VISION BASED SYSTEMS FOR HARDNESS TESTING AND NDT

by

Ian Colin Smith, B.Eng., M.Sc.(Eng.).

Department of Electrical Engineering and Electronics.

October 1990.

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor of Philosophy.

for my late father, Colin Smith

The artist is the one who gives form to difficult visions.

Theodore Gill

ACKNOWLEDGEMENT

The work presented in this thesis was carried out at the University of Liverpool. I wish to express gratitude and appreciation to my colleagues and to the technical staff at the Department of Electrical Engineering and Electronics for their assistance. I am grateful to Prof. J. Lucas, who supervised this work, for his advise and support during my research. I would like to thank Dr. A.B. Parker for reading the drafts of this thesis. Finally, I would like to thank my family for their support.

ABSTRACT

The work presented in this thesis concerns the development of vision based systems for two hardness (destructive) tests, namely; the Shore and Vickers and a quality assurance non-destructive test. In each case the vision system is based on an IBM PC compatible computer fitted with a commercially available frame store. Bespoke image analysis software was written using the C language for each system.

In the Shore test, hardness is judged by the maximum rebound height attained by an indenter incident on a test sample. The purpose of the vision system is to measure the rebound height automatically. Laser light is used to illuminate the indenter and a vidicon vision camera is used to view its motion. Two approaches to the problem are considered; one in which image data is analysed in real time and one in which image data is merely stored in real time and analysed *a posteriori*. Non-real time analysis is shown to be superior to real time analysis in terms of accuracy and reliablity and its software implementation is discussed in detail.

The Vickers test uses the size of the permanent impression left by an indenter forced into the test material under a known load as a hardness index. In this case the purpose of the vision system is to measure the size of the indentation automatically. The original image analysis algorithms are shown to be capable of analysing good quality samples but are unreliable when applied to poor quality specimens. Further, fault-tolerant, algorithms are described to provide reliable and accurate results over wide variations in sample quality.

The quality assurance application involves automated visual inspection of novel ferrite components for defects. Each component is approximately 8 mm in diameter, annular in shape, and coated with aluminium. Laser light is used to illuminate individual components which are viewed using a charge-coupled device (CCD) video camera. Image analysis algorithms for characterising defects in component geometry and surface finish are discussed. The system is shown to capable of measuring component edge eccentricity and hole offset as well as providing a quantitative description of surface chips and cracks. The system is further shown to be capable of separately classifying surface defects extending to the edge of a component. Calculation of shape parameters for surface defects also provides a means of distinguishing cracks from surface chips.

CONTENTS

Page

Appendix Page

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The subject of this research is the development of vision-based sensors for use in automated destructive and non-destructive testing (NDT) applications. In this chapter an overview of NDT techniques is given with particular attention paid to automated welding applications.

The applications of NDT, although wide and varied, can be divided into three broad categories: detection of flaws within components (e.g. cracks), measurement of physical dimensions of components (e.g. thread size of a bolt) while destrcutive testing is concerned with measurement of material properties of components (e.g. hardness).

1.2 NDT METHODS

Historically there were considered to be five NDT methods [1]: radiography, ultrasonic,

magnetic, electrical and penetrant; although visual inspection, either aided or unaided, could be considered another method. Direct visual inspection, although a simple process, can detect a large proportion of defects appearing at the surface of a material which will cause immediate rejection of the component. The detection of volumetric defects, however, requires use of more sophisticated methods.

In radiography [2], ionising radiation is directed onto the material under test with either a photographic plate or electronic detector placed behind the sample. The incident radiation is scattered throughout the bulk of the material but passes unhindered through any cavities present in the sample. The amount of radiation passing through the back face of the material at a given point is dependant on the presence and size of defects so that an image of the defects is formed on the plate or detector. This method is particularly suited to detection of deep volumetric defects.

Deep volumetric defects may also be revealed using ultrasonic methods [3]. Here a pulse of ultrasonic energy is launched into the material from a transmitter placed on the surface of the component. The wavelength of the acoustic wave produced is made deliberately small so that any cavities present within the sample reflect the pulse. The reflected energy is detected by a receiver placed on the surface of the sample and it's output is usually displayed on an oscilloscope. The delay between transmitted and reflected pulses gives a measure of the depth of a defect.

Magnetic particle testing [4] involves applying a magnetic field through the surface of the specimen with a permanent or electro-magnet held against the metal, or from a heavy current flowing through the coils wrapped around the component. The magnetised component is then sprayed with magnetic ink made of fine magnetic dust suspended in a thin oil. Where a crack is present a change in relative permeability exists and there is

a consequent change in the flux density. This change of flux density attracts magnetic particles into the crack thus highlighting it. The method can be used to show up cracks at, or very close to, the surface.

Electrical methods [5] fall into two categories: eddy current and electrical potential. In the eddy current method a coil supplied with a.c. current is placed near to the surface of the sample. A time-varying magnetic field is produced by the coil which induces eddy currents in the sample. These currents flow in such a direction that the field they produce opposes the applied field. If any defects are present within the sample the eddy currents are reduced and the net magnetic field is increased. The inductance of the coil is therefore dependant on the presence of defects within the sample and by measuring the change in impedance of the coil as it is moved over the sample any defects present can be detected. In the electrical potential method two probes, connected to a constant current source, are applied to the surface of the material. Any defects present increase the resistance between the probes and thus the potential difference between them. By monitoring the voltage across the probes any defects present can be detected. Deep defects may be detected using a d.c source, however errors can occur due to irregularities in contact potential. Such errors do not occur with an a.c. source but here the depth to which cavities can be detected is limited by the skin depth of the material.

Penetrant methods [6] are essentially an extension of visual inspection, here the surface of the material is covered with a liquid which migrates into any cracks present by capillary action. The exact inspection procedure depends on the penetrant used e.g. ultra-violet light in conjunction with fluorescent dye shows up cracks as bright areas.

1.3 AUTOMATED TESTING

1.3.1 INTRODUCTION

All destructive and non-destructive tests methods require significant skill and concentration on behalf of the operator in order that accuracy is consistently guaranteed. The rising cost of skilled labour and falling cost of micro-electronic technology have made automated systems, in which test data is analysed by a computer, economically appealing to industry. Such systems can give significant reductions in operating costs, since expenditure is confined to the initial design and development costs, and improvements in reliability, since operator fatigue/boredom is no longer a problem [7].

In cases where the test parameter to be measured is a single analogue quantity (e.g. the voltage between two probes in an electrical potential test) the design of an automated system is a relatively trivial matter. In other cases, however, it may be necessary to compare the overall variation of the test parameter (e.g. the waveform produced by a detector in ultrasonic test) with some known reference; here it is necessary to the give the system more "intelligence".

Where the inspection involves a visual image, such as a magnetic particle or dye penetrant test the computer system needs to be given "sight" as well as "intelligence". The "sight" is usually in the form of a vision system comprising video camera and frame store and the "intelligence" in the form of a computer program. The program must be capable of giving some meaningful description of the image (e.g. size and position of a crack) to the operator. The process through which this description is obtained is known

as image recognition. The program may also need to provide intermediate steps in which one image is converted into another before recognition can be attempted, this process is known as image processing. An outline of image processing and recognition theory is given in Chapter 2 along with a description of vision system hardware.

1.3.2 APPLICATIONS IN AUTOMATED WELDING

In manufacturing industry many robotic systems exist for automated welding [8]. These systems select welding parameters (e.g. torch speed, arc current, gas flow etc.) from a data base using information gathered from a variety of sensors [9]. The use of open loop control means that weld quality cannot be fully guaranteed, to ensure this NDT sensors need to be incorporated in a closed loop system to monitor weld quality "a posteriori" [10]. If part of a weld is then found to be defective the control system can ensure that it is re-welded at a later stage. Cracks are an important class of weld defects and can be detected automatically using a vision-based system [11]. Weld hardness is another important factor and can be measured on-line using the Shore hardness test [12]. The Vickers hardness test [13], although unsuited to on-line use, provides more accurate results than the Shore test and may be used to guarantee weld quality at the final quality assessment stage of production. Hardness is defined as the ability of a material to resist permanent (plastic) deformation. The Vickers test uses the size of the permanent impression left by an indenter forced into the test material under a known load as a hardness index. In the Shore test hardness is judged by the maximum rebound height attained by an indenter incident on the material.

In many industrial components, e.g. cog teeth, machine tools, tunnelling equipment etc., resistance to deformation (i.e. hardness) is the most important property in choosing

which material the component is to be manufactured from. Hardness is related to other material properties, for example, ultimate tensile strength which is of importance in the selection of material for load bearing members such as steel beams.

In the manufacture of machine tools, the material employed is steel alloyed with a small amount of carbon. Initially the material is relatively soft, but is hardened by heating to approximately 1200 degrees Celsius, then cooling rapidly by quenching. If full hardness is to be achieved, then the cooling rate must exceed a value known as the critical cooling velocity. One way of measuring this is to quench one end of a cylindrical bar (known as a Jominy bar) then take readings of Vickers hardness along the length of the bar. Those points furthest from the bar will cool slower than those nearest the quenched end and will have correspondingly lower hardness values. By analysing the hardness profile, the critical cooling velocity can be ascertained.

In service, a machine tool will only experience loading at the surface and therefore it is both unnecessary and uneconomic to harden the material throughout it's bulk. The use of the Jominy bar and hardness tester allows the depth of hardening to be measured and thus the most economic combination of component geometry, alloying ratio and quenching to be arrived at.

In welding, hardness measurement is used to assess the quality of the heat affected zone (HAZ); this being the fused zone where the weld material is deposited. If the HAZ is not hard enough then cracks may occur either during the welding process, due to cooling stresses or hydrogen embrittlement, or in service when the working load is applied. It should be noted that in welds of insufficient hardness, the crack propagation rate will be significantly increased as material is less able to resist plastic stresses.

Vision system implementations of the Shore and Vickers tests are described in Chapter 3 and Chapter 4, respectively. The background to both tests is given in Chapter 2, test results are presented in Chapter 6 and suggested improvements discussed in Chapter 7.

1.3.3 AN APPLICATION IN QUALITY ASSESSMENT

The Vickers and Shore tests are examples of non-destructive tests in which the material properties of a component are to be measured. A vision system implementation of the other two types of test, namely measurement of physical dimensions of components and detection of defects within components is given in Chapter 5. The actual application is for the quality assessment of machined components, these being ferrite discs, approximately 8 mm in diameter, coated with aluminium. The system is shown to be capable of characterising geometric defects by parameters such as edge eccentricity and hole offset as well as characterising other defects such as chips and cracks by parameters such as size, position and shape. Results obtained from the system are presented in Chapter 6 and suggestions for improvements to the system given in Chapter 7.

CHAPTER 2

FUNDAMENTALS OF HARDNESS TESTING AND COMPUTER VISION

2.1 INTRODUCTION

In this chapter the background theory underpinning subsequent chapters is discussed. The development of hardness testing, as an NDT method, is outlined in section 2.2. with particular attention paid to the Vickers and Shore types of test. Vision system implementations of the Shore and Vickers test are described in Chapter 3 and Chapter 4, respectively.

In this work all the applications of computer vision fall into the category of automated visual inspection (AVI). Since AVI is a system technology and not simply an information technology (as are computer tomography, image compression etc.) it encompasses a variety of areas [14], namely:-

(a) mechanical handling of objects

(b) optics, lasers, illumination and viewing techniques

(c) television, image scanners and digitisers

(d) electronics

(e) computer architecture and software

(f) algorithms

None of these items is more or less important than the others, all are mutually dependant and essential. The first item falls under the area of robotics, which is outside the scope of this work, and is not discussed further. Items (b) to (e) are discussed in section 2.3 while the theory leading to image processing and recognition algorithms is given in section 2.4. A detailed discussion of algorithms for specific applications is reserved for later chapters.

2.2 HARDNESS TESTING

2.2.1 THE NATURE OF HARDNESS

The concept of hardness is an old, if poorly understood, one. In common parlance the terms "hardness" and "strength" are taken to be synonymous, however a simple example reveals this not to be the case. Steel is far stronger than glass, but a steel needle will not scratch (deform) glass although glass will mark steel, therefore glass is apparently harder than steel. As another example, when one talks about the hardness of a cushion it is primarily the elastic properties of the cushion and not the hardness of it's components that is being considered. To resolve these ambiguities, hardness has been scientifically defined as *the ability of a material to resist plastic (permanent) deformation [13].* This definition is particularly applicable to metals since the range over which they deform elastically is relatively small and consequently any deformation is predominantly plastic.

2.2.2 HISTORICAL DEVELOPMENT

Prior to the twentieth century, the only hardness test in common use was the Mohs test [15]. This was based on the principle that the harder of two materials can scratch the other but not vice-versa. A ten point scale was employed with talcum having a Mohs number of 1 and diamond Mohs number 10 (hardest). Other materials were given a number between the hardest material it could scratch and the first one that it could not. The test was a crude one and in the early part of this century more accurate tests were developed giving a continuous measure of hardness. These tests fall into three categories: plough tests, rebound tests and static indentation tests.

