fe07ff ffffff ff8000	5.780575	3.423086
0278 0343	6	ffffff ffffff ffffff fe07ff ffffff ff0000	5.781361	3.410993
0279 0342	4	ffffff ffffff ffffff f803f0 0fffff ffe000	5.774999	3.410104
0280 0342	4	ffffff ffffff ffffff f80fe0 0fffff ffe000	5.768682	3.409139
0281 0342	4	ffffff ffffff ffffff f80f80 0fffff fff800	5.762411	3.408103
0282 0341	5	ffffff ffffff ffffff fc07c0 07ffff ffff00	5.759717	3.398107
0283 0340	5	ffffff ffffff ffffff fc07c0 03ffff ffff00	5.757041	3.388166
0284 0340	4	ffffff ffffff ffffff f807e0 03ffff fffe00	5.750876	3.387072
0285 0340	5	ffffff ffffff ffffff fc07f8 07ffff fffe00	5.748251	3.377193
0286 0340	6	ffffff ffffff ffffff fe07f8 3fffff fffe00	5.749128	3.365645
0287 0340	6	ffffff ffffff ffffff fe03ff ffffff fffe00	5.750000	3.354173
0288 0340	6	ffffff ffffff ffffff fe07ff 3fffff fffc00	5.750865	3.342783
0289 0340	6	ffffff ffffff ffffff fe03f8 03ffff ffe000	5.751724	3.331469
0290 0340	6	ffffff ffffff ffffff fe03f8 03ffff ffe000	5.752577	3.320232
0291 0340	5	ffffff ffffff ffffff fc0fe0 03ffff ff8000	5.750000	3.310796
0292 0340	5	ffffff ffffff ffffff fc0fe0 03ffff ff8000	5.747440	3.301408
0293 0340	4	ffffff ffffff ffffff f80fe0 03ffff fe0000	5.741496	3.300531
0294 0340	4	ffffff ffffff ffffff f80ff0 03ffff fe0000	5.735593	3.299588
0295 0340	6	ffffff ffffff ffffff fe0ff8 0fffff f80000	5.736486	3.288678
0296 0340	5	ffffff ffffff ffffff fc0fff 8fffff f80000	5.734006	3.279423
0297 0341	5	ffffff ffffff ffffff fc0fff dfffff f00000	5.731543	3.270222
0298 0341	5	ffffff ffffff ffffff fc0fff c1ffff e00000	5.729096	3.261069
0299 0342	4	ffffff ffffff ffffff f807fe 00ffff 800000	5.723333	3.260131
0300 0343	4	ffffff ffffff ffffff f80ffc 01fffe 000000	5.717607	3.259133
0301 0344	3	ffffff ffffff ffffff f01fe0 03fffc 000002	5.708609	3.272719
0302 0344	3	ffffff ffffff ffffff f01fe0 03fffc 0002aa	5.699669	3.286051
0303 0343	3	ffffff ffffff ffffff f00ffc 01fff8 000000	5.690789	3.299139
0304 0343	3	ffffff ffffff ffffff f007fc 01fff8 0007f4	5.681966	3.311984
0305 0343	5	ffffff ffffff ffffff fc0fff c7fffc 001fff	5.679738	3.302677
0306 0343	5	ffffff ffffff ffffff fc07ff fffffc 0007ff	5.677524	3.293417
0307 0345	3	ffffff ffffff ffffff f001ff f007f0 001fff	5.668830	3.305925
0308 0345	3	ffffff ffffff ffffff f000ff f003fc 001fff	5.660193	3.318202
0309 0346	7	ffffff ffffff ffffff ff00ff c007f8 000fff	5.664515	3.313270
0310 0347	6	ffffff ffffff ffffff fe01ff 0007f8 001fff	5.665594	3.302979
0311 0348	5	ffffff ffffff ffffff fc03fe 000ff8 000fff	5.663461	3.293808
0312 0349	3	ffffff ffffff ffffff f007fe 007ff0 007fff	5.654951	3.305879
0313 0350	2	ffffff ffffff ffffff e00fff e3fff0 003ffe	5.643311	3.337760
0314 0350	4	ffffff ffffff ffffff f801ff f80bf8 003ffe	5.638094	3.335710
0315 0351	3	ffffff ffffff ffffff f400ff fc003f 007fc0	5.629745	3.347107
0316 0351	7	ffffff ffffff ffffff ff007f fc001f c57f00	5.634068	3.342452
0317 0353	7	ffffff ffffff ffffff ff00ff f001ff c007ff	5.638363	3.337791
0318 0354	6	ffffff ffffff ffffff fe00ff f003ff c03fff	5.639497	3.327740
0319 0355	6	ffffff ffffff ffffff fe00ff f01fff e07f05	5.640624	3.317742
0320 0355	7	ffffff ffffff ffffff ff000f ff7fff f01401	5.644858	3.313144
0321 0353	11	ffffff ffffff ffffff fff101 ffffff fc0001	5.661489	3.391642
0322 0354	15	ffffff ffffff ffffff ffff03 fe3fff e00a02	5.690401	3.650301
0323 0354	14	ffffff ffffff ffffff fffe07 e007ff ab83ea	5.716047	3.851496
0324 0356	8	ffffff ffffff ffffff ff803f 800ffe 8fc2aa	5.723075	3.855644
0325 0357	7	ffffff ffffff ffffff ff01fe 001fe1 1ff800	5.726992	3.848806

0326 0356	8	fffff fffff fffff ff80ff 000fe0 2fff00	5.733943	3.852786
0327 0355	8	fffff fffff fffff ff81ff c00705 57fff0	5.740852	3.856646
0328 0354	8	fffff fffff fffff ff80ff f00e00 a3fffe	5.747719	3.860388
0329 0353	7	fffff fffff fffff ff00ff c7ff00 07ffff	5.751513	3.853431
0330 0353	7	fffff fffff fffff ff01ff 07ff00 07ffff	5.755285	3.846487
0331 0352	4	fffff fffff fffff f800f8 01ff00 0fffff	5.749998	3.844153
0332 0351	5	fffff fffff fffff fc07f0 007fc0 07ffff	5.747746	3.834292
0333 0350	5	fffff fffff fffff fc07f0 003fe0 0fffff	5.745507	3.824481
0334 0350	4	fffff fffff fffff f80ff8 007f80 0fffef	5.740297	3.822132
0335 0348	8	fffff fffff fffff ff83ff 803f80 03ffeb	5.747022	3.825908
0336 0347	9	fffff fffff fffff ffc1ff c03fc0 07fff5	5.756675	3.845864
0337 0347	8	fffff fffff fffff ff81ff ffff00 407550	5.763312	3.849330
0338 0346	6	fffff fffff fffff fe00ff ffffc0 0002a8	5.764010	3.838140
0339 0344	6	fffff fffff fffff fe03fe 03fff8 000280	5.764704	3.827012
0340 0344	6	fffff fffff fffff fe03f8 01ffe0 000000	5.765394	3.815951
0341 0344	5	fffff fffff fffff fc0fe0 01fff0 000000	5.763156	3.806498
0342 0343	6	fffff fffff fffff fe07f0 00fffc 000010	5.763847	3.795563
0343 0343	4	fffff fffff fffff f807f0 01fffc 000001	5.758719	3.793548
0344 0343	3	fffff fffff fffff f00ff8 01fffe 000000	5.750723	3.804547
0345 0343	5	fffff fffff fffff fc07fc 03ffff 000000	5.748554	3.795175
0346 0343	5	fffff fffff fffff fc0fff 1fffff c00010	5.746397	3.785850
0347 0343	5	fffff fffff fffff fc0fff ffffff c00000	5.744252	3.776567
0348 0342	4	fffff fffff fffff f807ff bfffff e00000	5.739254	3.774438
0349 0341	5	fffff fffff fffff fc07f0 07ffff fc0000	5.737142	3.765214
0350 0340	6	fffff fffff fffff fe03f8 03ffff fe0000	5.737891	3.754680
0351 0340	5	fffff fffff fffff fc0fe0 03ffff ff8000	5.735795	3.745555
0352 0339	6	fffff fffff fffff fe07f0 01ffff fff000	5.736543	3.735139
0353 0338	5	fffff fffff fffff fc03f8 00ffff fffe00	5.734463	3.726113
0354 0338	5	fffff fffff fffff fc03fc 01ffff ffff80	5.732394	3.717135
0355 0339	5	fffff fffff fffff fc07ff 1fffff ffff00	5.730337	3.708194
0356 0339	5	fffff fffff fffff fc07ff ffffff ffff80	5.728291	3.699297
0357 0339	7	fffff fffff fffff ff07ff 07ffff ffffc0	5.731843	3.693467
0358 0339	7	fffff fffff fffff ff07ff 03ffff ffffc0	5.735376	3.687647
0359 0339	7	fffff fffff fffff ff07f8 01ffff ffff00	5.738889	3.681832
0360 0339	7	fffff fffff fffff ff07f0 01ffff fffe00	5.742382	3.676027
0361 0339	7	fffff fffff fffff ff07f0 05ffff fffc00	5.745856	3.670228
0362 0339	5	fffff fffff fffff fc07f0 01ffff fffc00	5.743802	3.661644
0363 0339	5	fffff fffff fffff fc07fc 07ffff fff000	5.741758	3.653099
0364 0339	6	fffff fffff fffff fe07fc 07ffff fff800	5.742466	3.643271
0365 0339	6	fffff fffff fffff fe03ff ffffff fffc07	5.743170	3.633497
0366 0339	7	fffff fffff fffff ff03ff f7ffff fffe1f	5.746594	3.627890
0367 0340	6	fffff fffff fffff fe00ff 807fff ffffff	5.747283	3.618206
0368 0341	5	fffff fffff fffff fc01fe 007fff ffffff	5.745258	3.609909
0369 0341	6	fffff fffff fffff fe01fc 007fff fff07f	5.745946	3.600326
0370 0341	6	fffff fffff fffff fe01fc 007fff ffe07f	5.746631	3.590797
0371 0342	4	fffff fffff fffff f803f8 00ffff ff801f	5.741936	3.589325
0372 0342	4	fffff fffff fffff f803fc 00ffff ff000f	5.737266	3.587816
0373 0343	4	fffff fffff fffff f807fc 07ffff fc0007	5.732621	3.586267
0374 0343	5	fffff fffff fffff fc07ff f7ffff fc0007	5.730667	3.578129
0375 0344	4	fffff fffff fffff f803ff f803ff fa0002	5.726064	3.576557
0376 0344	4	fffff fffff fffff f8007f f803ff e80000	5.721486	3.574955
0377 0344	6	fffff fffff fffff fe003f f001ff f0ba00	5.722222	3.565704
0378 0344	10	fffff fffff fffff ffe01f e001ff f2ff80	5.733510	3.604449
0379 0345	9	fffff fffff fffff ffc03f c007ff f7ff1f	5.742105	3.622970

0380 0345	9	ffffff ffffff ffffff ffc03f e007ff ffff7f	5.750657	3.641245
0381 0347	5	ffffff ffffff ffffff fc007f fc3fff ffffff	5.748692	3.633183
0382 0348	5	ffffff ffffff ffffff fc007f ff003f ffffff	5.746737	3.625158
0383 0349	4	ffffff ffffff ffffff f8001f ff8007 df1ffe	5.742188	3.623646
0384 0350	4	ffffff ffffff ffffff f8003f ff000f c07ff0	5.737662	3.622097
0385 0352	8	ffffff ffffff ffffff ff803f fc003f 80f802	5.743523	3.625941
0386 0353	9	ffffff ffffff ffffff ffc03f fc01fc 7c01ff	5.751938	3.643902
0387 0355	7	ffffff ffffff ffffff ff007f f007f7 f87fff	5.755155	3.638514
0388 0356	6	ffffff ffffff ffffff fe007f f83fff f9fff8	5.755784	3.629316
0389 0358	6	ffffff ffffff ffffff fe0007 ffffff e02f88	5.756410	3.620163
0390 0359	11	ffffff ffffff ffffff fff00f e3ffff 800000	5.769821	3.681047
0391 0361	9	ffffff ffffff ffffff ffc070 00fc01 ffc000	5.778061	3.698206
0392 0361	5	ffffff ffffff ffffff fd0078 007f07 fff000	5.776081	3.690331
0393 0359	6	ffffff ffffff ffffff fe01fc 001fd0 fffe00	5.776649	3.681091
0394 0358	7	ffffff ffffff ffffff ff00fe 000fff bfff80	5.779747	3.675550
0395 0355	9	ffffff ffffff ffffff ffc07f c007ff fffff0	5.787879	3.692387
0396 0354	10	ffffff ffffff ffffff ffe03f f003ff fffff8	5.798489	3.727662
0397 0354	8	ffffff ffffff ffffff ff803f e3ffff fe0aaa	5.804020	3.730444
0398 0353	8	ffffff ffffff ffffff ff807f e7ffff ff0005	5.809524	3.733151
0399 0352	6	ffffff ffffff ffffff fe003c 00fff3 fe0002	5.810000	3.723908
0400 0351	6	ffffff ffffff ffffff fe0078 007ff8 7d0000	5.810474	3.714707
0401 0350	6	ffffff ffffff ffffff fe03f8 003ff8 228ae0	5.810946	3.705553
0402 0350	5	ffffff ffffff ffffff fc03f8 003ff8 0201f8	5.808933	3.697987
0403 0350	5	ffffff ffffff ffffff fc03fe 007ff8 0003ff	5.806931	3.690449
0404 0350	6	ffffff ffffff ffffff fe03fe 00fff8 0003ff	5.807408	3.681427
0405 0348	8	ffffff ffffff ffffff ff80ff ffffff 8023ff	5.812809	3.684169
0406 0347	9	ffffff ffffff ffffff ffc07f ffffff c101ff	5.820640	3.700011
0407 0346	5	ffffff ffffff ffffff fc03fc 07ffff f803ff	5.818628	3.692587
0408 0346	6	ffffff ffffff ffffff fe07f0 03ffff fe0fff	5.819072	3.683638
0409 0344	6	ffffff ffffff ffffff fe03f0 00ffff ffffff	5.819513	3.674736
0410 0344	6	ffffff ffffff ffffff fe03e0 00ffff ffffff	5.819952	3.665874
0411 0344	4	ffffff ffffff ffffff f803e0 00ffff ffe0ff	5.815535	3.664997
0412 0343	4	ffffff ffffff ffffff f803f8 007fff fff01f	5.811139	3.664086
0413 0343	5	ffffff ffffff ffffff fc03fc 01ffff ffc000	5.809179	3.656821
0414 0343	7	ffffff ffffff ffffff ff03fe 07ffff ffc000	5.812049	3.651415
0415 0343	6	ffffff ffffff ffffff fe03ff ffffff ffc000	5.812501	3.642720
0416 0343	5	ffffff ffffff ffffff fc07ff 7fffff ff8000	5.810553	3.635561
0417 0342	4	ffffff ffffff ffffff f807e0 07ffff ffe000	5.806221	3.634687
0418 0341	5	ffffff ffffff ffffff fc03f0 03ffff fff800	5.804297	3.627558
0419 0340	6	ffffff ffffff ffffff fe03f0 00ffff ffff80	5.804763	3.619014
0420 0341	5	ffffff ffffff ffffff fc07c0 01ffff ffff80	5.802851	3.611952
0421 0339	6	ffffff ffffff ffffff fe03f0 00ffff ffffc0	5.803319	3.603484
0422 0339	5	ffffff ffffff ffffff fc03f8 01ffff ffff80	5.801419	3.596491
0423 0339	7	ffffff ffffff ffffff ff01fe 07ffff ffffc0	5.804246	3.591392
0424 0339	7	ffffff ffffff ffffff ff01ff 1fffff ffffc0	5.807059	3.586301
0425 0340	6	ffffff ffffff ffffff fe03ff bfffff fffe00	5.807512	3.577969
0426 0340	6	ffffff ffffff ffffff fe03fe 07ffff fffe00	5.807963	3.569679
0427 0340	6	ffffff ffffff ffffff fe03f8 03ffff fff800	5.808412	3.561423
0428 0340	7	ffffff ffffff ffffff ff03f8 01ffff fff000	5.811189	3.556424
0429 0340	6	ffffff ffffff ffffff fe03e0 01ffff ffe000	5.811628	3.548234
0430 0340	6	ffffff ffffff ffffff fe03e0 01ffff ff8000	5.812066	3.540080
0431 0340	4	ffffff ffffff ffffff f803f8 03ffff ff8000	5.807871	3.539472
0432 0340	5	ffffff ffffff ffffff fc03f8 03ffff fe0000	5.806005	3.532802
0433 0341	5	ffffff ffffff ffffff fc07fc 1fffff fc0000	5.804148	3.526156