In the plough test a pointed indenter, usually made of diamond, is drawn across the surface of the material producing a scratch. The width of the scratch is then measured and used as a hardness index. The test was both cumbersome and slow and has largely been superseded by the other types. In rebound tests, a steel ball, or diamond tipped hammer is dropped onto the surface of the material from a fixed height. The maximum height attained after the initial impact is then measured and used as a hardness index. The most well known rebound test is the Shore type [16] in which a drop height of 250 mm is employed. In static indentation tests an indenter is forced into the surface of the material under a known load and the surface area of the permanent impression left used as a hardness measure.

The first indentation test was developed by Brinell in 1900 [17,18]. A hard steel ball, 10 mm in diameter, was originally used as an indenter and since no hardness scale was in existence Brinell devised his own:-

$$\text{Brinell Hardness Number (BHN)} = \frac{\text{load}}{\text{spherical area of indentation}} \qquad [2.1]$$

The load is varied such that the chordal diameter of the indentation is between a quarter and a half of the diameter of the ball (Fig. 2.1(a)); this is to average out any heterogeneities on the surface of the material. A drawback with this test is that for BHN greater than about 400 the scale becomes non-linear due to elastic deformation of the indenter. For this reason, later versions of the test utilised a tungsten carbide indenter, however this merely delayed the onset of elastic deformation to about 600 BHN. Another problem with this test is that it cannot be applied to small or precision components because of the large size of the deformation.

In light of these difficulties, tests employing pyramidal or conical indenters were developed. Here the small contact area produces plastic deformation of the test material even at small loads since the high stress (pressure) produced will almost always be greater than the yield stress of the material. In 1922, Smith and Sanderland [19] produced a test in which a diamond indenter in the form of a right pyramid was used; this effectively overcame the problems of elastic deformation of both indenter and material. This test, now known as the Vickers hardness test, has become pre-eminent.

Another indentation test which merits discussion is the Rockwell test [20]. Here a small load is applied to an indenter to cause deformation of the material and the load subsequently increased until no further penetration occurs. The size of the additional load is then used as a hardness measure. This test has advantages over the Vickers test in that it is direct reading and does not suffer from "spring back" (distortion of the indentation shape due to the release of elastic stress when the indenter is removed), however local work hardening of the material can cause erroneous results.

2.2.3 VICKERS HARDNESS TEST

In the Vickers test [13], a diamond indenter in form of a right pyramid is forced into the material under a known load and the size of the permanent impression left used as a hardness index (Fig. 2.2). For the test to be universally accepted correlation was necessary between the Vickers scale and an existing scale; this reference scale was chosen to be the Brinell one. If the opposite sides of the pyramidal indenter were tangents to a sphere, of diameter D, the distance between the contacting points would be 0.375D (Fig. 2.1(b)), corresponding to the median of the optimum range for the indenter's diameter in the Brinell test (0.25D to 0.5D). The test is performed in the following way:-

(1) The diamond indenter is placed just above and normal to the surface of the test piece.

(2) The indenter, subjected to a constant load, is forced into the piece for 10-15 seconds.

(3) The load and indenter are removed simultaneously and in the same direction as entry i.e. perpendicular to the surface.

(4) The test piece is placed under a microscope fitted with a vernier scale and the two opposing diagonals of the square shape are measured (in millimetres) and their average recorded.

(5) Having obtained the average diagonal length the hardness is found either from look-up tables supplied by the British Standards Institute [21] or from the formula derived below.

Since hardness is defined as the resistance to plastic deformation, the Vickers hardness must be inversely proportional to the indentation size (for a given load) and proportional to the applied load (for a given indentation size), therefore:-

$$\text{Vickers hardness} = \frac{\text{load applied (in Kg)}}{\text{surface area of the indentation (in } mm^2)} \qquad [2.2]$$

In the case of a pyramidal indenter this is equivalent to:-

$$\text{Vickers hardness} = \frac{2F\sin\alpha}{D^2} \qquad [2.3]$$

where F is the applied load in Kgf, D the average diagonal length (i.e. the mean of d_1 and d_2 in Fig. 2.2) in mm and α is the semi-angle of the pyramid ($= 68\ °$). If D is expressed in millimetres this collapses to:-

$$\text{Vickers hardness} = \frac{1.854F}{D^2} \qquad Kg.m^{-6} \qquad [2.4]$$

For example the hardness is usually expressed in the form 100 HV30, indicating a hardness reading of 100 when a 30 Kg load was used. The hardness numbers obtained are almost identical to those of the Brinell test up to about 400 BHN, above this there is an increasing discrepancy due to the non-linear nature of the Brinell scale. Errors in the Vickers test may arise due to two effects; "sinking in" and "piling up" (Fig. 2.3). "Piling up" occurs when metal, displaced by penetration of the indenter, is forced upwards in the immediate vicinity of the indenter.. This produces an indentation which appears larger than that expected and is a characteristic of highly worked metals. "Sinking in" occurs when metal is displaced both upwards and laterally outwards from the indenter so that the material around the indenter is at a lower level than that further away. This produces an indentation which appears slightly smaller than expected and is a characteristic of annealed metals.

The Vickers test is subject to certain requirements laid down by the British Standards Institute [22]. These are, briefly:

- The distance from the centre of the indentation to the edge of the test piece (or between the centres of two adjacent indentations) must be greater than 2.5 × length of the average diagonal.

- The semi-angles of the pyramid indenter must all be 68 ± 0.3 °.

- The point size of the pyramid must be within 0.5% of it's stated value.

- The thickness of the test piece must be greater than 1.5 × average diagonal.

- The diagonals must be measured with an accuracy of 0.002 mm.

2.2.4 SHORE HARDNESS TEST

In the Shore Scleroscope [12] a hard steel ball, 3 mm in diameter, is dropped vertically through a guide tube onto the test material and the maximum rebound height attained after the initial impact is measured. Two common scales are in use, the magnifier scale, for polymers, and the universal scale, for metals. Under the universal scale, the 250 mm drop height is divided evenly into 140 divisions so that the Shore number is given by:-

$$\text{Shore number} = 140 \times \left[\frac{h}{250} \right] \qquad [2.5]$$

where h is the rebound height expressed in millimetres. Under this scale high carbon steels have a Shore number of 90 - 100.

It is instructive to examine the physical processes occurring during the test. When the indenter first falls from rest it's potential energy is converted to kinetic energy as it's height diminishes. At the point were the indenter strikes the surface of the material all of the potential energy has been converted into kinetic energy which is then dissipated in the setting up of elastic and plastic stresses in the material. As the elastic stresses are released the indenter regains kinetic energy causing it to travel upwards. At the top of it's flight all of the original kinetic energy has been converted into potential energy and therefore the difference between the potential energy at this point and at the initial drop point is equal to the energy dissipated in the material. It is reasonable to assume that the degree of plastic deformation produced in the material is proportional to the dissipated energy and since gravitational potential energy is proportional to height, the hardness of the material is directly proportional to the rebound height.

To justify the above assumption it is necessary to examine the physics of the impact in more detail. As the indenter initially impinges on the material surface, elastic deformation is produced (note that at very low impact energies the impact process is completely elastic). When the indenter penetrates further, the deformation becomes progressively more plastic until a point is reached when the indenter pressure exceeds 1.1 × yield stress, after which the deformation is completely plastic and any remaining kinetic energy is dissipated. Finally, the elastic stresses are released producing the rebound.

Clearly, for the measurements to be meaningful, the impact energy of the indenter must be such that the pressure it exerts exceeds the yield stress of the material. In the Shore Scleroscope, which produces relatively small indentations, this limit is reached at about 70 - 80 Shore making the scale slightly non-linear. This deviation is, however significantly less than the error range caused by variations in surface roughness, angle

of impact etc. and unaccounted energy loss mechanisms e.g. heating of the material, production of acoustic waves etc. In spite of these drawbacks, the portability and speed of the test make it useful for checking forgings *in situ* (e.g rolls in a mill, dies in a press etc.).

2.3 VISION SYSTEM COMPONENTS

2.3.1 ILLUMINATION TECHNIQUES

The choice of illumination scheme is an important consideration in the design of any vision system. Correct choice of lighting can significantly reduce the complexity of a scene and thus the complexity of the image analysis algorithms by minimising shadows, "hot spot" reflectances etc. It should also be noted that it is far easier to alter the illumination set up than to tailor the image analysis software to cope with a poorly lit scene.

There are four main illumination schemes (Fig. 2.4 and 2.5), the first being the diffuse lighting approach. This is useful for objects having smooth, regular surfaces and is employed in applications in which surface characteristics are important. The back lighting approach produces binary (black and white) images and is useful in applications in which recognition can be performed using object silhouettes.

In the structured lighting method, points, stripes or grids are projected onto the scene. Any distortion in the pattern produced indicates the presence of objects and the type of distortion gives information about the three dimensional characteristics of the scene.

Directional lighting is useful for the inspection of object surfaces. Here a collimated directed light beam (e.g. from a laser) is directed onto the surface of the object so that smooth surfaces cause specular reflection of the incident light whereas defects (e.g. cracks) cause scattering of the light. The position of the camera viewing the scene depends on whether a bright field or dark field method is employed [23]. In the dark field method the camera views the surface of the object at the specular angle so that any defects present appear as dark regions on a bright background. This is because the amount of light emanating from defects at the specular angle is significantly reduced by scattering. In the bright field method the camera is placed at a non specular angle so that only scattered light is collected; here defects appear as bright regions on a dark background.

2.3.2 IMAGE ACQUISITION

In a typical vision system the image acquisition process is usually shared between a detector, which produces an analogue signal whose amplitude varies according to the light intensity falling on the scene, and a digitiser, which produces a sampled, quantised version of the analogue signal suitable for processing by computer.

Detectors may be constructed from groups of phototransitors/photodiodes, however it is more usual to make use of existing closed-circuit television (CCTV) cameras. These fall into two distinct categories; vidicon and charge-coupled device (CCD) types. In the vidicon camera the detector consists of a glass vacuum tube containing an electron gun at one end and a photosensitive plate at the other. Light falling on the plate causes a charge separation between the inner and outer surface which may be sensed by scanning the plate with an electron beam. As the charge density varies across the plate the

electron beam current varies in sympathy. This variation in beam current, as scanning takes place, produces, after suitable conditioning, a video signal proportional to the input image intensity.

The scanning mechanism is usually governed by existing TV standards. In the UK, where the Phase Alternation Line (PAL) standard is adopted, this requires that the entire image is scanned 25 times a second and that 625 lines are used for each complete scan (known as a frame). The image is actually scanned from top to bottom such that all the even numbered lines and all the odd numbered lines are scanned alternately (Fig. 2.6). The set of even lines is known as the even field and the set of odd numbered lines the odd field.

In the CCD type of camera, a solid state detector consisting of a matrix of silicon imaging elements, known as photosites, is employed. When light falls on a photosite electron-hole pairs are created giving rise to local charge separation; the amount of charge collected at a photosite being proportional to the illumination intensity at that point. Gates and shift registers are used to transfer the charge from each photosite to an amplifier so that a voltage signal is produced whose amplitude is proportional to the image intensity. The timing is arranged such that the image appears to be scanned in the same manner as for the vidicon type.

Since vidicon cameras were originally developed for television and not computer vision applications there are a number of drawbacks to their use in vision systems, primarily:

- The length and width of the scanning area are different (the ratio of length to width being referred to as the aspect ratio); this means that the spatial separation corresponding to adjacent pixels (see section 2.4.1) is greater in the horizontal

direction than the vertical direction. This factor must be taken in to account in metrology applications.

- The relationship between video signal amplitude and illumination is generally not linear but exponential. This must be borne in mind when attempting absolute intensity measurements.

- Most cameras incorporate some form of automatic gain control (a.g.c.) so that the relationship between video signal amplitude and illumination intensity is time-variant. This must be taken into account when comparing the intensity levels of different frames.

These considerations also apply to CCD CCTV cameras, however CCD cameras designed specifically for computer vision are available.

The vidicon camera is more prone to saturation than the CCD type and unlike CCD types suffers from "burn out" (irreparable damage to the vacuum tube phosphors caused by over exposure); this usually makes them unsuited to applications involving laser illumination. The spectral response of the vidicon camera is similar to that of the human eye, unlike the CCD type whose response extends into the infra-red region; whether this is an advantage or a disadvantage will depend on the application.

To allow processing of images, the analogue video signal from the camera has to be converted into a digital form; a convenient way of achieving this is to use a frame store. The frame store contains a high speed A/D converter which digitises the analogue video signal. Data from the A/D converter is stored in sequential memory locations in the frame store on-board RAM. Most frame stores are capable of capturing images in

real-time, this requires that an entire frame is digitised and stored within 40 ms. The field sync and line syncs are extracted from the composite video signal and used as enables to counters. These counters generate addresses for the frame store memory so that the image data is stored line-by-line as a contiguous block. Frame store memory operations are controlled by hardware timing circuits and bus arbitrators so the image data can be accessed by the host computer at normal microcomputer bus speeds.

The size and organisation of the frame store memory determine the amplitude and spatial resolutions of the frame store (see section 2.4.1). Typical devices have a spatial resolution of 512 x 512 pixels and an amplitude resolution of 256 levels. Insufficient spatial resolution produces a "chess board" pattern on the output image, whereas insufficient amplitude resolution leads to ridge like structures (also known as false contours) within the output image. The minimum spatial resolution generally acceptable is 256 x 256 pixels and the minimum amplitude resolution 64 grey levels; for comparison, the requirements for images of equivalent quality to monochrome television are 512 x 512 pixels with 128 amplitude levels.

2.3.3 PROCESSING HARDWARE

In software based systems the digital image information is stored in memory, usually by using a frame store and usually processed by a single microprocessor, executing the image analysis program. Such systems are flexible, since the analysis algorithms can easily be altered by executing different program code, however speed is limited by the power of the microprocessor. Another inherent speed limitation is that the large amount of data contained within an image (typically 256 kbyte) can only be accessed by the microprocessor a small amount at a time (typically 2 bytes per memory access) - this is

usually referred to as the "Von Neumann bottle-neck". The problem may be overcome, to some extent, by use of a multi-processor system running concurrent software (e.g. transputer based system); such systems are relatively rare in computer vision, at present. In cases where processing speed is critical, image analysis operations may be realised by hard-wiring. These, hardware based, systems are expensive and provide no inherent flexibility.

2.3.4 CHOICE OF PROGRAMMING LANGUAGE

Programming languages, designed specifically for the production of image analysis software are relatively rare and do not conform to any one standard. It is, therefore, more usual to make use of conventional data processing languages to implement image processing and recognition algorithms. Ideally the chosen language should have the following attributes:-

- Fast execution speed. In an AVI system the required processing speed is mostly determined by the cycle time of the production process; typical cycle times varying from 100 ms to 10 s. Since AVI systems are required to manipulate large amounts of image data (typically 256 kbyte per captured frame) processing speed is of paramount importance.

- Structure. Languages which are structured allow programs to be easily written and understood by other people. This can significantly reduce development time when software is to be developed by a group of programmers or modified at some later stage. Structured languages may, however lead to inefficient object code and poor execution speed if the compiler is not carefully designed.

- Modularity. If large programs are to be developed it is convenient to break them down into smaller modules which can be linked together at a later stage. This modular approach can reduce development time since the entire program does not have to be compiled every time an alteration is made.

- Low level interface. In any vision system it is necessary for the host computer to control the digitising hardware. The programming language should, therefore be capable of either controlling peripherals directly or via a low level language e.g. assembler.

- Library functions. Development time can be reduced if the programming language includes a library of functions which can be incorporated into user written programs. This saves the user the task of writing code for low level functions e.g. fetching a character from the keyboard etc.

- Portability. It is often the case that software which is developed on one computer needs to to be run on another incompatible machine in the target system. It is, therefore useful to employ a language which is not specific to a given operating system or processor.

Assembler code is useful for controlling peripherals but is tedious to use for image analysis software. High level languages e.g. FORTRAN, Pascal etc., are more suitable for image analysis programs but have, historically, been developed for use on mainframe computers and therefore tend to be machine specific. FORTRAN is unstructured, has only limited library functions and cannot easily be used to control hardware or interface to assembler code. Pascal overcomes many of these disadvantages but produces inefficient object code because of strict run-time error checking. Intermediate level languages combine the structured nature of high level languages with the capability to

access system hardware. The most useful of these languages for computer vision is probably C. This is a modular language which provides a large amount of library functions and produces efficient object code. C programs can be run on a variety of different machines simply by changing the compiler and function library to suit the microprocessor.

2.4 IMAGE PROCESSING AND RECOGNITION

2.4.1 BASIC RELATIONSHIPS

The result of the image acquisition process is a set of data representing a spatial and amplitude quantised version of the original image formed on the camera. It is convenient to define an image function f(x,y) which represents the quantised image intensity (grey level) at spatial co-ordinates (x,y) [24]. If the image is sampled uniformly into an array containing N rows and M columns then an image array (digital image) can be defined such that:

$$f(x,y) \equiv \begin{bmatrix} f(0,0) & f(0,1) & \ldots & f(0,M-1) \\ f(1,0) & f(1,1) & \ldots & f(1,M-1) \\ \ldots & \ldots & \ldots & \ldots \\ f(N-1,0) & f(N-1,1) & \ldots & f(N-1,M-1) \end{bmatrix} \qquad [2.6]$$

$$f(x,y) = 0,1,2,\ldots,G-1$$

where x and y are now discrete variables such that x=0,1,2,...,N-1 and y=0,1,2,...,M-1 and G is the number of amplitude quantisation intervals (grey levels). Each element of the image array is referred to as a pixel.

A pixel p at co-ordinates (x,y) has four vertical and horizontal neighbours whose co-ordinates are given by:

$$(x+1,y) \quad (x-1,y) \quad (x,y+1) \quad (x,y-1)$$

This set of pixels is called the 4-neighbours of p and is denoted by $N_4(p)$ .The four diagonal neighbours of p, denoted by $N_D(p)$, have co-ordinates:

$$(x+1,y+1) \quad (x+1,y-1) \quad (x-1,y+1) \quad (x-1,y-1)$$

These points, together with the 4-neighbours are called the 8-neighbours of p, denoted by $N_8(p)$.

A set of pixels is said to be connected if all the members share some common property e.g. the same grey level. Consider a pair of pixels p and q having intensity values within a set, V. There are three types of connectivity which can be associated with p and q:-

- 4-connectivity if q is in the set $N_4(p)$.
- 8-connectivity if q is in the set $N_8(p)$.
- m-connectivity (mixed connectivity) if

  1. q is in $N_4(p)$ *and*
  2. q is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ is empty.

A pixel p is said to be adjacent to a pixel q if they are connected, so that pixels can be 4-, 8- or m- adjacent depending on the type of connectivity.

A path from a pixel p with co-ordinates (x,y) to a pixel q with co-ordinates (s,t) is defined as a sequence of distinct pixels with co-ordinates:-

$$(x_0, y_0),\ (x_1, y_1),\ \ldots,\ (x_n, y_n)$$

such that $(x_0, y_0) = (x, y)$ and $(x_n, y_n) = (s,t)$ and (x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}), $1 \leq i \leq n$, where n is the length of the path. If p and q are pixels from the same image subset S, then p is connected to q (also in S) if there is a path from p to q consisting entirely of pixels in S. The set of pixels in S, connected to p, is called a connected component of S.

2.4.2 PREPROCESSING

There are four major preprocessing operations; grey scale conversion, geometric equalisation, histogram transformation and linear filtering. These operations can contribute significantly to the effectiveness of segmentation (section 2.4.3) and feature extraction (section 2.4.4) as well as improving visibility to the user [25].

Grey scale conversion ensures that the full dynamic range of the digitising device (e.g. frame store) is utilised. The grey level of each pixel within the image, g is mapped to a new grey level g' according to:-

$$g' = \begin{cases} g_{min}, & g \leq t_1 \\ \left(\frac{g_{max} - g_{min}}{t_2 - t_1} \right)(g - t_1) + g_{min}, & t_1 < g < t_2 \\ g_{max}, & g \geq t_2 \end{cases} \qquad [2.7]$$

$$g_{max} > t_2 > t_1 > g_{min}$$

where g_{max} is the maximum and g_{min} the minimum permissible grey level of the digitising device and t_2 and t_1 are two threshold parameters (usually the maximum and minimum grey level present within the original image, respectively).

Geometric equalisation is used to compensate for optical aberrations within images and is usually performed by linear co-ordinate system transformations.

Histogram transformation is the process by which the grey level of each pixel in an image is mapped to a new grey level to yield an image with a specified intensity histogram. Commonly, the desired histogram is flat so that all parts of the image have equal contrast. This, histogram equalisation, process is capable of compensating for effects such as shadows and specular reflections.

Linear filtering is often used to suppress image noise caused, for example, by dust on the vision system optics and photographic film imperfections. Noise in grey scale images may be reduced by an averaging process (also called smoothing); this can be either temporal or spatial. In the former case, the grey level of each pixel within the image is replaced by the average intensity at that point taken over several frames [26]. In the latter case, the grey level of each pixel is replaced by the average grey level of itself and it's neighbours (usually those which are 8-adjacent). This process produces blurring of the image unlike the median filtering method of smoothing [27] in which the grey level

of each pixel is replaced by the median grey level value of itself and it's neighbours. The median filtering method is, however, more computationally demanding. Smoothing may also be applied to binary images [28] produced by backlighting or thresholding (see below). The process is described in detail in section 4.4.1.

2.4.3 SEGMENTATION

Segmentation is the process through which an image is divided into meaningful regions [29] and is based on two basic principles; similarity and discontinuity. Region segmentation attempts to group together pixels having similar characteristics (e.g. intensity, texture etc.) whereas contour segmentation attempts to group together pixels marking sharp transitions within the image to form closed contours which outline homogenous regions.

There are three main region segmentation methods; pixel aggregation, splitting-and-merging and thresholding. In the first method a "seed" pixel is chosen and neighbouring pixels, having similar properties, are added to grow regions. The choice of seed pixel is critical and usually requires *a priori* knowledge of the scene. The method is most successful when a range of properties are considered (e.g. multi-spectral data in remote sensing applications) these are seldom available in AVI applications.

Split-and-merge [30] is essentially the opposite process to pixel aggregation. The starting point here is the entire image which is successively divided into smaller areas (usually quadrants) according to a given predicate. If the predicate is false for a given region then the region is split, however if the predicate is true for any two adjacent regions then the two regions are are merged together. The process is repeated until no further

splitting or merging is possible. At each stage of the process a record is kept of which pixels belong to regions that have been merged so that the single segmented region left at the end can be identified. The process is usually too computationally demanding for AVI applications where speed is critical.

Thresholding [24] is the most widely used segmentation method in AVI since it can easily be realised in hardware and therefore carried out in real time. Consider the intensity histogram of an image containing a light object on a dark background (Fig. 2.7). One mode in the histogram will correspond to the object and the other to the background. To segment the image requires only that each pixel is designated an object pixel if it's intensity exceeds a threshold value (T in Fig. 2.7) or a background pixel if it's intensity falls below this value.

The thresholding process can be extended to work with images having tri-modal intensity histograms by using two threshold limits (T1 and T2 in Fig. 2.7). Pixels are assigned as object pixels if they fall between these limits or background pixels if they fall outside of them. Threshold limits can either be fixed or adaptive (so that changes in ambient light can be compensated for) and be either global (apply to the whole image) or local (vary from one part of the image to another) - the latter is useful for images in which illumination varies across the scene.

There are two main approaches to contour segmentation [31]; edge enhancement and edge detection. In the edge enhancement method the image intensity is differentiated with respect to both spatial co-ordinates so that sudden changes in intensity (edges) are highlighted. The first derivative shows the presence of edges and the second derivative both the presence and relative position of edges i.e. whether the edge is on the dark or light side of the transition. The differentiation process is performed by convolving, in

two dimensions, the image array with a small array of multiplier co-efficients, known as a mask. For first order derivatives two masks are required, one to give the horizontal component of the gradient and the other to give the vertical component. The magnitude of the gradient is given by

$$\begin{aligned} G(f(x,y)) &= (G_x^2 + G_y^2)^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned} \qquad [2.8]$$

Where G_x and G_y are the horizontal and vertical components of gradient given by:-

$$G_x = f^{**}M_x \qquad [2.9]$$

$$G_y = f^{**}M_y \qquad [2.10]$$

where M_x and M_y are two matrices representing the vertical and horizontal gradient masks and "**" represents two dimensional convolution. In practise, an approximation to equation [2.8] is used, leading to faster implementations:

$$G(f(x,y)) \approx |G_x| + |G_y| \qquad [2.11]$$

The direction of the gradient, which is perpendicular to the edge orientation, is given by

$$\theta(x,y) = \arctan\left(\frac{G_y}{G_x}\right) \qquad [2.12]$$

The co-efficients within a mask are derived using numerical differentiation methods. As an example, the first order central difference equations, in two dimensions, are:

$$G_x = [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)] \quad [2.13]$$

$$G_y = [f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)] \quad [2.14]$$

these give the masks shown in Fig. 2.8, known as Sobel operators. Other first order operators are shown in Fig. 2.9 and a second order operator, known as the Laplace operator, is shown in Fig. 2.10.

In the edge detection approach to contour segmentation the image array is convolved with a series of templates of different orientations (Fig. 2.11). For each position the largest value is retained and when this value exceeds a threshold the area is considered to contain an edge element. The edge direction is considered to be the orientation of the mask giving the largest product ($\pm \frac{\pi}{4}$). This approach can be extended to line and point detection templates.

Once the original image has been processed using either edge enhancement or edge detection operators the next stage is to link together edge segments to form closed contours. The simplest approach to this to link each pair of edge segments in a predefined neighbourhood if their gradient magnitude and direction are similar. Finally, the contours are thinned to a single pixel in width before description is attempted.

2.4.4 DESCRIPTION

Description is the process of extracting features from segmented images for the purpose of recognition. It provides not only information about the size and position of objects

but also shape information which uniquely identifies one object from another, Descriptors fall into two main categories; regional and boundary.

Regional descriptors are particularly suitable to region segmented images and examples include area (i.e. number of pixels within a segmented region), centroid, and perimeter (i.e. length of the boundary of a region). The ratio $perimeter^2/area$ is a measure of the compactness of a region and is insensitive to changes in scale. The ratio is a minimum for disc shaped regions and is useful in NDT applications for identifying cracks, which by definition are long and thin and therefore have high ratio values (see Chapter 5).