0434 0341	5	ffffff ffffff ffffff fc07ff ffffff fc0000	5.802299	3.519533
0435 0341	5	ffffff ffffff ffffff fc07ff e1ffff fc0000	5.800459	3.512933
0436 0341	5	ffffff ffffff ffffff fc01ff c07fff f00000	5.798627	3.506356
0437 0343	5	ffffff ffffff ffffff fc07fc 007fff 000000	5.796804	3.499805
0438 0344	5	ffffff ffffff ffffff fc0ff8 00fffe 000000	5.794989	3.493274
0439 0344	4	ffffff ffffff ffffff f80ff8 00fffe 000000	5.790910	3.492639
0440 0344	4	ffffff ffffff ffffff f80ff8 00fffe 000000	5.786849	3.491979
0441 0344	3	ffffff ffffff ffffff f007fe 01fff8 0003a8	5.780543	3.501613
0442 0345	3	ffffff ffffff ffffff f00ffc 07fff8 001ffd	5.774267	3.511123
0443 0345	4	ffffff ffffff ffffff f807ff f7fffc 001fff	5.770271	3.510287
0444 0345	5	ffffff ffffff ffffff fc07ff fc1ffc 0007ff	5.768540	3.503729
0445 0346	4	ffffff ffffff ffffff f800ff f001fc 001fff	5.764575	3.502869
0446 0346	5	ffffff ffffff ffffff fc003f f001fe 003fff	5.762864	3.496337
0447 0347	8	ffffff ffffff ffffff ff807f c001fc 0007ff	5.767858	3.499680
0448 0347	8	ffffff ffffff ffffff ff807f c001fe 0501ff	5.772829	3.502955
0449 0349	5	ffffff ffffff ffffff fc01ff 801ffc 001fff	5.771111	3.496495
0450 0350	4	ffffff ffffff ffffff f803ff 80fff8 003fff	5.767184	3.495685
0451 0351	5	ffffff ffffff ffffff fc01ff fc07fc 005fff	5.765487	3.489254
0452 0352	4	ffffff ffffff ffffff f800ff fe003e 083ff8	5.761590	3.488416
0453 0352	8	ffffff ffffff ffffff ff803f fe000f e03f80	5.766520	3.491742
0454 0352	9	ffffff ffffff ffffff ffc01f fe000f e0bf00	5.773627	3.506992
0455 0355	8	ffffff ffffff ffffff ff807f f801ff c007ff	5.778509	3.510150
0456 0357	6	ffffff ffffff ffffff fe01ff f00fff c07fff	5.778994	3.502578
0457 0358	6	ffffff ffffff ffffff fe003f f83fff e0a002	5.779476	3.495036
0458 0358	7	ffffff ffffff ffffff ff0007 ffffff e0000a	5.782135	3.490657
0459 0356	15	ffffff ffffff ffffff ffff01 fe3fff e0022a	5.802174	3.667387
0460 0356	15	ffffff ffffff ffffff ffff01 f80fff c00faa	5.822126	3.842550
0461 0357	9	ffffff ffffff ffffff ffc007 c007ff 57c104	5.829004	3.856045
0462 0359	7	ffffff ffffff ffffff ff007f 001ff0 1fc000	5.831533	3.850670
0463 0359	7	ffffff ffffff ffffff ff01ff 001fc0 1ffc00	5.834052	3.845308
0464 0358	8	ffffff ffffff ffffff ff80ff 800f80 0fff80	5.838709	3.847107
0465 0356	9	ffffff ffffff ffffff ffc0ff f00e00 03fffe	5.845493	3.860249
0466 0355	9	ffffff ffffff ffffff ffc07f f07f00 07ffff	5.852248	3.873246
0467 0354	9	ffffff ffffff ffffff ffc07f e3ff80 03ffff	5.858974	3.886100
0468 0353	7	ffffff ffffff ffffff ff007f 01ffc0 01ffff	5.861407	3.880585
0469 0352	6	ffffff ffffff ffffff fe01f8 003f80 0fffff	5.861701	3.872370
0470 0352	6	ffffff ffffff ffffff fe03f8 003fc0 0fffff	5.861995	3.864189
0471 0352	4	ffffff ffffff ffffff f807f8 007f80 0fffef	5.858050	3.863330
0472 0351	5	ffffff ffffff ffffff fc07fe 007f80 17ffc7	5.856236	3.856712
0473 0349	9	ffffff ffffff ffffff ffc1ff c01fc0 07ffc1	5.862869	3.869384
0474 0349	9	ffffff ffffff ffffff ffc1ff c1ff80 03ff00	5.869473	3.881912
0475 0347	6	ffffff ffffff ffffff fe007f ffffe0 000000	5.869748	3.873791
0476 0347	5	ffffff ffffff ffffff fc01ff 07ffe0 000000	5.867924	3.867255
0477 0347	5	ffffff ffffff ffffff fc07f0 07ffc0 010000	5.866108	3.860737
0478 0346	5	ffffff ffffff ffffff fc07f0 03ffe0 002800	5.864300	3.854241
0479 0344	6	ffffff ffffff ffffff fe03f8 007ffe 000800	5.864583	3.846248
0480 0344	6	ffffff ffffff ffffff fe03f8 00fffe 000000	5.864864	3.838290
0481 0344	4	ffffff ffffff ffffff f807fc 00ffff 000000	5.860995	3.837527
0482 0344	4	ffffff ffffff ffffff f807fe 00ffff 800000	5.857142	3.836735
0483 0344	6	ffffff ffffff ffffff fe07ff 8fffff c00008	5.857438	3.828849
0484 0344	6	ffffff ffffff ffffff fe07ff ffffff c00000	5.857731	3.820998
0485 0343	5	ffffff ffffff ffffff fc07ff 1fffff f00000	5.855967	3.814646
0486 0342	5	ffffff ffffff ffffff fc03fc 03ffff fc0000	5.854209	3.808316
0487 0342	6	ffffff ffffff ffffff fe07f0 03ffff fe0000	5.854508	3.800554

0488 0342	6	ffffff ffffff ffffff fe0fe0 03ffff ff0000	5.854805	3.792824
0489 0340	6	ffffff ffffff ffffff fe03f8 00ffff fff800	5.855102	3.785130
0490 0340	4	ffffff ffffff ffffff f803f8 00ffff fffe00	5.851324	3.784412
0491 0340	4	ffffff ffffff ffffff f803fe 03ffff ffff00	5.847561	3.783670
0492 0340	6	ffffff ffffff ffffff fe03fe 03ffff ffff80	5.847870	3.776043
0493 0341	6	ffffff ffffff ffffff fe07ff ffffff ffff00	5.848178	3.768446
0494 0341	6	ffffff ffffff ffffff fe07ff dfffff ffff00	5.848485	3.760881
0495 0341	7	ffffff ffffff ffffff ff03fe 03ffff ffff80	5.850806	3.755966
0496 0341	7	ffffff ffffff ffffff ff03f8 01ffff ffff00	5.853118	3.751060
0497 0341	6	ffffff ffffff ffffff fe07f0 01ffff fffc00	5.853414	3.743569
0498 0341	6	ffffff ffffff ffffff fe07f0 01ffff fff800	5.853707	3.736109
0499 0341	5	ffffff ffffff ffffff fc07f0 03ffff fff000	5.852000	3.730093
0500 0341	4	ffffff ffffff ffffff f807f8 07ffff fff000	5.848303	3.729477
0501 0341	5	ffffff ffffff ffffff fc07fc 07ffff fff000	5.846613	3.723480
0502 0341	6	ffffff ffffff ffffff fe07ff dfffff fff801	5.846919	3.716120
0503 0341	6	ffffff ffffff ffffff fe03ff f1ffff fffc1f	5.847222	3.708791
0504 0342	6	ffffff ffffff ffffff fe03ff 80ffff fffbff	5.847525	3.701494
0505 0342	6	ffffff ffffff ffffff fe01fe 007fff ffffff	5.847826	3.694224
0506 0343	6	ffffff ffffff ffffff fe03fc 007fff fff1ff	5.848126	3.686985
0507 0343	7	ffffff ffffff ffffff ff01f8 007fff ffc07f	5.850393	3.682333
0508 0343	6	ffffff ffffff ffffff fe01fc 007fff ffc01f	5.850687	3.675147
0509 0344	3	ffffff ffffff ffffff f003fc 03ffff fe000f	5.845098	3.683841
0510 0344	4	ffffff ffffff ffffff f803fe 03ffff fe000f	5.841487	3.683278
0511 0345	3	ffffff ffffff ffffff f007ff ffffff fc0007	5.835937	3.691824
0512 0345	3	ffffff ffffff ffffff f003ff fc07ff f00007	5.830409	3.700273
0513 0345	5	ffffff ffffff ffffff fc001f f003ff f40000	5.828794	3.694413
0514 0346	4	ffffff ffffff ffffff f8003f e003ff e00000	5.825243	3.693720
0515 0347	8	ffffff ffffff ffffff ff807f c007ff c7fe00	5.829457	3.695708
0516 0347	8	ffffff ffffff ffffff ff807f 8007ff c7ff1f	5.833656	3.697650
0517 0348	6	ffffff ffffff ffffff fe00ff 803fff fffeff	5.833977	3.690568
0518 0349	5	ffffff ffffff ffffff fc01ff c07fff ffffff	5.832370	3.684795
0519 0350	6	ffffff ffffff ffffff fe003f fe000f ffffff	5.832692	3.677761
0520 0350	6	ffffff ffffff ffffff fe001f ff0007 ff9ffe	5.833013	3.670760
0521 0352	8	ffffff ffffff ffffff ff803f fc000f 80ff80	5.837164	3.672705
0522 0352	8	ffffff ffffff ffffff ff801f fe000f 80fe00	5.841300	3.674609
0523 0355	8	ffffff ffffff ffffff ff807f f001f9 fc1ffc	5.845420	3.676472
0524 0356	7	ffffff ffffff ffffff ff007f f003ef f87fff	5.847619	3.672005
0525 0358	6	ffffff ffffff ffffff fe0007 fcffff f3ffe0	5.847909	3.665066
0526 0359	6	ffffff ffffff ffffff fe0007 ffffff c15f44	5.848197	3.658156
0527 0361	9	ffffff ffffff ffffff ffc01e 01fc7e 100000	5.854167	3.670006
0528 0362	8	ffffff ffffff ffffff ff8020 00fc03 ff8000	5.858223	3.671754
0529 0361	5	ffffff ffffff ffffff fc01fc 001f01 fff000	5.856604	3.666215
0530 0359	7	ffffff ffffff ffffff ff007c 0007de 7fff00	5.858757	3.661766
0531 0357	8	ffffff ffffff ffffff ff807f 8007ff ffffc0	5.862782	3.663483
0532 0356	9	ffffff ffffff ffffff ffc03f e007ff fbffe0	5.868669	3.675040
0533 0355	7	ffffff ffffff ffffff ff003f f07fff fc1551	5.870787	3.670550
0534 0354	8	ffffff ffffff ffffff ff803f e3ffff fe0280	5.874767	3.672150
0535 0354	4	ffffff ffffff ffffff f80038 00ffef fa0028	5.871269	3.671844
0536 0354	4	ffffff ffffff ffffff f800e0 00ffe2 e80000	5.867785	3.671515
0537 0353	4	ffffff ffffff ffffff f807e0 007ff1 455504	5.864313	3.671162
0538 0353	3	ffffff ffffff ffffff f007f0 007ff1 1407c0	5.858998	3.679547
0539 0352	4	ffffff ffffff ffffff f803fc 00fff8 0007fe	5.855556	3.679121
0540 0351	7	ffffff ffffff ffffff ff01fe 007ffc 0147ff	5.857671	3.674738
0541 0351	6	ffffff ffffff ffffff fe01ff fffffc 001fff	5.857934	3.667994