Compact regions may be approximated to an ellipse [32] which is defined by four parameters; centroid, major and minor axis length, and rotation. The major and minor axis lengths provide size information and the rotation defines the orientation of the object (see Chapter 5). Other regional descriptors include moments of area, maximum and minimum radius to the boundary from the centroid and RMS of boundary radius from the centre. The former provides useful shape information, since moments which are invariant to scaling and translation can be employed, while the others give size information.

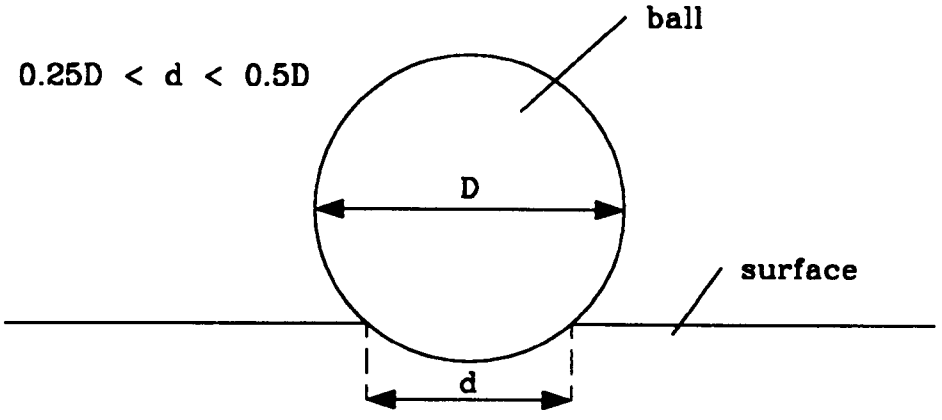
Boundary descriptors are particularly suitable for contour segmented images. One of the simplest descriptors is the chain code [33] in which the boundary contour is represented by a series of integers. The contour is traced and the sequence developed by recording values corresponding to the orientation of edge segments forming the contour. The code can be modified so that it is invariant under scaling and starting point and therefore provides useful shape information.

The signature (section 4.3.6) is another useful descriptor and is a one-dimensional functional representation of the boundary [34]. The simplest way to generate a signature is to plot the distance of the boundary from the centroid as a function of angle. The overall appearance of the plot provides shape information while the absolute measurements give size information.

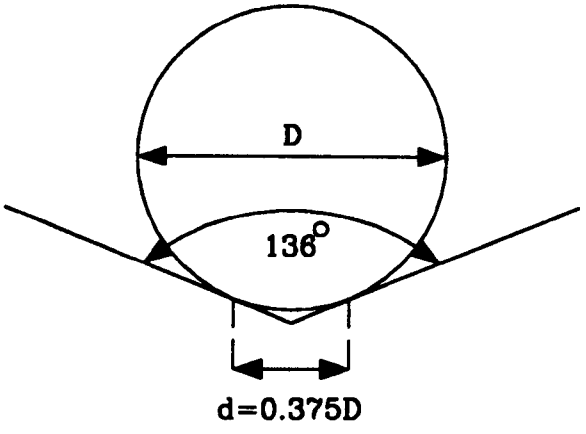
The discrete 1D Fourier transform can also be used to provide shape information. Here the boundary is treated as a contour in the complex plane so that a boundary point at (x,y) is represented by a complex number $x+jy$. By examining the Fourier transform of the sequence (which can easily be normalised with respect to size, rotation and translation) it possible to distinguish between shapes that are reasonably distinct.

2.4.5 USE OF MOTION

Motion provides a powerful means of extracting objects of interest from the background. One of the simplest ways of using motion to segment an image is to form the difference, on a pixel-by-pixel basis of the image with a reference image. Pixels within this difference image are assigned a 1 if the difference is less than a threshold and a 0 otherwise. The resulting image contains 1's only in places where movement of the object has occurred. In practise large amounts of noise can result if the two images are not spatially registered or if the illumination is not constant. This noise may be reduced by removing pixels labelled a 1's if they have less a given number of 8-neighbours, this is at the expense of ignoring slow moving objects. It should be noted that for binary images the difference operation is equivalent to the Boolean exclusive-OR operation.



(a)



(b)

Figure 2.1: INDENTATION TEST REQUIREMENTS

(a) Brinell test, (b) Vickers test

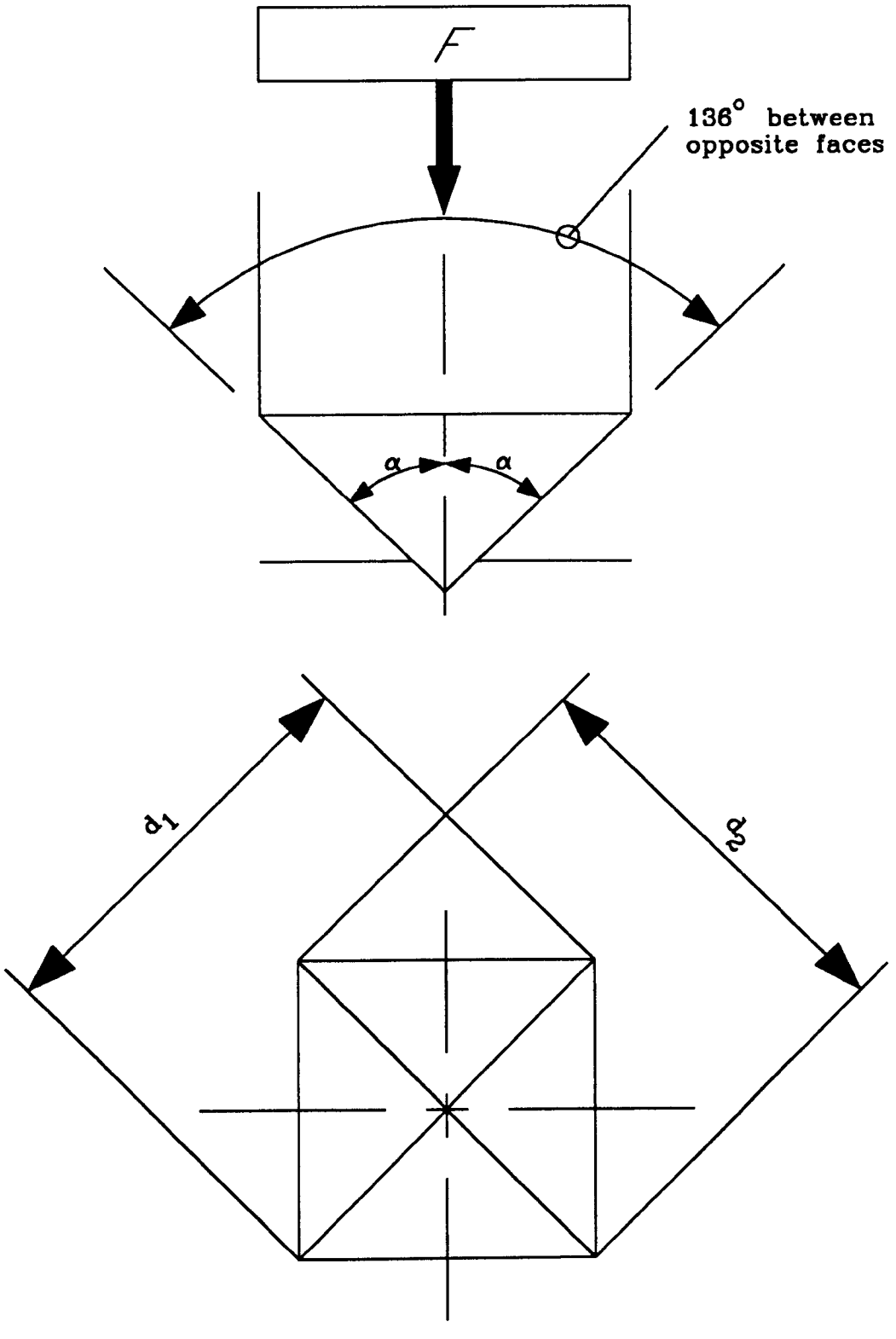
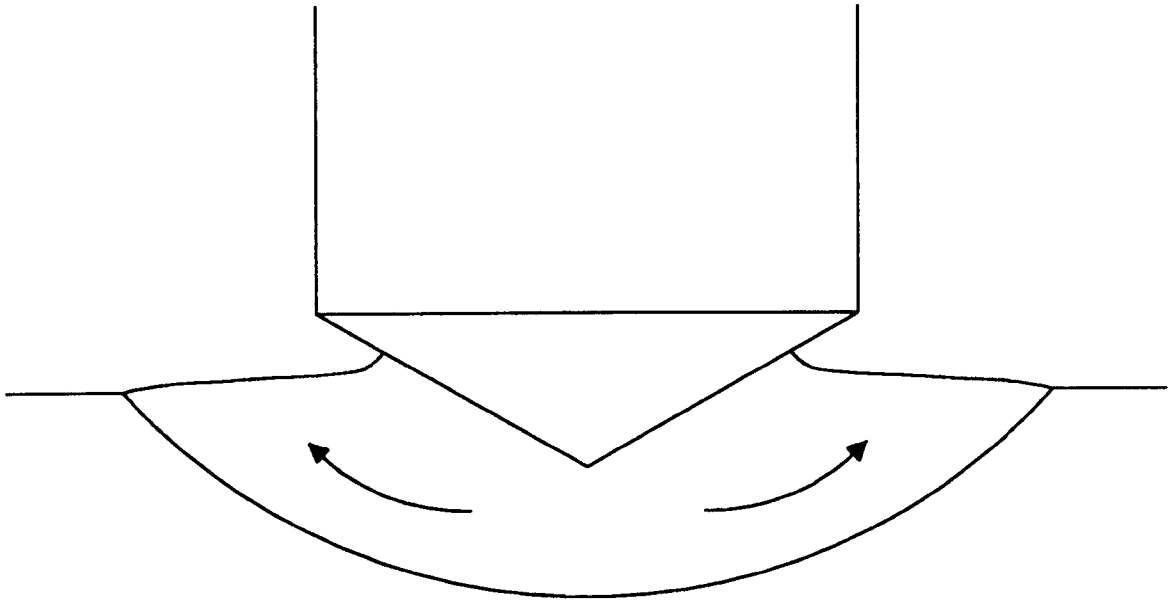
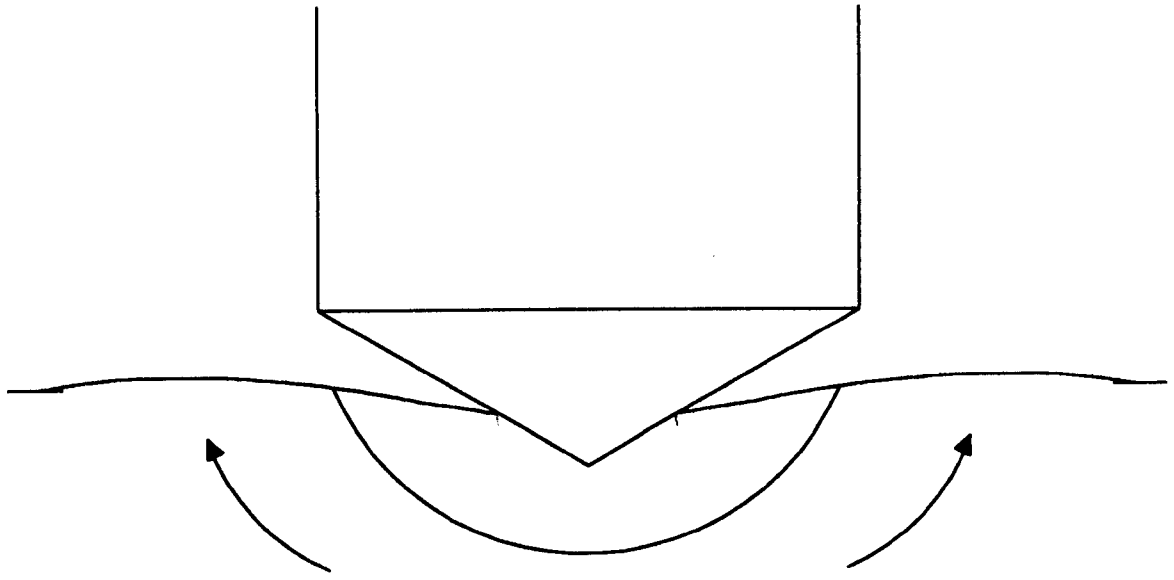


Figure 2.2: FORM OF THE VICKERS INDENTATION



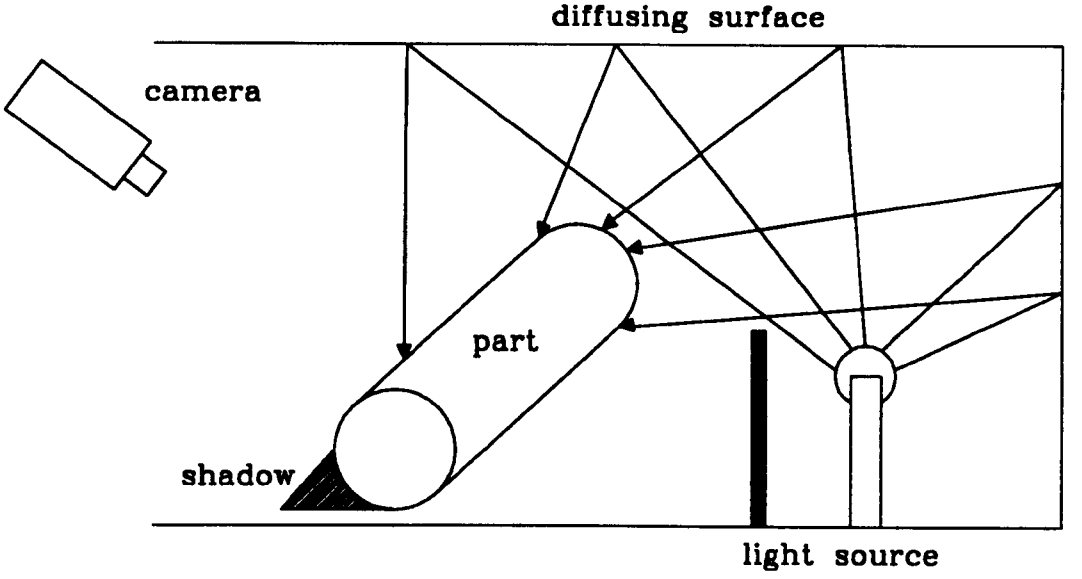
(a)



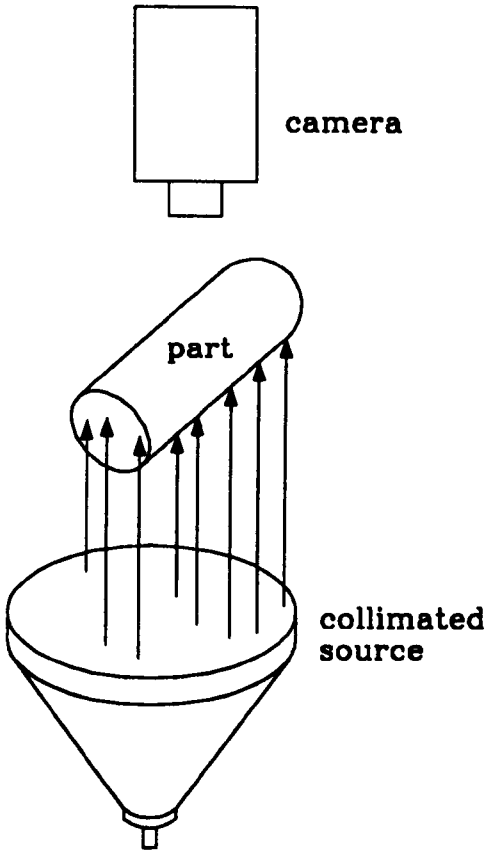
(b)

Figure 2.3: TWO TYPES OF DISTORTION

(a) 'piling-up', (b) 'sinking-in'; arrows indicate flow of metal



(a)



(b)

Figure 2.4: ILLUMINATION METHODS

(a) diffuse lighting, (b) back lighting

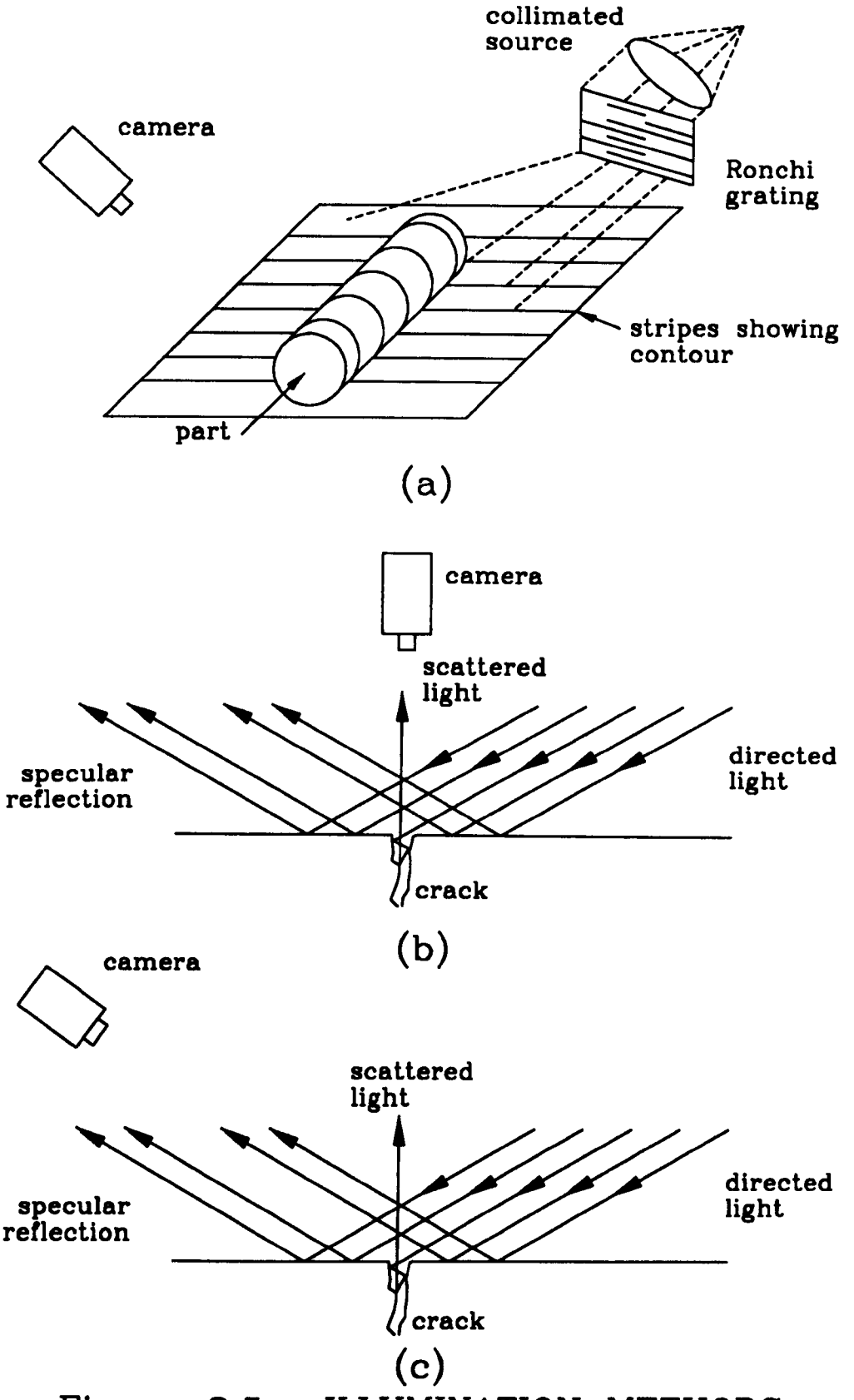
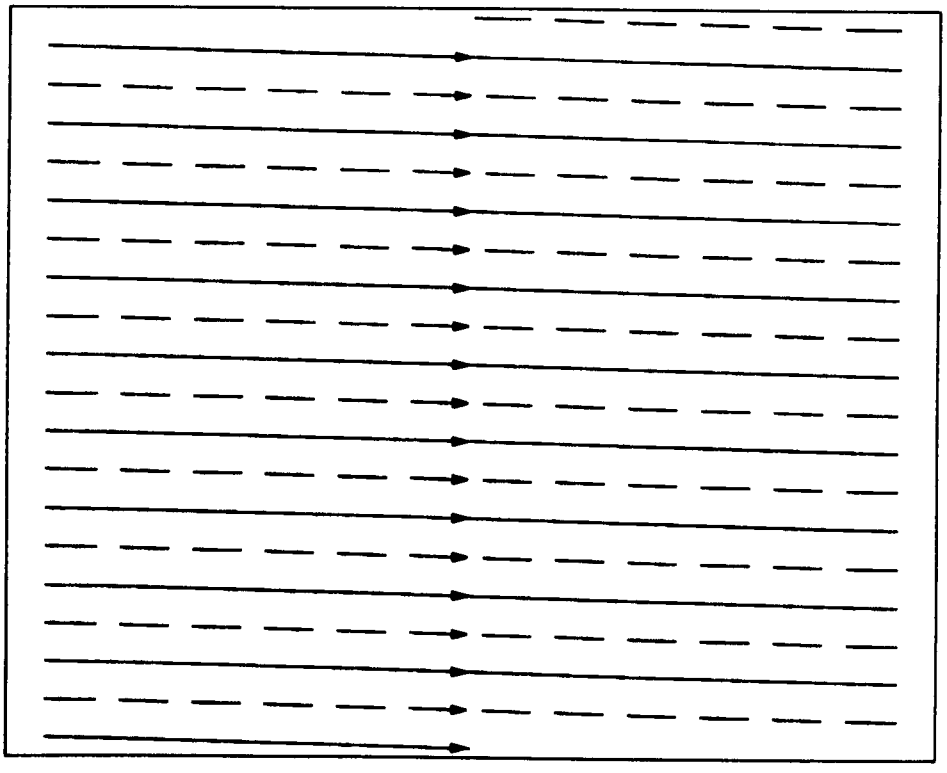


Figure 2.5: ILLUMINATION METHODS

(a) structured lighting, (b) directed lighting (bright field), (c) directed lighting (dark field)



even field lines

odd field lines

arrows indicate direction
of scan

Figure 2.6: ELECTRON BEAM SCANNING PATTERN

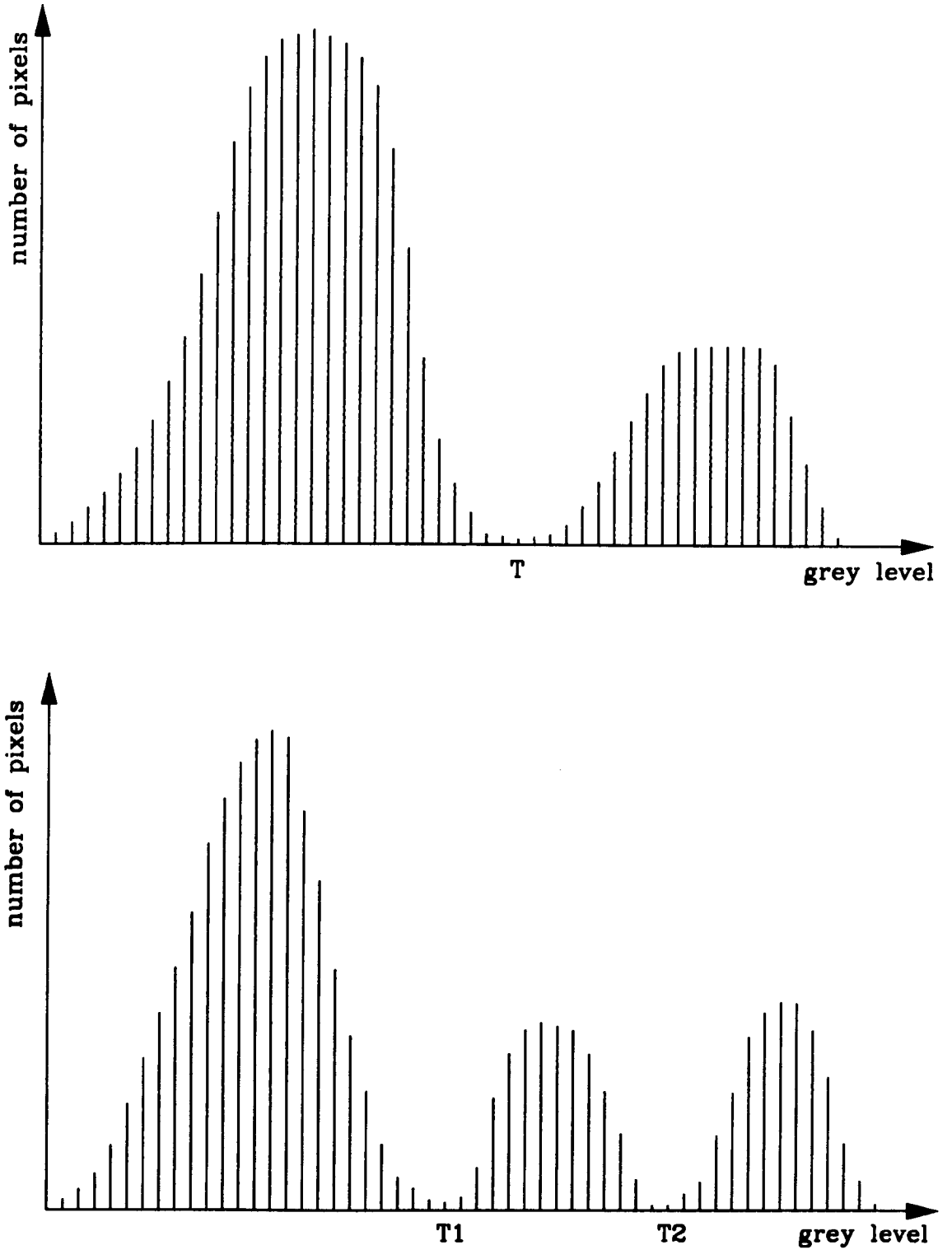


Figure 2.7: INTENSITY HISTOGRAMS FOR IMAGES WHICH CAN BE SEGMENTED BY THRESHOLDING

−1	−2	−1
0	0	0
1	2	1

(a)

−1	0	1
−2	0	2
−1	0	1

(b)