0542 0350	7	fffff fffff fffff ff00ff fffffe 000fff	5.860037	3.663634
0543 0348	5	fffff fffff fffff fc03f8 0fffff e00fff	5.858456	3.658255
0544 0348	5	fffff fffff fffff fc03e0 07ffff fe3fff	5.856881	3.652891
0545 0346	6	fffff fffff fffff fe03f0 00ffff ffffff	5.857143	3.646237
0546 0345	7	fffff fffff fffff ff01f0 007fff ffffff	5.859233	3.641951
0547 0346	4	fffff fffff fffff f803e0 00ffff ff83ff	5.855840	3.641600
0548 0345	5	fffff fffff fffff fc01fc 00ffff ffc07f	5.854281	3.636298
0549 0345	7	fffff fffff fffff ff01ff 07ffff ff8003	5.856365	3.632066
0550 0345	7	fffff fffff fffff ff01ff dfffff ff8001	5.858440	3.627841
0551 0345	7	fffff fffff fffff ff01ff 5fffff ff0000	5.860508	3.623626
0552 0345	5	fffff fffff fffff fc01fc 07ffff ff0000	5.858953	3.618407
0553 0345	5	fffff fffff fffff fc03f0 07ffff ff8000	5.857402	3.613207
0554 0344	6	fffff fffff fffff fe03f0 03ffff ffe000	5.857659	3.606732
0555 0344	6	fffff fffff fffff fe03e0 03ffff fff800	5.857915	3.600280
0556 0343	5	fffff fffff fffff fc01f0 01ffff fffe00	5.856375	3.595136
0557 0343	4	fffff fffff fffff f807fc 01ffff fffc00	5.853048	3.594858
0558 0343	6	fffff fffff fffff fe07fc 03ffff fffc00	5.853311	3.588468
0559 0343	7	fffff fffff fffff ff03ff dfffff fffc00	5.855358	3.584406
0560 0344	6	fffff fffff fffff fe07ff ffffff fff800	5.855616	3.578051
0561 0344	6	fffff fffff fffff fe03fc 0fffff ffe000	5.855873	3.571719
0562 0344	6	fffff fffff fffff fe03f8 03ffff ffe000	5.856129	3.565412
0563 0344	6	fffff fffff fffff fe0ff0 03ffff ff8000	5.856384	3.559128
0564 0344	5	fffff fffff fffff fc0fe0 03ffff ff0000	5.854868	3.554127
0565 0344	2	fffff fffff fffff e00fe0 03ffff fe0000	5.848058	3.574055
0566 0344	2	fffff fffff fffff e01fe0 07ffff fc0000	5.841271	3.593822
0567 0344	4	fffff fffff fffff f81ff0 0fffff f80000	5.838029	3.593450
0568 0344	4	fffff fffff fffff f81ff8 0fffff e00000	5.834799	3.593060
0569 0344	4	fffff fffff fffff f80fff ffffff e00000	5.831580	3.592656
0570 0344	4	fffff fffff fffff f80fff ffffff e00000	5.828372	3.592231
0571 0345	3	fffff fffff fffff f007fc 01ffff 800000	5.823428	3.599912
0572 0345	5	fffff fffff fffff fc07fc 01ffff 000000	5.821990	3.594814
0573 0346	4	fffff fffff fffff f81ff0 03fff8 000000	5.818816	3.594324
0574 0346	3	fffff fffff fffff f01ff0 03fff8 000000	5.813914	3.601867
0575 0345	3	fffff fffff fffff f01ff8 01fffc 000015	5.809029	3.609335
0576 0345	3	fffff fffff fffff f00ffc 03fffc 000000	5.804160	3.616733
0577 0345	4	fffff fffff fffff f80ffe 07fff0 001700	5.801039	3.616096
0578 0345	4	fffff fffff fffff f80fff 07fff8 001ff4	5.797928	3.615442
0579 0346	4	fffff fffff fffff f80fff fffff8 083fff	5.794828	3.614773
0580 0346	4	fffff fffff fffff f801ff f83ff8 083fff	5.791739	3.614089
0581 0347	7	fffff fffff fffff ff00ff e007f8 007fff	5.793815	3.610383
0582 0347	9	fffff fffff fffff ffc0ff c007fc 007fff	5.799314	3.621793
0583 0349	6	fffff fffff fffff fe01ff 000ff0 001fff	5.799658	3.615657
0584 0349	5	fffff fffff fffff fc03ff 001ff0 0007ff	5.798291	3.610570
0585 0351	3	fffff fffff fffff f007ff 00fff0 007fff	5.793516	3.617746
0586 0352	4	fffff fffff fffff f80fff 83ffe0 00ffff	5.790461	3.617054
0587 0352	4	fffff fffff fffff f800ff f80fe0 00fffe	5.787416	3.616345
0588 0353	3	fffff fffff fffff f001ff f80170 01fff0	5.782683	3.623376
0589 0353	7	fffff fffff fffff ff007f f8007f 15ff00	5.784746	3.619742
0590 0354	7	fffff fffff fffff ff00ff f000ff abf820	5.786802	3.616114
0591 0357	5	fffff fffff fffff fc03ff c00fff 003fff	5.785473	3.611050
0592 0357	5	fffff fffff fffff fc01ff c01fff 81ffff	5.784149	3.606002
0593 0355	9	fffff fffff fffff ffc001 ff1fff f81d01	5.789563	3.617312
0594 0355	11	fffff fffff fffff fff001 ffffff fc1d01	5.798320	3.656781
0595 0355	15	fffff fffff fffff ffff01 fc1fff f00401	5.813759	3.792474

0596 0355	15	ffffff ffffff ffffff ffff01 f007ff c00740	5.829146	3.927234
0597 0356	9	ffffff ffffff ffffff ffc00f c007fe 2fc280	5.834449	3.937452
0598 0358	7	ffffff ffffff ffffff ff00fe 000ff8 0f8000	5.836395	3.933143
0599 0357	8	ffffff ffffff ffffff ff807f 0007c0 0ffe00	5.840001	3.934378
0600 0356	8	ffffff ffffff ffffff ff80ff 800380 07ff80	5.843595	3.935582
0601 0355	8	ffffff ffffff ffffff ff807f f01f00 07fffc	5.847177	3.936759
0602 0354	8	ffffff ffffff ffffff ff80ff e0bf80 03ffff	5.850747	3.937904
0603 0353	5	ffffff ffffff ffffff fc007e 01ff00 01ffff	5.849338	3.932579
0604 0352	6	ffffff ffffff ffffff fe0078 00ff80 03ffff	5.849587	3.926120
0605 0351	7	ffffff ffffff ffffff ff03f0 003fc0 07ffff	5.851485	3.921822
0606 0350	6	ffffff ffffff ffffff fe03f8 003fe0 03ffff	5.851730	3.915396
0607 0349	7	ffffff ffffff ffffff ff01ff 001fc0 07ffff	5.853619	3.911121
0608 0348	9	ffffff ffffff ffffff ffc0ff 801fe0 03fffb	5.858785	3.920929
0609 0348	8	ffffff ffffff ffffff ff80ff e0ffc0 0bffea	5.862295	3.922009
0610 0348	8	ffffff ffffff ffffff ff80ff ebff80 83ffea	5.865794	3.923057
0611 0347	5	ffffff ffffff ffffff fc01ff dfffc0 141d55	5.864379	3.917871
0612 0346	6	ffffff ffffff ffffff fe03fe 07ffe0 000a20	5.864600	3.911511
0613 0345	7	ffffff ffffff ffffff ff07f0 01ffe0 010000	5.866449	3.907236
0614 0345	5	ffffff ffffff ffffff fc07f0 01ffe0 015414	5.865040	3.902102
0615 0344	6	ffffff ffffff ffffff fe07f0 00fff8 002000	5.865259	3.895801
0616 0343	5	ffffff ffffff ffffff fc07f8 007ffc 000100	5.863857	3.890696
0617 0343	5	ffffff ffffff ffffff fc07fc 01ffff 000000	5.862459	3.885604
0618 0344	5	ffffff ffffff ffffff fc0ffc 03fffe 000002	5.861066	3.880527
0619 0343	6	ffffff ffffff ffffff fe07ff dfffff c00055	5.861290	3.874299
0620 0343	5	ffffff ffffff ffffff fc07ff ffffff c00000	5.859903	3.869256
0621 0342	4	ffffff ffffff ffffff f807f8 0fffff e00000	5.856913	3.868587
0622 0342	4	ffffff ffffff ffffff f80ff0 0fffff f00000	5.853932	3.867906
0623 0341	5	ffffff ffffff ffffff fc07e0 07ffff fc0000	5.852563	3.862874
0624 0341	5	ffffff ffffff ffffff fc0fc0 07ffff fc0000	5.851199	3.857856
0625 0340	4	ffffff ffffff ffffff f80fe0 03ffff ffe000	5.848242	3.857161
0626 0340	3	ffffff ffffff ffffff f00ff0 03ffff fff800	5.843699	3.863931
0627 0340	5	ffffff ffffff ffffff fc0ff8 0fffff fffc00	5.842356	3.858907
0628 0340	5	ffffff ffffff ffffff fc0ffc 3fffff fffe00	5.841016	3.853899
0629 0340	6	ffffff ffffff ffffff fe07ff bfffff fffe00	5.841269	3.847825

curve 1: average offset is 5.841269 pixels from edge.
maximum offset is 15
curve 2

0000 0461	-2	ffffff ffffff 000007 ffffff ffffff ffffff	-2.000000	0.000000
0001 0462	-1	ffe0ff fffffc 000003 ffffff ffffff ffffff	-1.500000	0.250000
0002 0464	-2	ff80ff ffffe0 000007 ffffff ffffff ffffff	-1.666667	0.222222
0003 0467	-2	f8007f fffc00 000007 ffffff ffffff ffffff	-1.750000	0.187500
0004 0468	-1	f0003f fff800 000003 ffffff ffffff ffffff	-1.600000	0.240000
0005 0471	1	0001ff ff0001 400000 ffffff ffffff ffffff	-1.166667	1.138889
0006 0472	2	0000ff fc0002 000000 7fffff ffffff ffffff	-0.714286	2.204082
0007 0478	3	0003ff 00800a 000fe0 3fffff ffffff ffffff	-0.250000	3.437500
0008 0482	1	000ff0 0a8000 003f00 ffffff ffffff ffffff	-0.111111	3.209877
0009 0488	-3	08b000 02a800 00800f ffffff ffffff ffffff	-0.400000	3.640000
0010 0490	-4	000020 002800 00001f ffffff ffffff ffffff	-0.727273	4.380166
0011 0490	0	280000 02a800 000001 ffffff ffffff ffffff	-0.666667	4.055555
0012 0490	1	e80000 bff800 000000 ffffff ffffff ffffff	-0.538462	3.940828
0013 0492	0	000020 3fe000 000001 ffffff ffffff ffffff	-0.500000	3.678571
0014 0492	1	a00020 3ff800 0a2800 ffffff ffffff ffffff	-0.400000	3.573333
0015 0495	-2	400147 fff000 050007 ffffff ffffff ffffff	-0.500000	3.500000
0016 0495	-2	c00007 fff000 150007 ffffff ffffff ffffff	-0.588235	3.418685

0017 0496	-2	000007 ffe00a 280007 ffffff ffffff ffffff	-0.666667	3.333333
0018 0496	-1	000003 ffe028 000003 ffffff ffffff ffffff	-0.684211	3.163435
0019 0498	-2	80000f ff8000 00a007 ffffff ffffff ffffff	-0.750000	3.087500
0020 0498	-2	80003f fe0000 2a8007 ffffff ffffff ffffff	-0.809524	3.011338
0021 0497	-2	50005f ff007f f40007 ffffff ffffff ffffff	-0.863636	2.935951
0022 0496	-1	a0028f ff82ff fe0003 ffffff ffffff ffffff	-0.869565	2.809074
0023 0496	-2	e802bf fe0fff f80007 ffffff ffffff ffffff	-0.916667	2.743056
0024 0496	-2	fa00ff f80fff f82807 ffffff ffffff ffffff	-0.960000	2.678400
0025 0496	-2	fe03fe 00ffff eaa007 ffffff ffffff ffffff	-1.000000	2.615385
0026 0496	-1	fe0008 00ffff e82003 ffffff ffffff ffffff	-1.000000	2.518519
0027 0496	-1	ff8000 007fff aa8283 ffffff ffffff ffffff	-1.000000	2.428572
0028 0496	-1	ff8000 003ffe 000fe3 ffffff ffffff ffffff	-1.000000	2.344828
0029 0496	-1	ff8000 000020 003ffb ffffff ffffff ffffff	-1.000000	2.266667
0030 0496	-1	ff8080 000000 000fe3 ffffff ffffff ffffff	-1.000000	2.193549
0031 0497	-2	ff05c1 41f000 000007 ffffff ffffff ffffff	-1.031250	2.155274
0032 0498	-2	fe2802 07f000 000007 ffffff ffffff ffffff	-1.060606	2.117539
0033 0499	-2	f00000 7fc000 000007 ffffff ffffff ffffff	-1.088235	2.080450
0034 0501	-2	c00007 fc0000 07c007 ffffff ffffff ffffff	-1.114286	2.044082
0035 0503	-2	00017f c00001 fe0007 ffffff ffffff ffffff	-1.138889	2.008488
0036 0504	-3	200bff 80000f e0000f ffffff ffffff ffffff	-1.189189	2.045289
0037 0507	-3	007ff0 0003f0 00000f ffffff ffffff ffffff	-1.236842	2.075485
0038 0508	-3	03ffe0 000fe0 00000f ffffff ffffff ffffff	-1.282051	2.099935
0039 0510	-3	aff800 01fc00 00000f ffffff ffffff ffffff	-1.325000	2.119375
0040 0511	-3	7fc000 07f001 ff000f ffffff ffffff ffffff	-1.365854	2.134444
0041 0513	-2	000001 ffc01f ffc007 ffffff ffffff ffffff	-1.380952	2.092971
0042 0513	-2	000003 ffc01f fff807 ffffff ffffff ffffff	-1.395349	2.053002
0043 0514	-2	38003f f800ff affe07 ffffff ffffff ffffff	-1.409091	2.014463
0044 0514	-1	fc00ff a000ff afff03 ffffff ffffff ffffff	-1.400000	1.973334
0045 0517	-3	c1fff4 0007fd 5ffc0f ffffff ffffff ffffff	-1.434783	1.984878
0046 0517	-2	007fd0 0003fc 1ffc07 ffffff ffffff ffffff	-1.446809	1.949299
0047 0515	2	0001f4 00007f ffffc0 7fffff ffffff ffffff	-1.375000	2.151042
0048 0514	3	8000fe 80003f ffffe0 3fffff ffffff ffffff	-1.285714	2.489797
0049 0513	2	f80007 f0000f ffffc0 7fffff ffffff ffffff	-1.220000	2.651600
0050 0513	2	ff0007 f00005 07ffc0 7fffff ffffff ffffff	-1.156863	2.798924
0051 0515	0	ff0007 ff0000 0fff01 ffffff ffffff ffffff	-1.134616	2.770341
0052 0515	-1	ff8001 ff8000 1ffe03 ffffff ffffff ffffff	-1.132076	2.718405
0053 0515	-2	ffc000 ffc000 7ff007 ffffff ffffff ffffff	-1.148148	2.681756
0054 0515	-2	7fc000 7fd001 fff007 ffffff ffffff ffffff	-1.163637	2.645950
0055 0513	-1	07ff00 01ff41 fff803 ffffff ffffff ffffff	-1.160715	2.599171
0056 0513	-2	05ff80 007f07 fff807 ffffff ffffff ffffff	-1.175439	2.565713
0057 0512	-1	003fe0 00003f ffa003 ffffff ffffff ffffff	-1.172414	2.521998
0058 0511	0	005fff 00001f ff0001 ffffff ffffff ffffff	-1.152542	2.502155
0059 0511	-2	40401f 00001f f00007 ffffff ffffff ffffff	-1.166667	2.472222
0060 0510	-2	a00002 e00000 000007 ffffff ffffff ffffff	-1.180328	2.442892
0061 0508	-5	020000 3e0000 00003f ffffff ffffff ffffff	-1.241935	2.635016
0062 0507	-4	000150 040000 00001f ffffff ffffff ffffff	-1.285714	2.712019
0063 0503	-5	c00000 0001c0 00003f ffffff ffffff ffffff	-1.343750	2.881836
0064 0501	-3	f00000 000150 00000f ffffff ffffff ffffff	-1.369231	2.879054
0065 0500	-3	f80000 003e80 003f8f ffffff ffffff ffffff	-1.393939	2.875115
0066 0499	-2	f40000 007ff0 001fe7 ffffff ffffff ffffff	-1.402985	2.837604
0067 0498	-1	fa0000 00fffe 0003e3 ffffff ffffff ffffff	-1.397059	2.798227
0068 0498	0	fa00a0 03fffe 000081 ffffff ffffff ffffff	-1.376812	2.785550
0069 0498	0	f803ff 803fff e00001 ffffff ffffff ffffff	-1.357143	2.772449
0070 0497	0	f405ff f807ff fc0001 ffffff ffffff ffffff	-1.338028	2.758977