Figure 2.8: SOBEL OPERATOR MASKS

(a) mask used to compute Gx (b) mask used to compute Gy

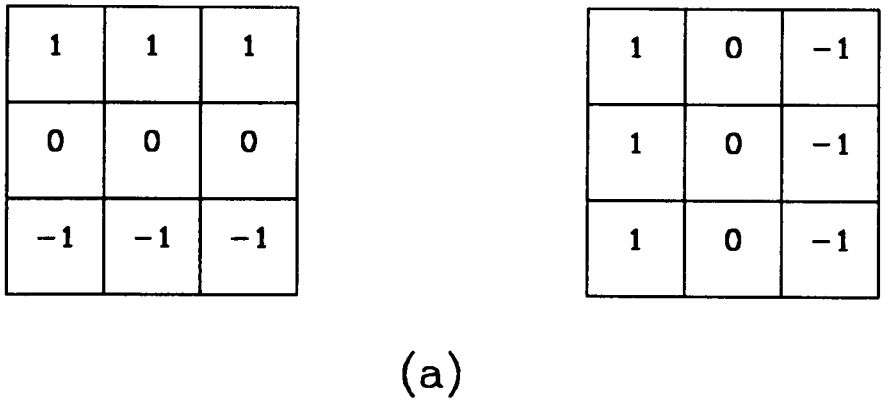


Figure 2.9: EDGE ENHANCEMENT MASKS

(a) smoothed gradient, (b) isotropic operator, (c) Robert's gradient

0	−1	0
−1	4	−1
0	−1	0

Figure 2.10: LAPLACE OPERATOR MASK

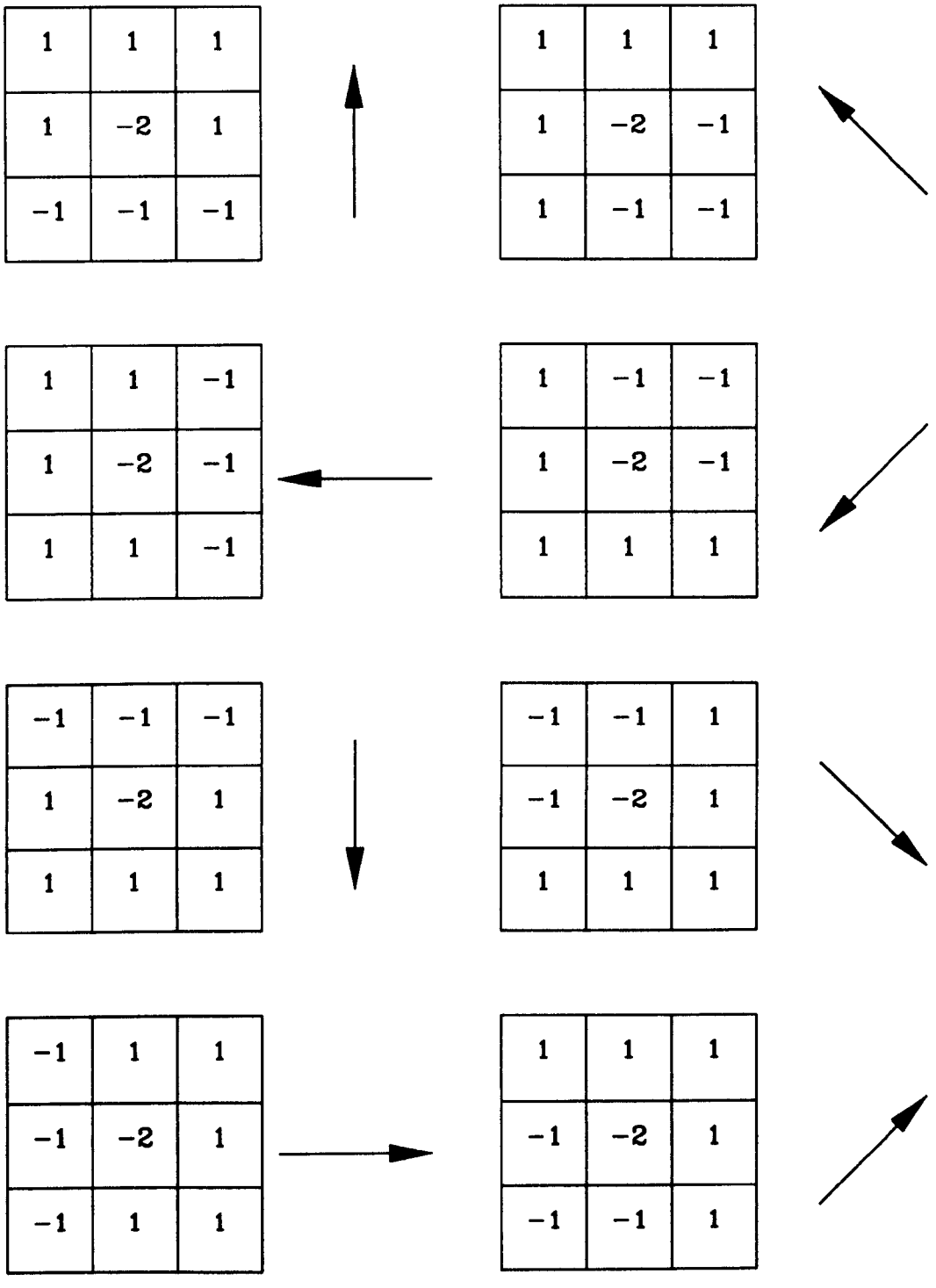


Figure 2.11: EDGE DETECTION TEMPLATES

arrows indicate gradient direction (perpendicular to edge orientation)

CHAPTER 3

SHORE HARDNESS TEST

3.1 INTRODUCTION

This chapter describes the development of a vision-based system for automation of the Shore hardness test, outlined in Chapter 2. The vision system is based on an IBM PC/AT compatible computer, fitted with a commercially available frame store. Details of the system hardware are given in section 3.2.

Two distinct approaches to the image analysis problem were attempted; one in which the image data is analysed in real-time and another in which the image data is first recorded and then analysed *a posteriori*. The second approach was later adopted in preference to the first since the limited computational power available gave rise to an unreasonable trade-off between speed and reliability. A comparison of the two schemes is given in section 3.3 along with a comparison of illumination methods.

The image processing software was written using the C language [35] with additional 8086 assembler code [36] used to provide a low level interface to the frame store. A full description of the C program code together with the assembler code routines is given in

section 3.4. Finally in section 3.5 a guide to setting up and using the hardness testing system is given.

3.2 SYSTEM HARDWARE

The vision system consists of five main parts; the test rig, illumination source, vision camera, frame store and host computer (Fig. 3.1 and Plate I). Each of these components is described below.

Test Rig

This consists of a solid metal stand which supports a glass guide tube perpendicular to the base on which the test piece rests (Fig. 3.2). The tube is attached to a perspex slide allowing the height of the tube above the base to be altered. A 250 mm scale is engraved on the slide so that the rebound height may be judged by eye.

The guide tube incorporates a near horizontal, but slightly sloping branch through which the 3 mm diameter ball bearing indenter is introduced. Since the slope is small the vertical component of the ball bearing's velocity is not significant and the ball essentially falls from rest. The results produced by the system should, therefore be comparable to those obtained using a commercially available Shore Scleroscope [12] in which the indenter is dropped vertically from the top of the guide tube. It is likely that the indenter will collide with the guide tube walls tending to reduce the impact velocity of the indenter and hence the rebound height. However, the additional fall through the branch of the guide tube will tend to increase the kinetic energy of the indenter and hence the rebound

height. It is not possible to predict, analytically, the combined effect of these two phenomena and an emipirical treatment is given in Chapter 6.

Illumination

Illumination is provided by a 15 mW HeNe laser [37]. Light from the laser is directed into the guide tube and onto the steel ball using a fibre optic. The laser beam is collimated to reduce reflections from the guide tube walls.

Monitors

Two monitors are used in the system, one for the camera output and one for the frame store output. Both are Hitachi monochrome types [38] with composite video inputs and a 625 line raster operating at 50 Hz.

Frame Store

The frame store is a Data Translation DT2853 [39] capable of storing two 512 x 512 pixel images, each with an 8 bit grey scale. The device is capable of capturing images from an interlaced sync video camera in real time (i.e. 25 frames/ sec) with modification of the digitised input made possible by one of eight 256 x 8 bit look-up tables (LUTs). The output may also be processed using one of eight 256 x 24 bit look-up tables giving pseudo-colour red-green-blue (RGB) video output. A feedback path is incorporated from output to input allowing mathematical operations to be performed on stored images using the look-up tables. Frames which have been stored in the frame store image buffer can be loaded into the memory of the PC. Conversely, image data can be

written to the frame store from the PC. The frame store is produced on a single board which is inserted into one of the PC-bus sockets of the computer.

Camera

The camera is a Hitachi CCTV [38] HV-739E vidicon type with a 625 line interlaced field raster operating at 25 frames/ sec. A 1:1.8/17.5-105 variable zoom lens with an adjustable aperture is fitted and the camera clamped approximately 1.5 m away from the test rig.

Computer

The host computer is an Opus PCV [40], IBM PC/AT - compatible machine with an Intel 80286 processor running at 12 MHz. A floating point 80287 co-processor is installed to increase the speed of mathematical operations. The mass storage devices of the computer are a 720 kbyte, 3.5 inch floppy disc and a 20 Mbyte hard disc. 640 kbyte of base memory and 384 kbyte of extended memory are resident and an EGA card and monitor included.

Software

The image analysis software was written in Lattice C code [41] and 8086/8088 assembler code [36] was used for the frame store interface routines. The program was developed under the MS-DOS operating system [42]. Version 3.30 was used and it is recommended that this be used in future to avoid compatibility problems (the programs will in fact work with any version from 3.00 to 3.30). Proprietary software used during program development was as follows:

Microsoft 8086 Macro Assembler [43]

Lattice C Compiler [41]

Plink '86 linker [44]

Pfix '86 interactive debugger [45]

Halo graphics library [46]

3.3 COMPARISON OF ANALYSIS TECHNIQUES

3.3.1 USE OF REAL-TIME ANALYSIS

Initial experiments were carried out using a 6 mm ball bearing as an indenter. Back-lighting was used to illuminate the guide tube in order to give strong contrast between the object (ball bearing) and background.

The grey level image, derived from the video camera, was thresholded in real time using one of the input LUTs on the frame store. This use of binary images simplified the subsequent processing. A routine was written in C code to provide grey level frequency histograms of captured images. From this information the threshold bounds were determined, manually, by the user. In the resulting binary images white pixels were chosen to represent the object and black chosen for the background.

Real-time binary images of the indenter motion were then recorded onto video tape and later analysed under slow-motion playback. This produced two observations. The first was that the ball bearing travelled too fast on the initial descent to be detected by the camera. The second was that only the extremes of the object motion (at the top of the

trajectory and at the impact point) were visible because of motion blur. It was recognised that by exclusive-OR'ing consecutive binary images in real time the extremes of the object motion would be revealed as clusters of white pixels. The rebound height could then be determined from the location of the white pixel cluster having the largest ordinate.

The real time exclusive-OR (XOR) operation was accomplished with the aid of an input LUT (see table 3.1) and the following sequence of operations. A frame is first captured and thresholded with bit planes 0, 1, 2 and 3 (least significant nibble) of the frame store image buffer write-protected. The frame is then fed-back through the XOR LUT with no write protection on the image buffer and the sequence repeated for each newly captured frame. Note that because each XOR operation occupies an entire frame period the effective refresh rate is halved from 25 frames/ sec to 12.5 frames/ sec.

Input	Output
00000000	00000000
00001111	11111111
11110000	11111111
11111111	00000000

table 3.1: Exclusive-OR Look-Up Table Mapping

The scheme gave good results using the 6 mm ball bearing indenter but failed to work using a 3 mm ball bearing in conjunction with the test rig mentioned earlier. This was mainly because the smaller indenter covered such a small area of the field of view that it was almost impossible to determine a threshold from the grey level histogram; the mode corresponding to the object being "swamped" by the background. In addition, changes in ambient light intensity over the object path meant that a single threshold

could not be used reliably for the entire image. The back lighting approach was later abandoned because of these problems.

A possible solution was sought by altering the illumination so that the object appeared brighter than the background. This was effected by placing an incoherent light source at the top of the guide tube. Under these conditions the trajectory appeared as a bright line because of motion blur (due to phosphor persistence in the vision camera). The set-up also produced large amounts of reflection from the guide tube wall, a problem later solved by using a collimated laser light source. Using this illumination method the initial descent of the indenter could now clearly be seen and in to order calculate the rebound height, it was necessary to start the real time exclusive-OR operation after the initial rebound had occurred. Otherwise the subsequent indenter motion would have been masked by the residual image of the initial descent.

To detect the initial impact a window of pixels close to the bottom of the trajectory was scanned and the XOR operation "triggered" when the number of white pixels in the window had exceeded a certain threshold. The resulting image should then have contained a vertical white line, corresponding to the object in motion. The size of this line is proportional to the rebound height and hence the Shore number.

To ensure accurate results it was necessary to perform the triggering operation in real time and this required that all pixels in the window be scanned during a single video blanking period (630 μs for the frame store used). This proved difficult to achieve in practise since the frame store image memory is mapped to the the extended memory of the host computer and thus each read/ write operation requires use of a Basic Input/ Output System (BIOS) function call. The BIOS call employs a low level interrupt and therefore each read/ write operation incurs a significant time penalty. Such was the

delay that the window scanning operation could not be performed during a single blanking period and hence not in real time.

Instead of accessing the image buffer on a "byte-by-byte" basis a contiguous block of data can be transferred using a single BIOS call thus reducing the read/ write overhead. Furthermore, because the video signal is digitised line by line and the data written to consecutive memory locations, each row of pixels may be read as a block using a single BIOS call. To take advantage of this fact the video camera was rotated through 90 degrees to make the object motion appear horizontal. The window scanning operation could then be replaced by scanning a row of pixels containing the object motion. In practise it was necessary to scan several rows since geometric distortion caused by the zoom lens made the straight line trajectory appear slightly curved.

The scheme was eventually made to work in real time. However even small variations in ambient light intensity were enough to cause false triggering giving unreliable results. This led to the real time processing approach being curtailed in favour of the scheme outlined below.

3.3.2 USE OF NON-REAL TIME ANALYSIS

Having recognised that real time processing of the image data was unworkable with the present system, a new approach was adopted in which the camera and illumination set-up were kept as before. In order to analyse the image data *a posteriori* it would have been necessary to store each and every video frame for the entire duration of the motion. The amount of data in each frame is 256 kbyte and with at least 50 frames needed to record the entire event, the amount of data is clearly far in excess of that which can be

stored within the host computer. In each frame however, the object motion is contained entirely within a few neighbouring rows of pixels. If just the data contained in these rows is extracted from each frame then the storage requirements are reduced dramatically and a non-real time approach then becomes a viable proposition.

In the scheme originally used four rows of pixels (2 kbyte of data) are read from the frame store after each odd field sync and stored as a contiguous block in the host computer base memory. This operation can be completed within a video blanking period since a single BIOS call is used to transfer each set of data, thus giving real time operation at 25 frames/sec. The data is at present thresholded in real time using an input LUT, however this operation could be carried out later allowing automatic/adaptive methods.

At present the indenter is released by hand and the analysis program started by the operator. After all the data has been collected corresponding pixels within rows belonging to the same frame are OR'ed together (Fig. 3.3) to overcome any geometric distortion effects (see section 3.3.1). This means that the location of the indenter can be determined if it's image is present within any of the group of four rows read from each captured frame (in practise it was found that four rows were more that enough to span any apparent curvature in the indenter trajectory caused by geometric distortion). The size of the block of stored data in base memory was, arbitrarily, limited to 256 kbyte giving an elapsed time of approximately 5 seconds; this was more than enough to cover the entire event. The scheme, in it's final form, was improved on so that even and odd fields are separated giving twice the effective refresh rate. Assuming that the geometric distortion is the same in both the vertical and horizontal directions, the maximum change in the indenter position due to this distortion is equal to four rows of the image array. This corresponds to an error of $(4/512) \times 250 \approx 2$ mm when the entire test rig is

in view since the frame store resolution is 512 x 512 pixels and the field of view is approximately 250 mm.

Another potential error is caused by the automatic gain control (a.g.c.) in the camera. Since the laser occupies only a small part of the captured image, the average intensity level will be low and the gain will be boosted producing blurring of the laser spot. It is reasonable to assume that the degree of blur is equal at the top of the indenter trajectory and at the rest position since the indenter velocity is zero in both cases. Since the analysis program uses the distance between the *top of the indenter* at rest and the *top of the indenter* at the top of the trajectory in calculating the Shore number, the results should be unaffected by this phenomenon.

If the 256 kbyte block of data is then transferred from the host computer memory to the frame store image buffer the output image can be viewed as a series of "snap shots" of the motion laid side by side. The result is equivalent to moving the camera sideways relative to the test rig at constant velocity and appears as a graph of object height versus time (Fig. 3.4). From this representation the individual rebounds can clearly be seen (Fig. 3.5).

Prior to the ball bearing entering the guide tube, laser light reflected from the base of the stand produces a horizontal line indicating no vertical displacement with time. This line disappears when the ball enters the tube and the parabolic motion of the object becomes apparent. As the ball loses energy a continuum is eventually reached which appears as a thick horizontal line indicating the ball's rest position.

To calculate the rebound height the analysis program first extracts the motion information by windowing the image (Fig. 3.6). The lower limit of the window

corresponds to the top of the ball bearing at rest and provides a zero reference point in calculating the rebound height. This bound is located by scanning the image row by row from the bottom, calculating the number of white pixels in each row*. When this number exceeds 25% of a row (128 pixels) the position of the base is judged to have been found. The process is then continued until the fraction of white pixels in a given row falls below 25%, this gives the lower window bound.

The left hand limit of the window corresponds to the initial descent of the ball and is located by scanning the image column by column from left to right. The required column is taken to be the first one containing 10% or more white pixels. If no column satisfies the criterion an error is flagged and subsequent analysis is abandoned. Note that the initial descent of the indenter should always be in the path record if the thresholding process is working correctly (see Appendix I).

The point at which the ball comes to rest (i.e the point at which the trajectory becomes a continuum) is used as the right hand window bound. This is found by examining a row of pixels mid-way between the base level and the zero reference level. Groups of 50 contiguous pixels are examined in turn, moving from left to right across the row. When the fraction of white pixels in a given group exceeds 80% the starting point of the group is taken as the right hand window bound. If no such condition is found a worst-case estimate is used (see Appendix II). The window is finally completed by taking the top of the image as the upper bound.

---

* *Note that in the following discussion the rows and columns referred to are in fact transposed compared to the frame store.*

Any noise in the image, caused by reflections from the sides of the guide tube, is then removed. Such noise appears as horizontal lines (since the reflections are stationary) and is extracted by scanning the windowed image row by row and deleting all white pixels from rows containing more than 12.5% white pixels.

A simple gradient method is then used to calculate the rebound height. First each column inside the window is examined in turn and the co-ordinates of the upper-most white pixels recorded into a pair of arrays (one for x and one for y). The differences between successive pairs of elements in the y array are recorded into another array. This difference set is then scanned until three consecutive positive values are found (indicating the rising slope) and then further scanned until three consecutive negative values are found (indicating the falling slope). Between these two locations the element having the smallest difference value is located. This element corresponds to the turning point and hence the y value of it's co-ordinate less the zero reference value gives the rebound height in pixels.

3.4 DESCRIPTION OF THE VISION SYSTEM SOFTWARE

The image processing software is divided into two main parts, a set of routines written in 8086 assembler language which provide a low level interface to the frame store and a C program which implements the image analysis functions. Before describing the program code in detail it is necessary to outline the frame store architecture and the method in which the assembler and C programs are linked together.

The frame store contains two 256 kbyte image memories which are at present mapped to the host computer memory at locations A00000H - A3FFFFH (buffer 0) and A40000H - A7FFFFH (buffer 1). The flow of data to and from the image buffers is controlled by eight 16 bit registers on the frame store which are mapped to consecutive memory locations in the host computer base memory. At present these locations start at 00250H. The function of each of the registers is described in Appendix IV.

8086 assembly language code was written to manipulate the registers in order to control the frame store. The code consists of a set of routines which can be called from the C program as user defined functions and is described in Appendix V. A detailed description and listing of the C program in given in Appendix VI. The mechanism by which parameters are passed to and from the assembler code is described fully in the Lattice "C" Language Reference Manual [41], however an outline is given below.

On encountering the function call the C program first pushes the function arguments onto the stack in reverse (right to left) order. The return address is then pushed onto the stack. Here the address is four bytes long since the Large C model is used, thus requiring FAR type jumps. It is the responsibility of the called, assembler, routine to save the caller's base pointer (BP) register by pushing the contents of BP onto the stack. The BP contents are then replaced by those of the stack pointer (SP) register so that the function arguments may be accessed using indexed addressing. Before returning from the called function the assembler code must restore the caller's BP register contents by popping the stack. If the function is to return a result to the calling program, then the result must be placed in predefined registers. The size of the value returned determines the register(s) used as follows:

8 bits	AL

16 bits	AX
32 bits	(AX,BX)
64 bits	(AX,BX,CX,DX)

The main() function of the C program (Fig. 3.7-3.10) consists of a large switch construct which transfers control according to options entered by the user. The 'r' case produces a real time image on the output monitor by calling the camera() function. The 'p' case produces another menu selection for the user by calling the parameters() function. The 'm' and 'c' cases produce the measurements and calibration menus by calling the measmenu() and calibration() functions respectively. A detailed description of each of the C program functions is given below.

3.5 SET-UP AND USE OF THE VISION SYSTEM

3.5.1 SETTING UP THE HARDWARE

In order to produce good results the camera and test rig need to be carefully set up. The two components should be placed on a flat surface with the camera mounted horizontally (i.e. with the image on the monitor apparently rotated by 90 degrees) about 1.5 m away from the test rig. The test piece should then be placed under the glass slide and the indenter dropped into the guide tube. Laser light can be introduced via a fibre optic placed at the top of the guide tube. The camera lens should then be adjusted so that the entire guide tube and scale are focused and in view.

3.5.2 RUNNING THE SOFTWARE

The image processing program is executed by typing

shore <enter>

This displays the main menu from which the user should select the "display Real time image" option in order to produce a grey scale view of the test rig on the frame store output monitor.

The next step is to call up the Calibration menu and select the "set Scaling" option. This is used to calibrate the camera against the scale on the test rig. The user may set the scaling factor either by entering the known value in mm/pixel or by placing two cursors at known points on the test rig scale, as seen on the output monitor. In the latter case some alteration of the illumination (e.g. use of back-lighting) may be necessary to produce a clear image of the scale on the output monitor.

The next stage involves setting up the threshold limits, the option for which may be found in the Parameters menu. Choice of limits may be aided by selecting the "Histogram" option which gives a plot of grey level frequency for the image. The results of this may, however prove inconclusive; this is because the object (indenter) fills such a small part of the captured image that it is difficult to delineate those grey levels corresponding to the object from those of the background. Here a plot of grey level along a row or column may be more useful. This can be achieved by selecting the "Grey level cross-section" option.

By selecting the "alter Thresholds" option the threshold bounds can be set and a real time binary image produced. Note that the program allows for "fine tuning" of the thresholds limits.

Once this operation is completed, the set "Centre line" option should be selected. This option can be found in the Calibration menu and is used to record the position of the guide tube centre line. All that is required of the user is to place a cursor at the centre of the ball bearing indenter as it appears on the monitor.

In order to take results, the Measurements menu should be called up. This gives the option of taking either a single measurement or a group of measurements (the group mean and standard deviation being given in the latter case). In both cases the user is instructed to press a key on the computer and immediately drop the indenter into the guide tube to start the analysis.

The results of either a single measurement or a group measurements of may be recorded onto the hard disk by selecting the "Write results to file" option. The user is then prompted for a name for the results file, including the extension. If the given file is already present the results are appended to the existing file, if not a new file is created (note that all results files must reside in the current directory). The program first writes a description string (supplied by the user) to the file, followed by the current date and time. Finally the results, their average and their standard deviation are recorded.

Once the system has been calibrated and the thresholds set a variety of materials may be tested using just the Measurement menu options.