0071 0497	0	c0057f ff05ff fc5001 ffffff ffffff ffffff	-1.319444	2.745178
0072 0497	0	80057f ff01ff fc0001 ffffff ffffff ffffff	-1.301370	2.731095
0073 0500	-3	80203f fe02bf 80000f ffffff ffffff ffffff	-1.324324	2.732652
0074 0500	-3	00003f fe002a 00000f ffffff ffffff ffffff	-1.346667	2.733156
0075 0501	-2	00007f fc0155 400007 ffffff ffffff ffffff	-1.355263	2.702736
0076 0501	-2	0001ff fc0005 004007 ffffff ffffff ffffff	-1.363636	2.672964
0077 0499	-2	00501f ff0000 004007 ffffff ffffff ffffff	-1.371795	2.643820
0078 0498	-1	80000f ff8000 822003 ffffff ffffff ffffff	-1.367089	2.612082
0079 0497	0	000017 ff4000 005001 ffffff ffffff ffffff	-1.350000	2.602501
0080 0497	-1	00001f fd4000 001003 ffffff ffffff ffffff	-1.345679	2.571865
0081 0497	-1	00555f fd0000 000003 ffffff ffffff ffffff	-1.341463	2.541940
0082 0495	0	000575 7d0000 000001 ffffff ffffff ffffff	-1.325301	2.532734
0083 0495	-2	000500 001400 000007 ffffff ffffff ffffff	-1.333333	2.507937
0084 0494	-3	000000 80aa00 00000f ffffff ffffff ffffff	-1.352941	2.510727
0085 0491	-4	07c005 700000 1fc01f ffffff ffffff ffffff	-1.383721	2.562061
0086 0488	-3	03fc02 800000 0ff80f ffffff ffffff ffffff	-1.402299	2.562294
0087 0484	-7	01ffe0 002800 0020ff ffffff ffffff ffffff	-1.465909	2.885202
0088 0479	-4	001fff 800000 00001f ffffff ffffff ffffff	-1.494382	2.924127
0089 0471	-2	80007f ffe000 000007 ffffff ffffff ffffff	-1.500000	2.894445
0090 0469	-1	c0001f fffc00 000003 ffffff ffffff ffffff	-1.494506	2.865355
0091 0467	-2	f8007f ffff80 000007 ffffff ffffff ffffff	-1.500000	2.836957
0092 0465	-2	ff007f fffff0 000007 ffffff ffffff ffffff	-1.505376	2.809112
0093 0463	-1	ffc3ff ffffff 000003 ffffff ffffff ffffff	-1.500000	2.781916
0094 0462	-1	fffbff fffffe 000003 ffffff ffffff ffffff	-1.494737	2.755236
0095 0460	-1	ffffff fffffe 3fe003 ffffff ffffff ffffff	-1.489583	2.729059
0096 0459	-2	fffc3f fffffe 3fe007 ffffff ffffff ffffff	-1.494845	2.703583
0097 0457	-1	fffc07 ffffff 3fe003 ffffff ffffff ffffff	-1.489796	2.678469
0098 0456	-1	fffe00 fffff8 3ff003 ffffff ffffff ffffff	-1.484848	2.653812
0099 0454	-1	fffe00 0ffe02 bff803 ffffff ffffff ffffff	-1.480000	2.629601
0100 0454	-2	fffe00 07fc03 fff007 ffffff ffffff ffffff	-1.485148	2.606217
0101 0452	-1	ffff80 00fc00 3ff803 ffffff ffffff ffffff	-1.480392	2.582951
0102 0451	0	1fffd0 507000 1ff801 ffffff ffffff ffffff	-1.466019	2.578944
0103 0450	-1	0fff80 020000 0ff803 ffffff ffffff ffffff	-1.461538	2.556215
0104 0450	-1	0fff80 e80288 0ff803 ffffff ffffff ffffff	-1.457143	2.533879
0105 0450	-3	07ffef fe0ea0 0fe00f ffffff ffffff ffffff	-1.471698	2.532220
0106 0449	-3	07fff7 ff1ff0 0fe00f ffffff ffffff ffffff	-1.485981	2.530179
0107 0449	-4	1fffff ff7ff0 07001f ffffff ffffff ffffff	-1.509259	2.564731
0108 0448	-4	8fffff fffffc 00001f ffffff ffffff ffffff	-1.532110	2.597594
0109 0445	-2	ff0007 ffdfff 010007 ffffff ffffff ffffff	-1.536364	2.575952
0110 0444	-2	ff8003 ff9ffe 03a007 ffffff ffffff ffffff	-1.540540	2.554665
0111 0444	-1	ff8003 f03fe0 2a0003 ffffff ffffff ffffff	-1.535714	2.534441
0112 0444	-1	ffc003 f08f80 220003 ffffff ffffff ffffff	-1.530973	2.514529
0113 0444	-1	ffe00f efe03f f80003 ffffff ffffff ffffff	-1.526316	2.494923
0114 0444	-2	ffe00f ffe0ff fa0007 ffffff ffffff ffffff	-1.530435	2.475162
0115 0444	-1	03ffff fffbff e80003 ffffff ffffff ffffff	-1.525862	2.456229
0116 0444	-1	01ffff fff8ff ea0003 ffffff ffffff ffffff	-1.521367	2.437579
0117 0444	-2	00f03f efe000 000007 ffffff ffffff ffffff	-1.525424	2.418846
0118 0444	-3	008007 e00ff8 00a00f ffffff ffffff ffffff	-1.537815	2.416638
0119 0445	-2	1f8007 f07fff 000007 ffffff ffffff ffffff	-1.541667	2.398265
0120 0445	-2	7f0003 f47fff 000007 ffffff ffffff ffffff	-1.545455	2.380167
0121 0445	0	7f0007 ffffff 007001 ffffff ffffff ffffff	-1.532787	2.380074
0122 0446	-1	ff003f ffffff 00f803 ffffff ffffff ffffff	-1.528455	2.363013
0123 0448	-3	fe0fff ffbff8 03f00f ffffff ffffff ffffff	-1.540323	2.361279
0124 0449	-3	f1ffff fe0750 07f00f ffffff ffffff ffffff	-1.552000	2.359298

0125 0450	-1	03ffcf fa0020 0ff803 ffffff ffffff ffffff	-1.547619	2.342972
0126 0451	-2	03ff97 d00000 1ff007 ffffff ffffff ffffff	-1.551181	2.326122
0127 0452	-3	07fe00 2e0000 3fe00f ffffff ffffff ffffff	-1.562500	2.324220
0128 0453	-3	0ffc51 547c00 7fe00f ffffff ffffff ffffff	-1.573643	2.322097
0129 0455	-2	7ff000 03fc00 7fe007 ffffff ffffff ffffff	-1.576923	2.305622
0130 0455	0	7ff000 07ff40 7ff001 ffffff ffffff ffffff	-1.564885	2.306860
0131 0457	-1	ffe000 7ffffc 3fc003 ffffff ffffff ffffff	-1.560606	2.291783
0132 0458	-1	ffe000 fffffe 3fe003 ffffff ffffff ffffff	-1.556391	2.276897
0133 0460	-3	ff800f fffffe 0fe00f ffffff ffffff ffffff	-1.567164	2.275341
0134 0461	-3	ffc07f ffffff 00000f ffffff ffffff ffffff	-1.577778	2.273582
0135 0463	-2	ffffff ffffff 000007 ffffff ffffff ffffff	-1.580882	2.258165
0136 0463	-2	ffffff ffffff 000007 ffffff ffffff ffffff	-1.583941	2.242955
0137 0464	-1	ffe0ff fffffc 000003 ffffff ffffff ffffff	-1.579710	2.229155
0138 0466	-2	ff83ff ffffe0 000007 ffffff ffffff ffffff	-1.582734	2.214380
0139 0469	-2	f0007f fffc00 000007 ffffff ffffff ffffff	-1.585714	2.199797
0140 0470	-1	e0007f fff800 000003 ffffff ffffff ffffff	-1.581560	2.186611
0141 0473	2	0001ff ff0000 000000 7fffff ffffff ffffff	-1.556338	2.260912
0142 0474	3	0001ff fc0000 000020 3fffff ffffff ffffff	-1.524475	2.389262
0143 0484	0	003ff0 000000 00fe01 ffffff ffffff ffffff	-1.513889	2.388697
0144 0485	0	001fc0 040000 007e01 ffffff ffffff ffffff	-1.503448	2.387921
0145 0489	-2	041000 001500 000007 ffffff ffffff ffffff	-1.506849	2.373242
0146 0492	-3	200000 002a00 00000f ffffff ffffff ffffff	-1.517007	2.372161
0147 0494	-2	a00000 0a8000 000007 ffffff ffffff ffffff	-1.520270	2.357699
0148 0494	-1	000000 ffa000 000003 ffffff ffffff ffffff	-1.516778	2.343680
0149 0495	0	000000 7ff000 000001 ffffff ffffff ffffff	-1.506667	2.343291
0150 0495	0	c00000 7ff400 000001 ffffff ffffff ffffff	-1.496689	2.342706
0151 0496	-1	000000 fff800 000003 ffffff ffffff ffffff	-1.493421	2.328906
0152 0497	-2	000007 fff000 000007 ffffff ffffff ffffff	-1.496732	2.315351
0153 0499	-2	000007 ffc014 400007 ffffff ffffff ffffff	-1.500000	2.301950
0154 0499	-2	000003 ffc005 400007 ffffff ffffff ffffff	-1.503226	2.288701
0155 0500	-1	a0003f ff8000 2a0003 ffffff ffffff ffffff	-1.500000	2.275643
0156 0500	-2	80003f ff000f fa0007 ffffff ffffff ffffff	-1.503185	2.262731
0157 0498	-1	0002af ffc0ff fe8003 ffffff ffffff ffffff	-1.500000	2.250002
0158 0498	-1	a802af ff83ff fe0003 ffffff ffffff ffffff	-1.496855	2.237413
0159 0498	-2	e800ff fe0fff f80007 ffffff ffffff ffffff	-1.500000	2.225002
0160 0498	-2	fa02ff e03fff f80007 ffffff ffffff ffffff	-1.503106	2.212725
0161 0498	-1	fe00fe 00ffff e00003 ffffff ffffff ffffff	-1.500000	2.200619
0162 0498	-1	fe0000 00ffff a00003 ffffff ffffff ffffff	-1.496932	2.188643
0163 0498	-1	ff8000 003ffe a003e3 ffffff ffffff ffffff	-1.493902	2.176794
0164 0498	-1	ff8038 000ffe 0007fb ffffff ffffff ffffff	-1.490909	2.165070
0165 0498	-1	ff80a0 000000 001ffb ffffff ffffff ffffff	-1.487952	2.153470
0166 0498	-1	ff80a8 008000 000be3 ffffff ffffff ffffff	-1.485030	2.141992
0167 0499	-1	ff0000 01fc00 000003 ffffff ffffff ffffff	-1.482143	2.130634
0168 0500	-1	fe0800 aff800 000003 ffffff ffffff ffffff	-1.479290	2.119395
0169 0501	0	f00001 ffc000 000001 ffffff ffffff ffffff	-1.470588	2.119724
0170 0503	-2	c01407 fc0000 03c007 ffffff ffffff ffffff	-1.473684	2.108957
0171 0505	-2	0001ff c00001 fc0007 ffffff ffffff ffffff	-1.476744	2.098297
0172 0507	-3	0007ff 00000f c0000f ffffff ffffff ffffff	-1.485549	2.099502
0173 0509	-2	107ff0 0001f8 000007 ffffff ffffff ffffff	-1.488506	2.088948
0174 0510	-1	23ffe0 0007e0 000003 ffffff ffffff ffffff	-1.485714	2.078368
0175 0512	-3	aff800 00fe00 00000f ffffff ffffff ffffff	-1.494318	2.079514
0176 0513	-2	ffc000 07f800 ff4007 ffffff ffffff ffffff	-1.497175	2.069201
0177 0515	-1	000001 ffc00f fff003 ffffff ffffff ffffff	-1.494382	2.058957
0178 0516	-1	000003 ff803f fff803 ffffff ffffff ffffff	-1.491620	2.048813

0179 0518	-3	f000ff e000ff aff80f ffffff ffffff ffffff	-1.500000	2.050000
0180 0518	-3	f80ffe a003fe affe0f ffffff ffffff ffffff	-1.508287	2.051036
0181 0520	-3	00ffea 000ffa bff80f ffffff ffffff ffffff	-1.516484	2.051926
0182 0520	-2	003faa 0003fe bffc07 ffffff ffffff ffffff	-1.519126	2.041984
0183 0518	1	0000ea a000ff ffff80 ffffff ffffff ffffff	-1.505435	2.065188
0184 0518	1	c000be a000ff ffff80 ffffff ffffff ffffff	-1.491892	2.087772
0185 0516	2	fc000f e0000f ffffc0 7fffff ffffff ffffff	-1.473118	2.141750
0186 0516	1	ff0003 fa0000 03ff80 ffffff ffffff ffffff	-1.459893	2.162830
0187 0517	0	ff8000 ffc000 0fff01 ffffff ffffff ffffff	-1.452128	2.162602
0188 0517	0	ffc000 7fc000 1ffe01 ffffff ffffff ffffff	-1.444445	2.162257
0189 0517	-1	7fc000 3fc000 7ff003 ffffff ffffff ffffff	-1.442106	2.151911
0190 0517	-2	5ff000 1ff000 7ff007 ffffff ffffff ffffff	-1.445026	2.142265
0191 0515	-1	01ff00 003f01 fffc03 ffffff ffffff ffffff	-1.442708	2.132134
0192 0515	-2	007fc0 001f07 fffc07 ffffff ffffff ffffff	-1.445596	2.122688
0193 0514	-1	00fff8 00001f ff8003 ffffff ffffff ffffff	-1.443299	2.112764
0194 0513	0	01557f 80001f ff0001 ffffff ffffff ffffff	-1.435897	2.112558
0195 0512	-1	a80003 e00002 a00003 ffffff ffffff ffffff	-1.433674	2.102744
0196 0511	-2	400000 1c0000 000007 ffffff ffffff ffffff	-1.436548	2.093690
0197 0509	-4	000405 000000 00001f ffffff ffffff ffffff	-1.449495	2.116136
0198 0507	-4	004000 000000 00001f ffffff ffffff ffffff	-1.462312	2.138027
0199 0502	-1	f80000 000028 000003 ffffff ffffff ffffff	-1.460000	2.128400
0200 0500	1	fe0000 000000 000000 ffffff ffffff ffffff	-1.447761	2.147769
0201 0499	1	ff0000 000ffc 0007f8 ffffff ffffff ffffff	-1.435644	2.166650
0202 0499	0	fc0000 001ffc 0007f9 ffffff ffffff ffffff	-1.428571	2.166080
0203 0499	1	f4007c 00ffff 000040 ffffff ffffff ffffff	-1.416667	2.184232
0204 0499	1	fc01ff 007fff c00000 ffffff ffffff ffffff	-1.404878	2.201927
0205 0499	1	f401ff f807ff f01400 ffffff ffffff ffffff	-1.393204	2.219177
0206 0499	0	d0017f fc05ff fc1f01 ffffff ffffff ffffff	-1.386474	2.217787
0207 0499	0	00005f ff01ff fc0001 ffffff ffffff ffffff	-1.379808	2.216322
0208 0499	0	00001f ff8017 fc0001 ffffff ffffff ffffff	-1.373206	2.214784
0209 0501	-2	00005f ff0000 000007 ffffff ffffff ffffff	-1.376191	2.206099
0210 0501	-2	00005f ff0000 000007 ffffff ffffff ffffff	-1.379147	2.197479
0211 0503	-2	0001ff f40000 004007 ffffff ffffff ffffff	-1.382076	2.188923
0212 0503	-2	0001ff fd0000 000007 ffffff ffffff ffffff	-1.384977	2.180431
0213 0502	-3	00003f fe0000 08800f ffffff ffffff ffffff	-1.392523	2.182374
0214 0501	-2	00001f fd0000 040007 ffffff ffffff ffffff	-1.395349	2.173932
0215 0498	-1	00002f fea800 002003 ffffff ffffff ffffff	-1.393519	2.164587
0216 0498	-1	00202f fea800 000003 ffffff ffffff ffffff	-1.391705	2.155323
0217 0496	0	8002a2 faa800 000001 ffffff ffffff ffffff	-1.385321	2.154280
0218 0497	-2	000515 155000 000007 ffffff ffffff ffffff	-1.388128	2.146160
0219 0494	-3	000002 000000 00000f ffffff ffffff ffffff	-1.395455	2.148161
0220 0493	-3	078000 400000 1c000f ffffff ffffff ffffff	-1.402715	2.150038
0221 0489	-2	01fc01 000000 07f807 ffffff ffffff ffffff	-1.405406	2.141953
0222 0487	-6	01ff00 500000 01f07f ffffff ffffff ffffff	-1.426009	2.226588
0223 0479	-4	001fff c00000 00001f ffffff ffffff ffffff	-1.437500	2.246094
0224 0475	-2	0001ff ff0000 000007 ffffff ffffff ffffff	-1.440000	2.237510
0225 0469	0	f0001f fffe00 000001 ffffff ffffff ffffff	-1.433628	2.236745
0226 0467	0	fc0007 ffffc0 000001 ffffff ffffff ffffff	-1.427313	2.235906
0227 0466	-1	ff003f fffff8 000003 ffffff ffffff ffffff	-1.425439	2.226897
0228 0465	0	ff80ff fffffe 000001 ffffff ffffff ffffff	-1.419214	2.226006
0229 0463	0	fff7ff ffffff 000001 ffffff ffffff ffffff	-1.413043	2.225047
0230 0463	-2	ffffff ffffff 050007 ffffff ffffff ffffff	-1.415584	2.216900
0231 0461	-2	fffd7f fffffe 1fe007 ffffff ffffff ffffff	-1.418103	2.208810
0232 0460	-1	fffc1f fffffe 3fe003 ffffff ffffff ffffff	-1.416309	2.200077

0233 0458	-1	fffc01 ffffe0 3fe003 ffffff ffffff ffffff	-1.414530	2.191413
0234 0457	-2	fffc00 7ffff00 7ff007 ffffff ffffff ffffff	-1.417021	2.183540
0235 0455	-2	fffe00 07fc01 fff007 ffffff ffffff ffffff	-1.419492	2.175721
0236 0454	-1	ffff00 01fc00 3ff803 ffffff ffffff ffffff	-1.417722	2.167281
0237 0453	-1	ffff00 005000 1ff003 ffffff ffffff ffffff	-1.415966	2.158905
0238 0452	-1	3fff80 000000 0ff803 ffffff ffffff ffffff	-1.414226	2.150593
0239 0451	-2	0fffc1 500100 0ff807 ffffff ffffff ffffff	-1.416667	2.143056
0240 0450	-1	03ffe3 fe0080 0ff803 ffffff ffffff ffffff	-1.414938	2.134881
0241 0448	-1	00fffb ff83f8 03f803 ffffff ffffff ffffff	-1.413223	2.126767
0242 0448	-1	03ffff ff8ffe 03e003 ffffff ffffff ffffff	-1.411523	2.118715
0243 0447	-2	03ffff ffdfff 000007 ffffff ffffff ffffff	-1.413934	2.111445
0244 0446	-2	f8ffff ffffff 800007 ffffff ffffff ffffff	-1.416327	2.104223
0245 0444	-1	ff8003 ef8fff 020003 ffffff ffffff ffffff	-1.414634	2.096371
0246 0444	-1	ff8001 f81ffe 000003 ffffff ffffff ffffff	-1.412955	2.088577
0247 0444	-1	ff8003 f80f80 028003 ffffff ffffff ffffff	-1.411290	2.080840
0248 0444	-1	ffc003 e3e003 e00003 ffffff ffffff ffffff	-1.409639	2.073160
0249 0444	-1	ffe00f efe0ff f80003 ffffff ffffff ffffff	-1.408000	2.065536
0250 0444	-1	ffe00f fff1ff f80003 ffffff ffffff ffffff	-1.406374	2.057967
0251 0444	-1	03ffff fff8ff e00003 ffffff ffffff ffffff	-1.404762	2.050453
0252 0444	-1	03feff fff000 200003 ffffff ffffff ffffff	-1.403162	2.042994
0253 0444	-2	03e00f e38fe8 000007 ffffff ffffff ffffff	-1.405512	2.036348
0254 0444	-2	038007 e03ffe 000007 ffffff ffffff ffffff	-1.407843	2.029742
0255 0445	-2	7f0007 f47fff 000007 ffffff ffffff ffffff	-1.410156	2.023178
0256 0445	-1	7f0007 ffffff 000003 ffffff ffffff ffffff	-1.408560	2.015958
0257 0446	-1	fe001f ffbfff 00f003 ffffff ffffff ffffff	-1.406977	2.008789
0258 0447	-2	fc007f ffdfff 01f007 ffffff ffffff ffffff	-1.409266	2.002385
0259 0449	-4	f07fff fe0551 07e01f ffffff ffffff ffffff	-1.419231	2.020400
0260 0450	-4	ebffff fc0000 0fe01f ffffff ffffff ffffff	-1.429119	2.038080
0261 0450	-3	03ffaf e000a8 0ff00f ffffff ffffff ffffff	-1.435115	2.039683
0262 0451	-3	07ffc7 d40050 1ff00f ffffff ffffff ffffff	-1.441065	2.041204
0263 0451	-2	07ff14 551000 1ff007 ffffff ffffff ffffff	-1.443182	2.034651
0264 0452	-3	0ffe00 a0fe00 3ff00f ffffff ffffff ffffff	-1.449057	2.036085
0265 0454	-3	fff800 23fe00 3fe00f ffffff ffffff ffffff	-1.454887	2.037439
0266 0454	-1	fff800 07ffe0 3ff003 ffffff ffffff ffffff	-1.453183	2.030581
0267 0456	-1	fff802 7ffffe 3fe003 ffffff ffffff ffffff	-1.451492	2.023767
0268 0457	-2	fff001 ffffff 1fc007 ffffff ffffff ffffff	-1.453532	2.017358
0269 0459	-3	ffe01f ffffff 07c00f ffffff ffffff ffffff	-1.459259	2.018712
0270 0460	-3	fff83f ffffff 02800f ffffff ffffff ffffff	-1.464945	2.019990
0271 0462	-3	ffffff ffffff 00000f ffffff ffffff ffffff	-1.470588	2.021194
0272 0462	-2	ffffff ffffff 800007 ffffff ffffff ffffff	-1.472528	2.014813
0273 0463	-2	fff1ff ffffc 000007 ffffff ffffff ffffff	-1.474453	2.008472
0274 0464	-1	ffe0ff fffff0 000003 ffffff ffffff ffffff	-1.472727	2.001984
0275 0467	-2	fc007f fffe00 000007 ffffff ffffff ffffff	-1.474638	1.995733
0276 0468	-1	f8003f fffc00 000003 ffffff ffffff ffffff	-1.472924	1.989339
0277 0471	1	80007f ff8000 000000 ffffff ffffff ffffff	-1.464029	2.004102
0278 0473	2	0001ff fc0005 000000 7fffff ffffff ffffff	-1.451613	2.039773
0279 0475	7	00007f f00000 0001fc 03ffff ffffff ffffff	-1.421429	2.286684
0280 0482	1	0003f8 080000 003f80 ffffff ffffff ffffff	-1.412811	2.299338
0281 0488	-3	003c02 0b8000 03e00f ffffff ffffff ffffff	-1.418440	2.300086
0282 0490	-3	080000 082800 00000f ffffff ffffff ffffff	-1.424028	2.300766
0283 0494	-3	a00000 00a000 00000f ffffff ffffff ffffff	-1.429577	2.301379
0284 0494	-3	a00000 3e8000 00000f ffffff ffffff ffffff	-1.435088	2.301927
0285 0493	0	000000 7fd000 000001 ffffff ffffff ffffff	-1.430070	2.301054
0286 0494	-1	a00280 ffe800 000003 ffffff ffffff ffffff	-1.428571	2.293679

0287 0494	-1	000002 fffa00 000003 ffffff ffffff ffffff	-1.427083	2.286350
0288 0495	-1	500007 fff000 001003 ffffff ffffff ffffff	-1.425606	2.279068
0289 0496	-1	000203 ffe002 000003 ffffff ffffff ffffff	-1.424138	2.271831
0290 0497	-2	000007 ffc01c 000007 ffffff ffffff ffffff	-1.426117	2.265160
0291 0499	-2	00005f ff0000 004007 ffffff ffffff ffffff	-1.428082	2.258527
0292 0500	-3	0000bf fe0002 a0000f ffffff ffffff ffffff	-1.433447	2.259223
0293 0499	-3	00107f ff007f f0400f ffffff ffffff ffffff	-1.438776	2.259857
0294 0498	-2	00003f ff00ff f80007 ffffff ffffff ffffff	-1.440678	2.253261
0295 0498	-3	8000ff f80fff f8000f ffffff ffffff ffffff	-1.445946	2.253835
0296 0498	-3	e803ff f00fff e0000f ffffff ffffff ffffff	-1.451178	2.254351
0297 0498	-2	fa03fe 00ffff e00807 ffffff ffffff ffffff	-1.453020	2.247793
0298 0498	-2	fa0020 03ffff 800007 ffffff ffffff ffffff	-1.454849	2.241273
0299 0498	-2	f80000 00fffe 000007 ffffff ffffff ffffff	-1.456667	2.234789
0300 0498	-1	fe0000 007ffe 000f83 ffffff ffffff ffffff	-1.455150	2.228055
0301 0498	-2	ff0020 0000a0 003fe7 ffffff ffffff ffffff	-1.456954	2.221658
0302 0498	-2	ff0080 000000 003fe7 ffffff ffffff ffffff	-1.458746	2.215296
0303 0499	-2	fe0000 05c000 000007 ffffff ffffff ffffff	-1.460526	2.208969
0304 0500	-3	fc0200 0fe000 00000f ffffff ffffff ffffff	-1.465574	2.209471
0305 0502	-3	e00a0a ff8000 00000f ffffff ffffff ffffff	-1.470588	2.209920
0306 0502	-3	80200f fe0000 00800f ffffff ffffff ffffff	-1.475570	2.210316
0307 0503	-2	00107f e00000 1fc007 ffffff ffffff ffffff	-1.477273	2.204030
0308 0504	-3	0002ff 800003 fe000f ffffff ffffff ffffff	-1.482200	2.204377
0309 0508	-5	a8fff8 0003fe 00003f ffffff ffffff ffffff	-1.493548	2.237056
0310 0508	-3	20ffe0 0003f8 00000f ffffff ffffff ffffff	-1.498392	2.237137
0311 0510	-3	0fff00 00ff80 00000f ffffff ffffff ffffff	-1.503205	2.237170
0312 0511	-3	5ff000 03fc00 00000f ffffff ffffff ffffff	-1.507987	2.237157
0313 0513	-3	ff0000 1fe007 ffc00f ffffff ffffff ffffff	-1.512739	2.237099
0314 0514	-3	000003 ffc03f ff800f ffffff ffffff ffffff	-1.517460	2.236997
0315 0515	-2	00003f f4007f fffc07 ffffff ffffff ffffff	-1.518987	2.230652
0316 0515	-2	f8007f d000ff fffe07 ffffff ffffff ffffff	-1.520505	2.224343
0317 0518	-3	e2ffe8 800ffa bff80f ffffff ffffff ffffff	-1.525157	2.224210
0318 0518	-3	00ffaa 8003fe bff80f ffffff ffffff ffffff	-1.529781	2.224035
0319 0517	0	0007d4 1001ff ffff01 ffffff ffffff ffffff	-1.525000	2.224375
0320 0516	1	0003fa 8800ff ffff80 ffffff ffffff ffffff	-1.517134	2.237246
0321 0514	3	f0000f 80001f ffffc0 3fffff ffffff ffffff	-1.503106	2.293468
0322 0513	3	fc0007 f00007 ffffc0 3fffff ffffff ffffff	-1.489164	2.348954
0323 0514	1	ff8003 ff8000 03ff80 ffffff ffffff ffffff	-1.481482	2.360768
0324 0515	0	ff8000 ffc000 07fe01 ffffff ffffff ffffff	-1.476923	2.360237
0325 0515	0	ffc000 7fd000 3ffc01 ffffff ffffff ffffff	-1.472393	2.359667
0326 0515	-1	7fc000 3ff000 7ff003 ffffff ffffff ffffff	-1.470948	2.353131
0327 0514	-1	8ffe00 03fe00 7ff803 ffffff ffffff ffffff	-1.469512	2.346631
0328 0513	-1	01ff00 007fc1 fffc03 ffffff ffffff ffffff	-1.468085	2.340166
0329 0513	-2	007fc0 00001f fff007 ffffff ffffff ffffff	-1.469697	2.333930
0330 0512	-1	003fe0 00000f ff0003 ffffff ffffff ffffff	-1.468278	2.327543
0331 0511	-1	00017f 80001f fc0003 ffffff ffffff ffffff	-1.466867	2.321191
0332 0511	-2	40001f c00007 f00007 ffffff ffffff ffffff	-1.468468	2.315071
0333 0509	-2	000000 180000 000007 ffffff ffffff ffffff	-1.470060	2.308983
0334 0508	-4	000000 080000 00001f ffffff ffffff ffffff	-1.477612	2.321140
0335 0505	-6	000000 000000 00007f ffffff ffffff ffffff	-1.491071	2.374920
0336 0503	-4	c00000 000000 00001f ffffff ffffff ffffff	-1.498516	2.386496
0337 0501	-2	f00001 000400 000007 ffffff ffffff ffffff	-1.500000	2.380177
0338 0500	-1	f80000 003f80 003fc3 ffffff ffffff ffffff	-1.498525	2.373891
0339 0499	-2	f40000 00fff0 001ff7 ffffff ffffff ffffff	-1.500000	2.367646
0340 0499	-1	f40000 01fffc 0007c3 ffffff ffffff ffffff	-1.498534	2.361434

0341 0499	-1	f007fc 007fff c00003 ffffff ffffff ffffff	-1.497076	2.355254
0342 0499	-1	f003ff c01fff c00003 ffffff ffffff ffffff	-1.495627	2.349106
0343 0498	-1	80003f fe03ff f80003 ffffff ffffff ffffff	-1.494186	2.342989
0344 0498	-1	800a3f fe03ff f80003 ffffff ffffff ffffff	-1.492754	2.336904
0345 0499	-2	00005f ff015f f00007 ffffff ffffff ffffff	-1.494220	2.330891
0346 0500	-3	0002bf fe000f a0000f ffffff ffffff ffffff	-1.498559	2.330689
0347 0501	-2	00007f fe0010 000007 ffffff ffffff ffffff	-1.500000	2.324712
0348 0501	-2	00007f fc0154 400007 ffffff ffffff ffffff	-1.501433	2.318766
0349 0501	-2	0001ff fd0000 014007 ffffff ffffff ffffff	-1.502857	2.312849
0350 0500	-2	000a3f fe8000 008007 ffffff ffffff ffffff	-1.504273	2.306962
0351 0498	-1	8000af ffa000 000003 ffffff ffffff ffffff	-1.502841	2.301128
0352 0497	0	00401f ffd000 005001 ffffff ffffff ffffff	-1.498583	2.300989
0353 0498	-2	00a0bf fe8000 000007 ffffff ffffff ffffff	-1.500000	2.295197
0354 0496	-1	000a0f fea000 000003 ffffff ffffff ffffff	-1.498591	2.289434
0355 0495	-2	000554 005000 000007 ffffff ffffff ffffff	-1.500000	2.283707
0356 0494	-1	0000a8 000000 000003 ffffff ffffff ffffff	-1.498599	2.278009
0357 0491	-4	01c015 700000 1f801f ffffff ffffff ffffff	-1.505586	2.289075
0358 0489	-3	01fc05 540000 0ff00f ffffff ffffff ffffff	-1.509749	2.288902
0359 0485	-6	01ffc0 114000 01f07f ffffff ffffff ffffff	-1.522222	2.338395
0360 0482	-7	00fffe 000000 0000ff ffffff ffffff ffffff	-1.537396	2.414807
0361 0476	-5	0007ff fc0000 00003f ffffff ffffff ffffff	-1.546961	2.441165
0362 0472	-3	8000ff ffe000 00000f ffffff ffffff ffffff	-1.550964	2.440241
0363 0468	-3	e0003f ffff00 00000f ffffff ffffff ffffff	-1.554945	2.439289
0364 0467	-2	f8001f ffffc0 000007 ffffff ffffff ffffff	-1.556164	2.433147
0365 0465	-1	ff01ff fffffc 000003 ffffff ffffff ffffff	-1.554644	2.427342
0366 0463	0	ffe1ff ffffff 000001 ffffff ffffff ffffff	-1.550408	2.427296
0367 0462	-3	ffffff fffffe 0f800f ffffff ffffff ffffff	-1.554348	2.426394
0368 0461	-2	ffffff ffffff 1fc007 ffffff ffffff ffffff	-1.555555	2.420355
0369 0459	-1	fffc1f fffffe 1fe003 ffffff ffffff ffffff	-1.554054	2.414645
0370 0458	-1	fffc0f fffffe 3fe003 ffffff ffffff ffffff	-1.552560	2.408962
0371 0457	-2	fffc00 ffff00 7fe007 ffffff ffffff ffffff	-1.553763	2.403023
0372 0456	-3	fffc00 3ffc00 7ff00f ffffff ffffff ffffff	-1.557640	2.402174
0373 0454	-1	fffe00 03fc00 fff003 ffffff ffffff ffffff	-1.556149	2.396580
0374 0453	-2	ffff00 01fc00 3ff007 ffffff ffffff ffffff	-1.557333	2.390713
0375 0451	0	1fffd0 001000 0ffc01 ffffff ffffff ffffff	-1.553191	2.390788
0376 0450	-1	0fffa0 000000 0ffc03 ffffff ffffff ffffff	-1.551724	2.385256
0377 0450	-1	07ffc2 e800a0 0ff803 ffffff ffffff ffffff	-1.550264	2.379749
0378 0449	-1	03ffe5 ff0140 07f003 ffffff ffffff ffffff	-1.548812	2.374267
0379 0448	-1	03fffb ff8ff8 03f003 ffffff ffffff ffffff	-1.547368	2.368809
0380 0448	-3	03ffff ffbff8 03e00f ffffff ffffff ffffff	-1.551181	2.368116
0381 0447	-3	c3ffff ffffff 00000f ffffff ffffff ffffff	-1.554974	2.367397
0382 0446	-2	fc00ff ffffff 000007 ffffff ffffff ffffff	-1.556136	2.361732
0383 0444	-1	ffc003 ef8fff 000003 ffffff ffffff ffffff	-1.554687	2.356385
0384 0444	-1	ffc003 f01ffc 000003 ffffff ffffff ffffff	-1.553246	2.351061
0385 0444	-1	ffc003 f80f80 200003 ffffff ffffff ffffff	-1.551813	2.345761
0386 0444	-1	ffe00f e3e00f f80003 ffffff ffffff ffffff	-1.550387	2.340485
0387 0444	-1	ffe00f eff0ff fe0003 ffffff ffffff ffffff	-1.548969	2.335232
0388 0444	-1	7ff83f fff9ff f80003 ffffff ffffff ffffff	-1.547558	2.330001
0389 0444	-1	01ffff fff80b e20003 ffffff ffffff ffffff	-1.546154	2.324794
0390 0444	-1	01fc7f fff000 000003 ffffff ffffff ffffff	-1.544757	2.319609
0391 0444	-1	038007 e00ffe 000003 ffffff ffffff ffffff	-1.543367	2.314447
0392 0444	-1	03c003 f83fff 800003 ffffff ffffff ffffff	-1.541985	2.309307
0393 0445	-1	7f0007 f43fff 800003 ffffff ffffff ffffff	-1.540609	2.304189
0394 0445	0	7f0007 ffdfff c00001 ffffff ffffff ffffff	-1.536709	2.304349

0395 0446	-1	fe003f ffffff 80f803 ffffff ffffff ffffff	-1.535353	2.299256
0396 0447	-2	fe007f ffffff 01f807 ffffff ffffff ffffff	-1.536524	2.294007
0397 0449	-2	f1ffff ff0555 47f007 ffffff ffffff ffffff	-1.537688	2.288781
0398 0449	-2	e7ffff ff0005 43f807 ffffff ffffff ffffff	-1.538847	2.283579
0399 0450	-1	03ffcf f80008 0ff803 ffffff ffffff ffffff	-1.537500	2.278594
0400 0451	-2	07ff87 d00000 0ff007 ffffff ffffff ffffff	-1.538653	2.273443
0401 0452	-2	0ffe08 a2b800 3ff007 ffffff ffffff ffffff	-1.539801	2.268317
0402 0455	-4	7ff004 03f000 7fc01f ffffff ffffff ffffff	-1.545906	2.277669
0403 0455	-1	fff000 07ff00 7ff003 ffffff ffffff ffffff	-1.544554	2.272767
0404 0456	-1	ffe000 0ffff8 3fe003 ffffff ffffff ffffff	-1.543210	2.267886
0405 0458	-1	ffe008 fffffe 3fe003 ffffff ffffff ffffff	-1.541872	2.263025
0406 0459	-2	ffc101 fffffe 1fe007 ffffff ffffff ffffff	-1.542997	2.257979
0407 0460	-1	ffe00f ffffff 808003 ffffff ffffff ffffff	-1.541667	2.253165
0408 0461	-2	fff07f ffffff 000007 ffffff ffffff ffffff	-1.542787	2.248169
0409 0462	-1	ffffff ffffff 800003 ffffff ffffff ffffff	-1.541463	2.243402
0410 0462	1	ffffff ffffff 800000 ffffff ffffff ffffff	-1.535280	2.253621
0411 0465	-1	ffc1ff fffff0 000003 ffffff ffffff ffffff	-1.533980	2.248845
0412 0467	-2	ff01ff ffffc0 000007 ffffff ffffff ffffff	-1.535109	2.243924
0413 0473	-5	0003ff ffe000 00003f ffffff ffffff ffffff	-1.543478	2.267432
0414 0474	-3	0003ff ff8000 00000f ffffff ffffff ffffff	-1.546988	2.267068
0415 0477	-2	000fff f00010 400007 ffffff ffffff ffffff	-1.548077	2.262111
0416 0478	-1	000fff c00000 800383 ffffff ffffff ffffff	-1.546763	2.257405
0417 0484	0	000ff0 080000 003e01 ffffff ffffff ffffff	-1.543062	2.257714
0418 0486	-1	000fc0 200000 007e03 ffffff ffffff ffffff	-1.541766	2.253028
0419 0490	-2	020002 800000 000007 ffffff ffffff ffffff	-1.542857	2.248162
0420 0492	-3	080000 000080 00000f ffffff ffffff ffffff	-1.546318	2.247854
0421 0492	1	2a0000 2aa000 000000 ffffff ffffff ffffff	-1.540284	2.257855
0422 0493	0	500040 7ff000 000001 ffffff ffffff ffffff	-1.536643	2.258113
0423 0495	0	000004 1ff000 000001 ffffff ffffff ffffff	-1.533019	2.258343
0424 0495	0	d00005 5ff400 005401 ffffff ffffff ffffff	-1.529412	2.258547
0425 0495	0	140001 fffc00 001401 ffffff ffffff ffffff	-1.525821	2.258723
0426 0497	-1	000007 fff000 014003 ffffff ffffff ffffff	-1.524590	2.254079
0427 0498	-1	000203 ffe02a 0a0003 ffffff ffffff ffffff	-1.523364	2.249454
0428 0499	-2	000501 ffc054 540007 ffffff ffffff ffffff	-1.524475	2.244739
0429 0500	-1	00002f ff8000 000003 ffffff ffffff ffffff	-1.523256	2.240157
0430 0502	-4	00003f fc000f aa001f ffffff ffffff ffffff	-1.529002	2.249159
0431 0500	-3	0002bf ff02ff f8000f ffffff ffffff ffffff	-1.532407	2.248950
0432 0500	-3	0002bf fe03ff f8000f ffffff ffffff ffffff	-1.535797	2.248719
0433 0499	-2	c0017f fc1fff f00007 ffffff ffffff ffffff	-1.536866	2.244033
0434 0499	-2	f401ff c01fff f00007 ffffff ffffff ffffff	-1.537931	2.239366
0435 0498	-1	fe80be 00ffff e00003 ffffff ffffff ffffff	-1.536697	2.234892
0436 0498	-1	fe0000 00ffff 800003 ffffff ffffff ffffff	-1.535469	2.230435
0437 0498	-1	ff0000 003fff 8001e3 ffffff ffffff ffffff	-1.534246	2.225996
0438 0498	-1	ff8000 000ffe 0003f3 ffffff ffffff ffffff	-1.533029	2.221574
0439 0498	-1	ff8000 000000 001ff3 ffffff ffffff ffffff	-1.531818	2.217169
0440 0498	-1	ff80e0 200000 0003e3 ffffff ffffff ffffff	-1.530612	2.212781
0441 0500	-2	fe0202 03f800 000007 ffffff ffffff ffffff	-1.531674	2.208272
0442 0502	-3	f82028 0fe000 00000f ffffff ffffff ffffff	-1.534988	2.208143
0443 0503	-3	c00007 ff0000 01000f ffffff ffffff ffffff	-1.538288	2.207993
0444 0503	-2	c00007 fc0000 07c007 ffffff ffffff ffffff	-1.539326	2.203509
0445 0505	-2	00007f c00001 fe0007 ffffff ffffff ffffff	-1.540359	2.199043
0446 0507	-4	5005ff 00001f c0001f ffffff ffffff ffffff	-1.545861	2.207628
0447 0509	-3	007ff0 0001f0 00000f ffffff ffffff ffffff	-1.549107	2.207410
0448 0510	-3	03ffe0 0007e0 00000f ffffff ffffff ffffff	-1.552338	2.207171

0449 0512	-3	2ff800 01fc00 00000f ffffff ffffff ffffff	-1.555555	2.206914
0450 0513	-2	7fc000 07f001 ffc007 ffffff ffffff ffffff	-1.556541	2.202457
0451 0515	-2	000000 ffc01f fff007 ffffff ffffff ffffff	-1.557522	2.198019
0452 0516	-3	000003 ff803f fff80f ffffff ffffff ffffff	-1.560706	2.197749
0453 0517	-3	f8007f d000ff 07fc0f ffffff ffffff ffffff	-1.563877	2.197462
0454 0516	-1	7e00ff a800ff abff03 ffffff ffffff ffffff	-1.562637	2.193329
0455 0520	-3	00ffea 000ffa bff80f ffffff ffffff ffffff	-1.565789	2.193040
0456 0520	-3	007fa2 0003fe bff80f ffffff ffffff ffffff	-1.568928	2.192732
0457 0518	1	0003fa 8000ff ffff80 ffffff ffffff ffffff	-1.563319	2.202322
0458 0517	1	c0017f 54007f ffffc0 ffffff ffffff ffffff	-1.557734	2.211808
0459 0516	1	f8000f ea000f ffff80 ffffff ffffff ffffff	-1.552174	2.221191
0460 0516	1	fe0003 fe8000 03ff80 ffffff ffffff ffffff	-1.546638	2.230471
0461 0517	-1	ff8001 ff0000 0fff03 ffffff ffffff ffffff	-1.545454	2.226289
0462 0517	-1	ffc000 7fc000 1ffc03 ffffff ffffff ffffff	-1.544276	2.222122
0463 0517	-2	7fc000 7fc000 7fe007 ffffff ffffff ffffff	-1.545258	2.217779
0464 0516	-2	3ff800 1ff800 7ff007 ffffff ffffff ffffff	-1.546236	2.213454
0465 0514	-1	00ff80 003f81 fffc03 ffffff ffffff ffffff	-1.545064	2.209342
0466 0514	-1	0affe0 000f8f fff803 ffffff ffffff ffffff	-1.543897	2.205246
0467 0513	0	101ff0 00001f ff0001 ffffff ffffff ffffff	-1.540598	2.205616
0468 0512	-1	282fff 80000f ff0003 ffffff ffffff ffffff	-1.539446	2.201535
0469 0511	-1	540007 c00001 700003 ffffff ffffff ffffff	-1.538298	2.197469
0470 0510	-1	000000 380000 000003 ffffff ffffff ffffff	-1.537155	2.193417
0471 0508	-3	00a002 8a0000 00000f ffffff ffffff ffffff	-1.540254	2.193295
0472 0506	-3	800000 000000 00000f ffffff ffffff ffffff	-1.543340	2.193153
0473 0501	-1	fc0000 000014 000003 ffffff ffffff ffffff	-1.542194	2.189148
0474 0500	0	fe0000 000000 000001 ffffff ffffff ffffff	-1.538947	2.189536
0475 0499	0	ff0000 000ffc 0007f1 ffffff ffffff ffffff	-1.535714	2.189901
0476 0499	0	fe0000 001ffc 0007f9 ffffff ffffff ffffff	-1.532495	2.190244
0477 0499	0	fc0000 007fff 000041 ffffff ffffff ffffff	-1.529289	2.190565
0478 0499	0	fc01fc 007fff c00001 ffffff ffffff ffffff	-1.526096	2.190864
0479 0499	0	fc01ff f007ff f00001 ffffff ffffff ffffff	-1.522917	2.191141
0480 0499	0	f4007f fc07ff fc0001 ffffff ffffff ffffff	-1.519750	2.191398
0481 0499	-1	80007f ff01ff fd0003 ffffff ffffff ffffff	-1.518672	2.187411
0482 0499	0	00001f ff005f fd0001 ffffff ffffff ffffff	-1.515528	2.187647
0483 0501	-2	00001f ff0141 400007 ffffff ffffff ffffff	-1.516529	2.183611
0484 0501	-2	00005f ff0000 000007 ffffff ffffff ffffff	-1.517526	2.179590
0485 0502	-2	0000ff fa000a 02e007 ffffff ffffff ffffff	-1.518518	2.175583
0486 0502	-2	0000ff fa0000 028007 ffffff ffffff ffffff	-1.519507	2.171591
0487 0501	-2	00001f ff0000 004007 ffffff ffffff ffffff	-1.520492	2.167613
0488 0501	-3	00001f ff4000 00000f ffffff ffffff ffffff	-1.523517	2.167647
0489 0500	-3	00000f fa2000 00a00f ffffff ffffff ffffff	-1.526531	2.167663
0490 0500	-3	0080bf fea000 00800f ffffff ffffff ffffff	-1.529532	2.167661
0491 0498	-2	0000af faa000 000007 ffffff ffffff ffffff	-1.530488	2.163704
0492 0497	-2	000055 514000 000007 ffffff ffffff ffffff	-1.531440	2.159762
0493 0495	-2	100000 000000 000007 ffffff ffffff ffffff	-1.532389	2.155833
0494 0494	-4	000002 000000 08001f ffffff ffffff ffffff	-1.537374	2.163754
0495 0490	-3	00f800 000000 0ff00f ffffff ffffff ffffff	-1.540323	2.163696
0496 0489	-4	01fc01 000000 07f01f ffffff ffffff ffffff	-1.545272	2.171491
0497 0484	-7	00fff8 000000 0000ff ffffff ffffff ffffff	-1.556225	2.226758
0498 0479	-4	001fff f00000 00001f ffffff ffffff ffffff	-1.561122	2.234240
0499 0474	-5	0000ff ffc000 00003f ffffff ffffff ffffff	-1.568000	2.253376
0500 0472	-4	80007f fff800 00001f ffffff ffffff ffffff	-1.572855	2.260660
0501 0470	-4	e000ff ffff80 00001f ffffff ffffff ffffff	-1.577690	2.267869
0502 0468	-3	fc00ff fffff0 00000f ffffff ffffff ffffff	-1.580517	2.267374

0503 0467	-2	ff07ff fffffc 000007 ffffff ffffff ffffff	-1.581349	2.263223
0504 0466	-3	ffbfff fffffc 00000f ffffff ffffff ffffff	-1.584159	2.262719
0505 0463	-4	ffffff fffffc 3fc01f ffffff ffffff ffffff	-1.588933	2.269759
0506 0462	-3	fff8ff fffffc 3fc00f ffffff ffffff ffffff	-1.591716	2.269202
0507 0460	-3	fff80f fffffc 7fc00f ffffff ffffff ffffff	-1.594488	2.268631
0508 0459	-2	fffc01 fffff0 7fe007 ffffff ffffff ffffff	-1.595285	2.264496
0509 0457	-2	fffc00 1ffc00 7ff007 ffffff ffffff ffffff	-1.596079	2.260377
0510 0456	-2	fffe00 0ffc03 fff007 ffffff ffffff ffffff	-1.596869	2.256272
0511 0454	-1	ffff80 00f800 3ff803 ffffff ffffff ffffff	-1.595703	2.252560
0512 0453	0	7fff00 007800 1ff801 ffffff ffffff ffffff	-1.592593	2.253123
0513 0452	-1	1fffa0 000000 0ff803 ffffff ffffff ffffff	-1.591440	2.249421
0514 0452	-2	0fff80 000000 0ff807 ffffff ffffff ffffff	-1.592233	2.245377
0515 0451	-2	07ffc7 fe0000 07f007 ffffff ffffff ffffff	-1.593024	2.241347
0516 0450	-2	03ffe3 ff8fa0 03f007 ffffff ffffff ffffff	-1.593811	2.237332
0517 0449	-3	07ffff ffdff8 01c00f ffffff ffffff ffffff	-1.596525	2.236823
0518 0448	-3	03ffff fffffe 00000f ffffff ffffff ffffff	-1.599229	2.236301
0519 0446	-1	fe000f ffffff 000003 ffffff ffffff ffffff	-1.598077	2.232690
0520 0445	-1	ff8003 ffcfff 074003 ffffff ffffff ffffff	-1.596929	2.229089
0521 0444	1	ffc000 f80ff8 000000 ffffff ffffff ffffff	-1.591954	2.237714
0522 0444	1	ffe000 f80fe0 000000 ffffff ffffff ffffff	-1.586998	2.246257
0523 0445	0	ffc007 e7f07f f00001 ffffff ffffff ffffff	-1.583970	2.246767
0524 0445	-1	ffe007 dff0ff ff0003 ffffff ffffff ffffff	-1.582857	2.243136
0525 0445	0	03fe7f fff9ff f00001 ffffff ffffff ffffff	-1.579848	2.243625
0526 0445	-1	01ffff fff057 d10003 ffffff ffffff ffffff	-1.578748	2.240005
0527 0445	-2	01e01f c7e100 000007 ffffff ffffff ffffff	-1.579546	2.236098
0528 0445	-2	010007 e01ffc 000007 ffffff ffffff ffffff	-1.580340	2.232205
0529 0446	-2	3f8003 e03ffe 000007 ffffff ffffff ffffff	-1.581132	2.228325
0530 0446	-1	3e0003 ef3fff 800003 ffffff ffffff ffffff	-1.580038	2.224763
0531 0447	-1	fe001f ffffff 007003 ffffff ffffff ffffff	-1.578947	2.221213
0532 0448	-2	fe007f ffbffe 01f007 ffffff ffffff ffffff	-1.579737	2.217377
0533 0450	-3	f83fff fe0aa8 87e00f ffffff ffffff ffffff	-1.582397	2.216995
0534 0450	-3	e3ffff fe0280 03f00f ffffff ffffff ffffff	-1.585047	2.216600
0535 0450	-1	00ffef fa0028 03f803 ffffff ffffff ffffff	-1.583955	2.213102
0536 0451	-1	01ffc5 d00000 07f803 ffffff ffffff ffffff	-1.582868	2.209615
0537 0452	-1	03ff8a 2aa820 0ff803 ffffff ffffff ffffff	-1.581785	2.206137
0538 0454	-3	0ffe22 80f800 3fe00f ffffff ffffff ffffff	-1.584416	2.205769
0539 0455	0	7ffc00 03ff00 3ff001 ffffff ffffff ffffff	-1.581482	2.206325
0540 0456	-1	fff802 8ffff8 3fe003 ffffff ffffff ffffff	-1.580407	2.202870
0541 0458	-1	ffe000 fffffe 3fe003 ffffff ffffff ffffff	-1.579336	2.199426
0542 0459	-2	ffc001 fffffe 3fc007 ffffff ffffff ffffff	-1.580111	2.195700
0543 0460	-2	ffe00f ffffff 03e007 ffffff ffffff ffffff	-1.580883	2.191988
0544 0461	-2	fffc7f ffffff 800007 ffffff ffffff ffffff	-1.581652	2.188287
0545 0463	-2	ffffff ffffff 800007 ffffff ffffff ffffff	-1.582418	2.184600
0546 0463	0	ffffff ffffff 800001 ffffff ffffff ffffff	-1.579525	2.185175
0547 0465	-1	ffc1ff fffff8 000003 ffffff ffffff ffffff	-1.578467	2.181799
0548 0467	-2	ff01ff ffffc0 000007 ffffff ffffff ffffff	-1.579235	2.178148
0549 0469	0	f8003f fffe00 000001 ffffff ffffff ffffff	-1.576364	2.178715
0550 0470	1	f0003f fffc00 000000 ffffff ffffff ffffff	-1.571688	2.186785
0551 0474	1	0000ff fe0000 800000 ffffff ffffff ffffff	-1.567029	2.194782
0552 0475	2	0001ff fc0000 000040 7fffff ffffff ffffff	-1.560579	2.213781
0553 0482	2	000ffe 000000 001fc0 7fffff ffffff ffffff	-1.554152	2.232627
0554 0486	-1	003fe0 000000 007e03 ffffff ffffff ffffff	-1.553154	2.229157
0555 0490	-3	0e3000 0ba000 00800f ffffff ffffff ffffff	-1.555756	2.228906
0556 0492	-4	280000 002800 00001f ffffff ffffff ffffff	-1.560144	2.235611

0557 0492	0	2a0000 00a800 000001 ffffff ffffff ffffff	-1.557348	2.235959
0558 0492	1	280000 2ff800 000000 ffffff ffffff ffffff	-1.552773	2.243637
0559 0494	0	000202 aff800 000001 ffffff ffffff ffffff	-1.550000	2.243929
0560 0494	1	e80002 8ffa00 000a00 ffffff ffffff ffffff	-1.545455	2.251499
0561 0496	-1	a00000 fff800 002803 ffffff ffffff ffffff	-1.544484	2.248021
0562 0496	-1	000003 fff800 002803 ffffff ffffff ffffff	-1.543517	2.244554
0563 0498	-2	000003 ffe000 280007 ffffff ffffff ffffff	-1.544326	2.240944
0564 0499	-2	000001 ffc014 500007 ffffff ffffff ffffff	-1.545133	2.237345
0565 0499	-1	500017 ffc000 000003 ffffff ffffff ffffff	-1.544170	2.233916
0566 0500	-2	80003f ff800b eaa007 ffffff ffffff ffffff	-1.544974	2.230341
0567 0499	-2	50001f ff801f fd5007 ffffff ffffff ffffff	-1.545775	2.226779
0568 0498	-1	a000bf ff80ff fe0003 ffffff ffffff ffffff	-1.544816	2.223388
0569 0498	-1	ea00bf fe0fff fe0003 ffffff ffffff ffffff	-1.543860	2.220007
0570 0498	-1	fa00ff f80fff f80003 ffffff ffffff ffffff	-1.542907	2.216636
0571 0498	-1	fe80fe 00ffff e00803 ffffff ffffff ffffff	-1.541958	2.213275
0572 0498	-1	fe0000 00ffff 800003 ffffff ffffff ffffff	-1.541012	2.209924
0573 0498	-1	ff8000 003fff 820003 ffffff ffffff ffffff	-1.540070	2.206583
0574 0498	-1	ff8008 001ffe 0003e3 ffffff ffffff ffffff	-1.539131	2.203252
0575 0498	-1	ff8028 000000 000fe3 ffffff ffffff ffffff	-1.538195	2.199931
0576 0498	-1	ff80a0 000000 001fe3 ffffff ffffff ffffff	-1.537262	2.196619
0577 0499	-2	ff0000 01f000 000007 ffffff ffffff ffffff	-1.538062	2.193188
0578 0500	-3	fe0000 07f000 00000f ffffff ffffff ffffff	-1.540587	2.193085
0579 0502	-3	e00282 ff8000 00000f ffffff ffffff ffffff	-1.543104	2.192970
0580 0502	-3	e00803 fe0000 00000f ffffff ffffff ffffff	-1.545611	2.192842
0581 0503	-2	80017f f00000 07c007 ffffff ffffff ffffff	-1.546392	2.189429
0582 0504	-1	0a02ff e00001 fe0003 ffffff ffffff ffffff	-1.545455	2.186185
0583 0507	-2	001ffc 00007f 000007 ffffff ffffff ffffff	-1.546233	2.182795
0584 0508	-2	00fff8 0001f8 000007 ffffff ffffff ffffff	-1.547009	2.179415
0585 0511	-2	57fc00 007f00 000007 ffffff ffffff ffffff	-1.547782	2.176045
0586 0512	-3	bfe000 03f800 00000f ffffff ffffff ffffff	-1.550256	2.175925
0587 0514	-3	fe0002 3fe00f ff800f ffffff ffffff ffffff	-1.552721	2.175792
0588 0515	-3	000001 ffc01f ffc00f ffffff ffffff ffffff	-1.555179	2.175649
0589 0516	-3	20003f fa007f eff80f ffffff ffffff ffffff	-1.557627	2.175494
0590 0516	-3	fc007f f800ff affe0f ffffff ffffff ffffff	-1.560068	2.175327
0591 0516	-1	380fff a001ff abff03 ffffff ffffff ffffff	-1.559122	2.172181
0592 0519	-4	007fd4 0007fd 5ffc1f ffffff ffffff ffffff	-1.563238	2.178548
0593 0519	-1	000755 4001ff 7ffe03 ffffff ffffff ffffff	-1.562290	2.175413
0594 0519	-1	0007d5 4001ff ffff03 ffffff ffffff ffffff	-1.561345	2.172287
0595 0518	0	80003e 0000ff fffe01 ffffff ffffff ffffff	-1.558725	2.172726
0596 0517	0	f0001f c0001f ffff01 ffffff ffffff ffffff	-1.556114	2.173149
0597 0516	1	ff8003 fe0000 03ff80 ffffff ffffff ffffff	-1.551840	2.180423
0598 0516	0	ff8000 ff8000 0fff01 ffffff ffffff ffffff	-1.549249	2.180797
0599 0516	-1	ffe000 7fe000 3ffc03 ffffff ffffff ffffff	-1.548334	2.177665
0600 0516	-1	3fe000 3fe000 3ff803 ffffff ffffff ffffff	-1.547421	2.174541
0601 0516	-2	0ffe00 03fa00 fff007 ffffff ffffff ffffff	-1.548173	2.171268
0602 0515	-2	01ff80 007d07 fff807 ffffff ffffff ffffff	-1.548922	2.168005
0603 0515	-3	007fc0 00003f ffe00f ffffff ffffff ffffff	-1.551325	2.167896
0604 0514	-2	003ff8 00003f fe0007 ffffff ffffff ffffff	-1.552066	2.164645
0605 0512	-1	a0000f 80000f f80003 ffffff ffffff ffffff	-1.551155	2.161576
0606 0511	-2	000000 400000 000007 ffffff ffffff ffffff	-1.551895	2.158346
0607 0509	-4	000004 040000 00001f ffffff ffffff ffffff	-1.555921	2.164637
0608 0508	-3	000000 000000 00000f ffffff ffffff ffffff	-1.558292	2.164501
0609 0503	-3	e00000 000004 00000f ffffff ffffff ffffff	-1.560656	2.164355
0610 0501	-2	f80000 000001 000007 ffffff ffffff ffffff	-1.561375	2.161128

0611 0499	0	fc0000 000ff0 000501 ffffff ffffff ffffff	-1.558824	2.161574
0612 0499	0	fc0000 001ff0 001ff1 ffffff ffffff ffffff	-1.556281	2.162005
0613 0499	0	f40000 01fffd 0007f1 ffffff ffffff ffffff	-1.553746	2.162422
0614 0499	0	f40050 01ffff 0001e1 ffffff ffffff ffffff	-1.551220	2.162825
0615 0499	0	f407ff c01fff c00001 ffffff ffffff ffffff	-1.548701	2.163214
0616 0499	0	d001ff f017ff f00001 ffffff ffffff ffffff	-1.546191	2.163589
0617 0499	-2	00157f fc07ff fc0007 ffffff ffffff ffffff	-1.546926	2.160421
0618 0499	-2	00007f fe05ff ff5407 ffffff ffffff ffffff	-1.547658	2.157261
0619 0500	-3	20003f fe02af e0080f ffffff ffffff ffffff	-1.550000	2.157179
0620 0501	-4	00017f fc0014 00001f ffffff ffffff ffffff	-1.553945	2.163356
0621 0502	-3	0000ff f802aa 00000f ffffff ffffff ffffff	-1.556270	2.163234
0622 0501	-2	00007f f00015 05c007 ffffff ffffff ffffff	-1.556982	2.160077
0623 0500	-2	00003f fe0000 0a8007 ffffff ffffff ffffff	-1.557692	2.156929
0624 0500	-3	00003f fe0000 00800f ffffff ffffff ffffff	-1.560000	2.156801
0625 0498	-1	00000f fe8000 000003 ffffff ffffff ffffff	-1.559106	2.153856
0626 0498	-2	00003f fa0000 000007 ffffff ffffff ffffff	-1.559809	2.150730
0627 0496	-1	00000b fa0000 000003 ffffff ffffff ffffff	-1.558917	2.147804
0628 0496	-2	000002 e0a000 000007 ffffff ffffff ffffff	-1.559618	2.144698
0629 0493	0	100044 110000 000001 ffffff ffffff ffffff	-1.557143	2.145149

curve 2: average offset is -1.557143 pixels from edge.
maximum offset is 7

Appendix 9 Encoder Error Test

Test Data (every 0.0833mm a reading):

2	7	11	15	19	24	28	32	37	41
45	49	53	58	62	66	71	75	79	83
88	92	97	101	105	109	114	118	122	127
131	135	139	144	149	153	157	161	165	170
174	178	183	187	191	196	200	204	209	213
217	221	226	230	234	239	243	247	252	256
261	265	269	274	278	282	287	291	295	300
304	308	313	317	322	326	330	334	339	343
347	351	356	360	364	369	373	377	382	386
390	394	399	403	408	412	416	421	425	430
434	438	442	447	451	456	460	464	468	473
477	481	486	490	495	499	504	508	513	517
521	525	529	534	538	542	547	551	556	560
564	568	572	577	581	586	590	594	599	603
607	612	616	620	624	629	633	637	641	646
650	655	659	664	668	672	677	681	685	689
693	698	702	706	710	715	719	724	728	732
737	741	745	750	754	758	762	766	771	775
780	784	788	793	797	801	806	810	814	818
822	827	831	835	839	844	848	853	857	862
866	871	875	880	884	888	893	897	901	905
910	914	919	923	927	932	936	940	945	949
953	958	962	966	971	974	979	983	988	992
996	1001	1005	1009	1014	1019	1023	1027	1032	1036
1041	1045	1049	1054	1059	1063	1067	1071	1076	1080
1084	1089	1093	1097	1102	1106	1110	1114	1119	1123
1127	1132	1136	1140	1144	1149	1153	1157	1161	1166
1170	1174	1178	1183	1187	1191	1196	1200	1204	1209
1213	1218	1222	1227	1231	1236	1240	1244	1249	1253
1257	1261	1266	1270	1274	1279	1283	1288	1292	1296
1301	1305	1310	1314	1319	1323	1328	1332	1336	1340
1345	1349	1353	1357	1362	1366	1370	1375	1380	1384
1388	1392	1397	1401	1405	1409	1414	1418	1423	1427
1431	1436	1440	1444	1449	1453	1457	1462	1466	1470
1475	1479	1483	1488	1492	1496	1500	1505	1509	1514
1518	1522	1526	1531	1535	1540	1544	1548	1553	1557
1561	1566	1570	1575	1579	1583	1588	1592	1596	1601
1605	1609	1614	1618	1623	1627	1631	1635	1640	1644
1648	1653	1657	1661	1665	1670	1674	1678	1683	1687
1692	1696	1700	1705	1709	1713	1717	1721	1726	1730

1734	1739	1743	1747	1751	1756	1760	1765	1769	1773
1777	1781	1786	1790	1795	1800	1804	1808	1813	1817
1821	1826	1830	1835	1839	1843	1848	1852	1857	1861
1865	1869	1873	1878	1882	1886	1891	1895	1899	1904
1908	1912	1916	1920	1925	1929	1933	1937	1941	1945
1949	1954	1958	1963	1967	1972	1976	1981	1986	1990
1994	1999	2003	2008	2012	2017	2021	2026	2030	2034
2039	2043	2047	2052	2056	2061	2065	2069	2074	2078
2083	2087	2091	2096	2100	2104	2109	2113	2117	2122
2126	2131	2135	2140	2144	2148	2153	2157	2162	2166
2171	2175	2179	2183	2188	2192	2196	2200	2205	2209
2213	2217	2222	2227	2231	2235	2240	2244	2249	2253
2257	2261	2266	2270	2274	2279	2283	2287	2292	2296
2300	2304	2309	2313	2317	2321	2326	2331	2335	2340
2344	2348	2353	2357	2361	2366	2370	2375	2379	2383
2388	2392	2396	2400	2404	2409	2413	2417	2422	2426
2430	2435	2439	2443	2448	2452	2456	2460	2465	2469
2473	2477	2482	2486	2491	2495	2500	2504	2508	2513
2517	2521	2526	2530	2534	2539	2543	2547	2552	2556
2560	2565	2569	2573	2577	2582	2586	2591	2595	2600
2604	2608	2613	2617	2621	2626	2630	2635	2639	2643
2647	2652	2656	2661	2665	2669	2674	2678	2683	2687
2691	2696	2700	2704	2708	2713	2717	2722	2726	2730
2735	2739	2744	2748	2752	2757	2761	2765	2770	2774
2779	2783	2787	2792	2796	2800	2805	2809	2813	2817
2821	2825	2830	2834	2839	2843	2847	2852	2856	2861
2865	2869	2874	2878	2883	2887	2891	2896	2900	2905
2909	2914	2918	2922	2927	2931	2935	2940	2944	2948
2953	2957	2962	2966	2970	2974	2979	2983	2987	2991
2995	3000	3004	3008	3013	3017	3021	3026	3030	3034
3039	3043	3047	3051	3055	3060	3064	3068	3072	3077
3081	3086	3090	3094	3099	3104	3108	3113	3117	3122
3126	3131	3135	3140	3144	3148	3153	3157	3161	3166
3170	3174	3179	3183	3187	3192	3196	3200	3205	3209
3213	3218	3222	3226	3231	3235	3239	3244	3248	3253
3257	3261	3265	3270	3275	3279	3283	3288	3293	3297
3301	3306	3310	3314	3319	3323	3327	3331	3336	3340
3344	3349	3353	3358	3362	3366	3371	3375	3380	3384
3388	3393	3397	3401	3405	3410	3414	3418	3423	3427
3432	3436	3440	3445	3449	3453	3458	3462	3466	3471
3475	3480	3484	3488	3493	3497	3501	3506	3510	3514
3518	3523	3527	3531	3535	3540	3544	3549	3553	3557
3562	3566	3570	3575	3579	3583	3588	3592	3596	3600

3605	3609	3613	3618	3622	3626	3631	3635	3640	3644
3649	3653	3657	3661	3666	3670	3675	3679	3683	3688
3692	3696	3701	3705	3709	3714	3718	3722	3726	3731
3735	3740	3744	3748	3752	3757	3761	3765	3770	3774
3778	3783	3787	3791	3796	3800	3804	3809	3813	3818
3822	3826	3831	3835	3840	3844	3848	3853	3857	3861
3865	3869	3873	3877	3881	3886	3890	3894	3899	3903
3908	3912	3916	3921	3925	3930	3934	3938	3942	3946
3951	3955	3959	3964	3968	3972	3977	3981	3985	3989
3994	3998	4002	4007	4011	4015	4020	4024	4029	4033
4037	4042	4046	4050	4055	4059	4064	4068	4072	4077
4081	4085	4090	4094	4098	4103	4107	4111	4115	4120
4124	4129	4133	4137	4141	4146	4150	4154	4159	4163
4167	4171	4176	4180	4184	4189	4193	4197	4202	4206
4211	4215	4220	4224	4229	4233	4237	4242	4247	4251
4255	4260	4264	4269	4273	4277	4282	4286	4290	4295
4299	4303	4307	4312	4316	4320	4325	4329	4333	4337

Encoder Pulse Error 1

Pulses for every mm:

49	52	52	51	52	52	52	52	52	53
51	52	52	52	51	52	53	52	51	53
53	51	51	53	52	53	52	52	52	52
52	52	52	52	52	52	51	52	54	52
53	52	52	52	53	52	51	52	52	53
52	52	53	52	52	53	52	51	51	54
52	52	53	52	53	52	52	52	52	52
52	52	52	53	50	52	52	52	52	52
52	54	51							

frequency	1	1	11	53	14	3
pulses	49	50	51	52	53	54

$\bar{x} = 52.048$

$\sigma^2 = 0.6855$

$3\sigma = 2.48$

Encoder Pulse Error 2

Pulses for every 0.25mm:

11	13	13	12	13	13	13	13	13	13
12	14	12	13	13	13	13	13	13	13
13	13	13	13	14	12	13	13	13	13
13	13	13	13	13	13	13	13	14	13
12	13	14	12	13	13	13	13	13	13
13	13	13	13	12	14	13	13	12	13
13	13	13	13	12	14	13	14	13	12
14	13	13	13	13	12	13	13	14	13
13	14	13	13	13	12	13	13	13	13
12	13	13	14	13	13	13	13	13	13
14	13	13	13	13	13	13	13	13	13
13	13	13	13	13	13	13	13	13	13
13	13	13	13	14	12	13	13	13	13
13	13	13	13	12	14	12	13	14	13
13	13	14	12	13	13	13	12	13	12
13	14	14	13	13	14	13	13	13	13
13	13	13	14	13	13	14	12	13	13
13	13	14	12	13	13	13	13	13	14
13	13	13	13	12	13	13	13	13	13
13	13	13	13	13	13	13	13	13	14
13	13	13	13	13	13	13	13	13	13
14	13	13	13	13	13	12	13	13	14
13	13	13	14	13	13	13	13	13	12
13	13	13	13	12	13	13	13	14	14
13	13	13	13	13	13	13	13	13	14
12	14	14	13	13	12	13	14	13	13
13	13	13	13	13	13	13	13	13	13
13	13	13	13	13	13	13	13	13	13
13	13	13	13	13	13	13	13	13	13
13	13	13	13	14	13	12	12	13	13
13	14	12	13	13	13	13	13	13	13
13	13	13	13	13	13	13	13	13	13
13	13	13	13	14	13	13	14	13	13
12	13	13							

frequency	1	28	270	34
pulses	11	12	13	14

$\bar{x} = 13.012$

$\sigma^2 = 0.198$

$3\sigma = 1.335$