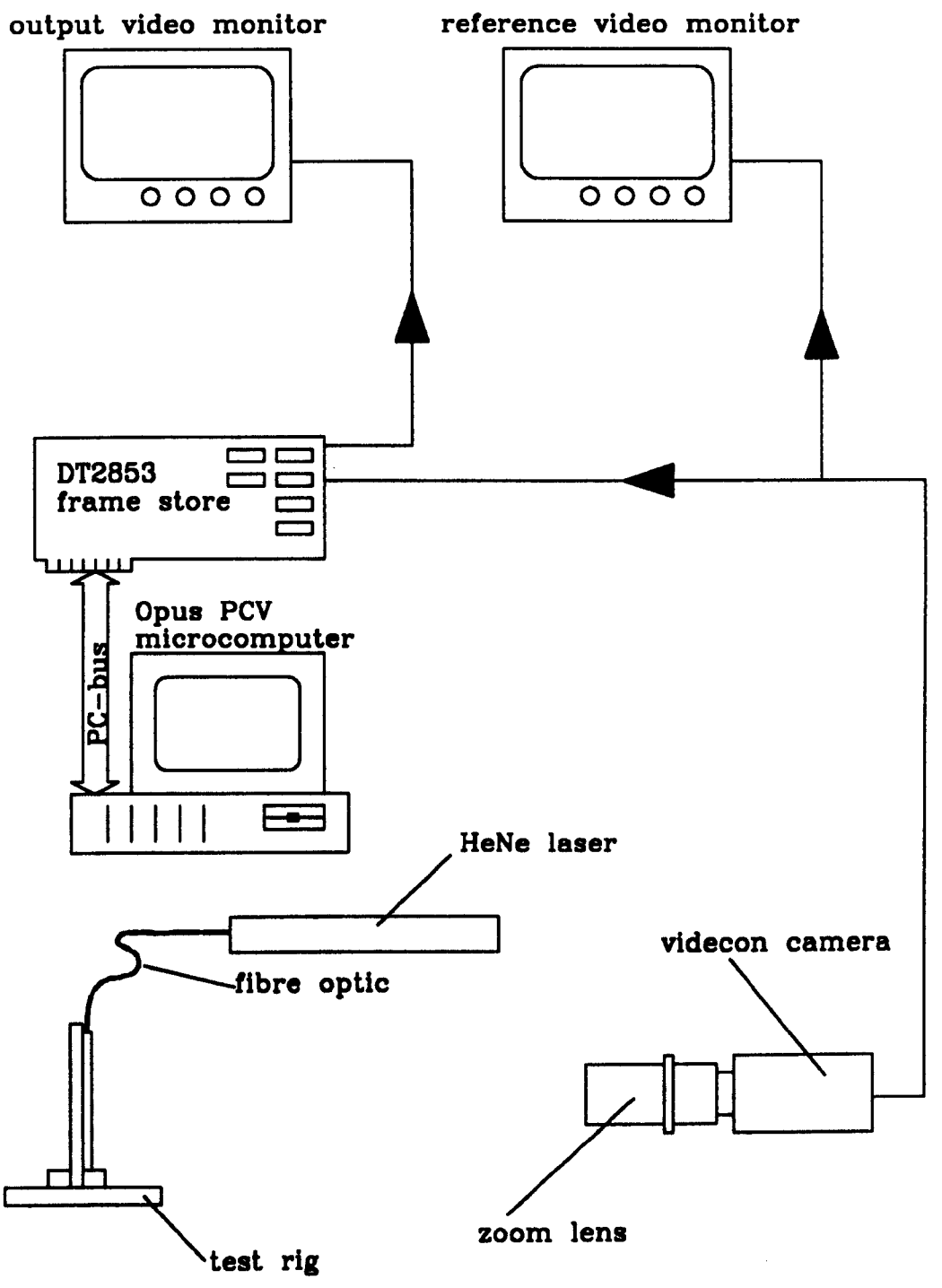


Figure 3.1: VISION SYSTEM HARDWARE

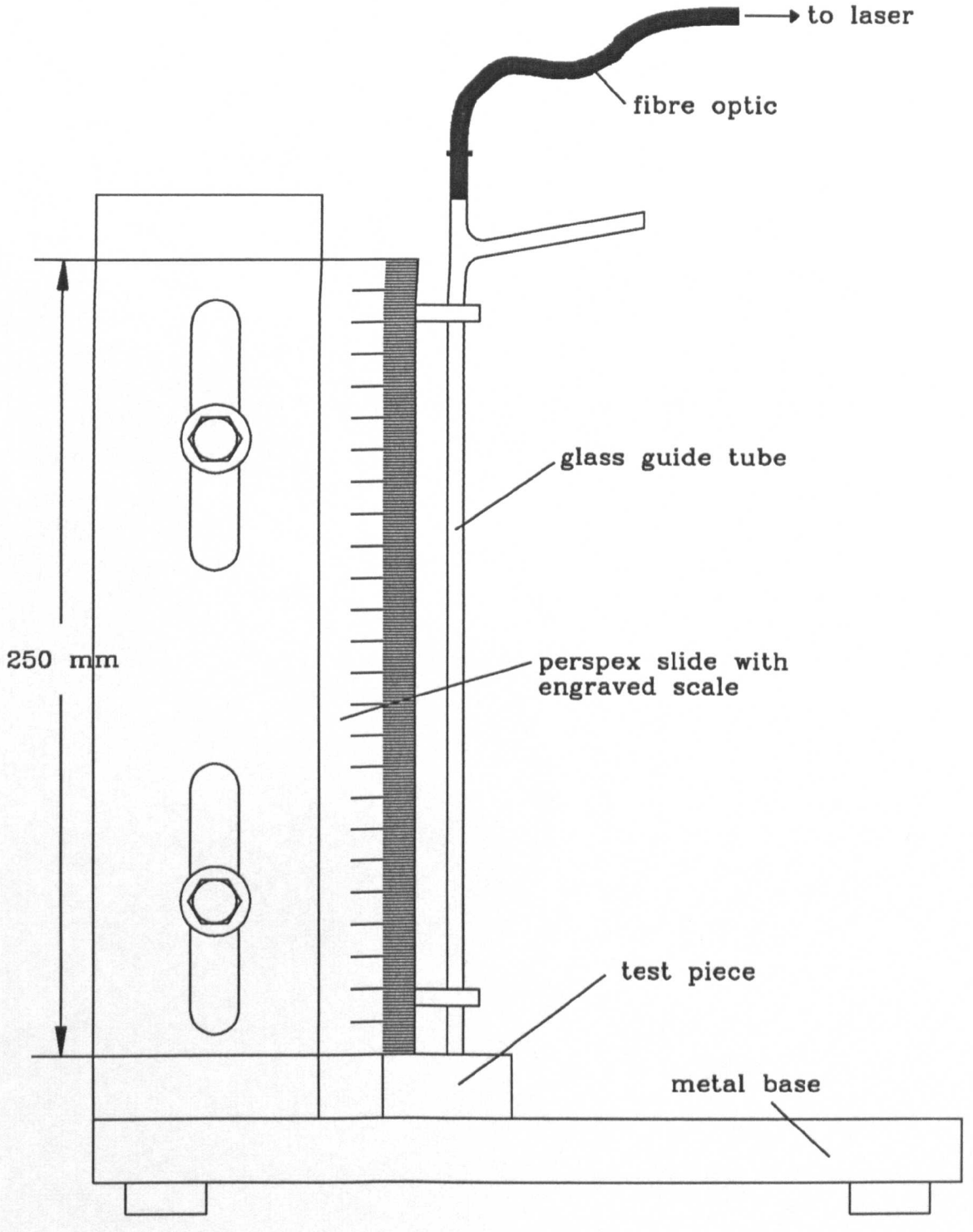


Figure 3.2: SHORE SYSTEM TEST RIG

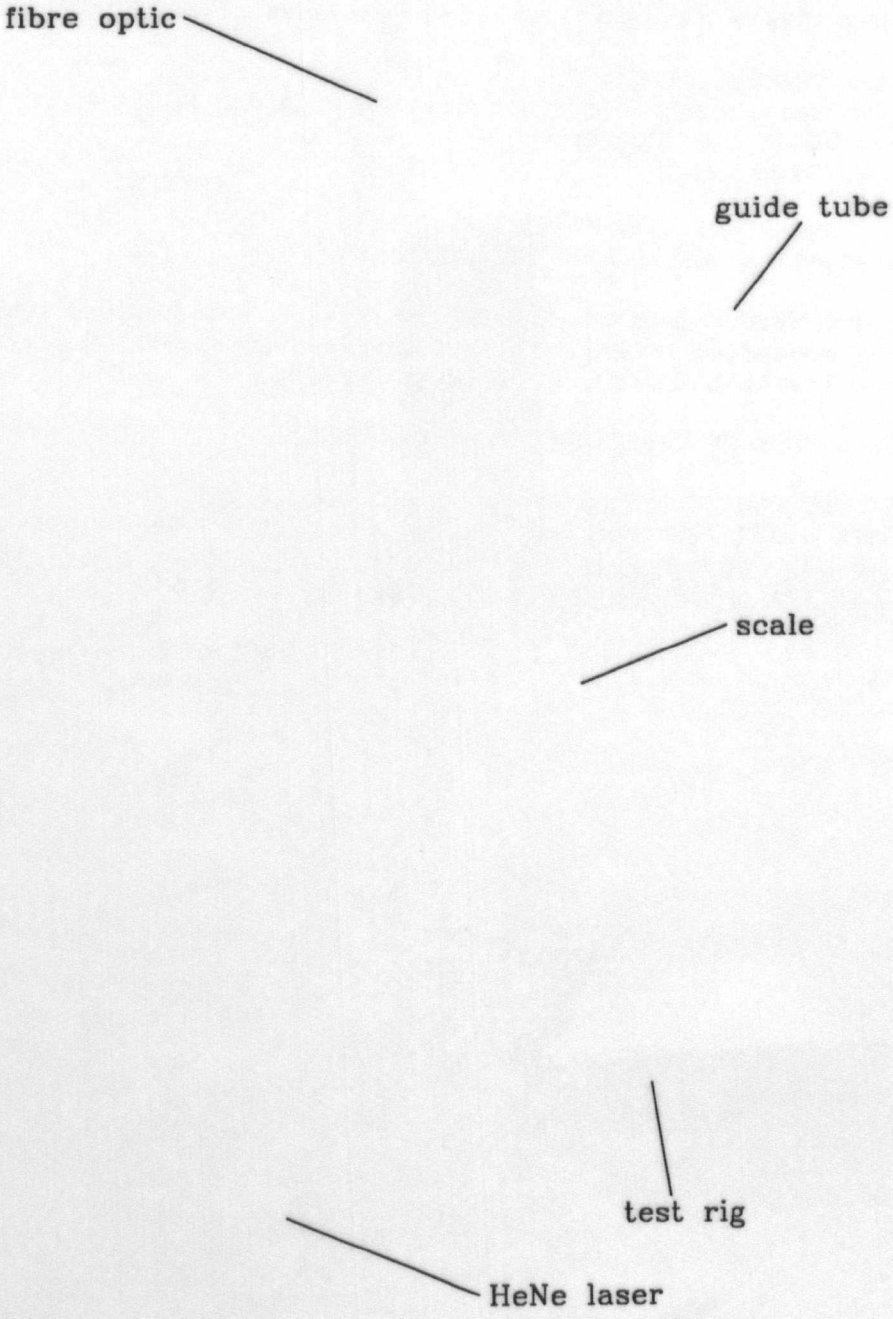


PLATE I: SHORE SYSTEM HARDWARE

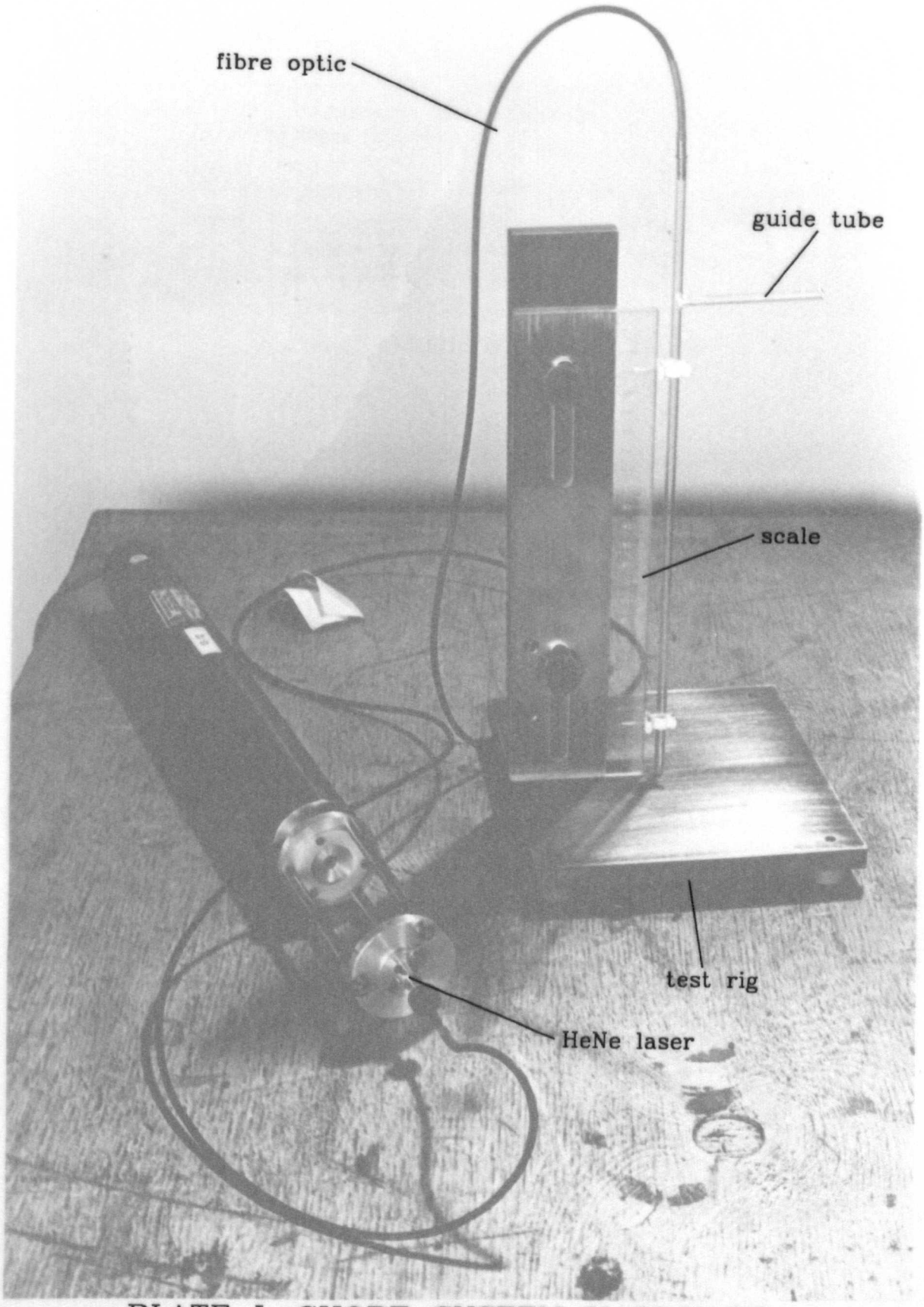
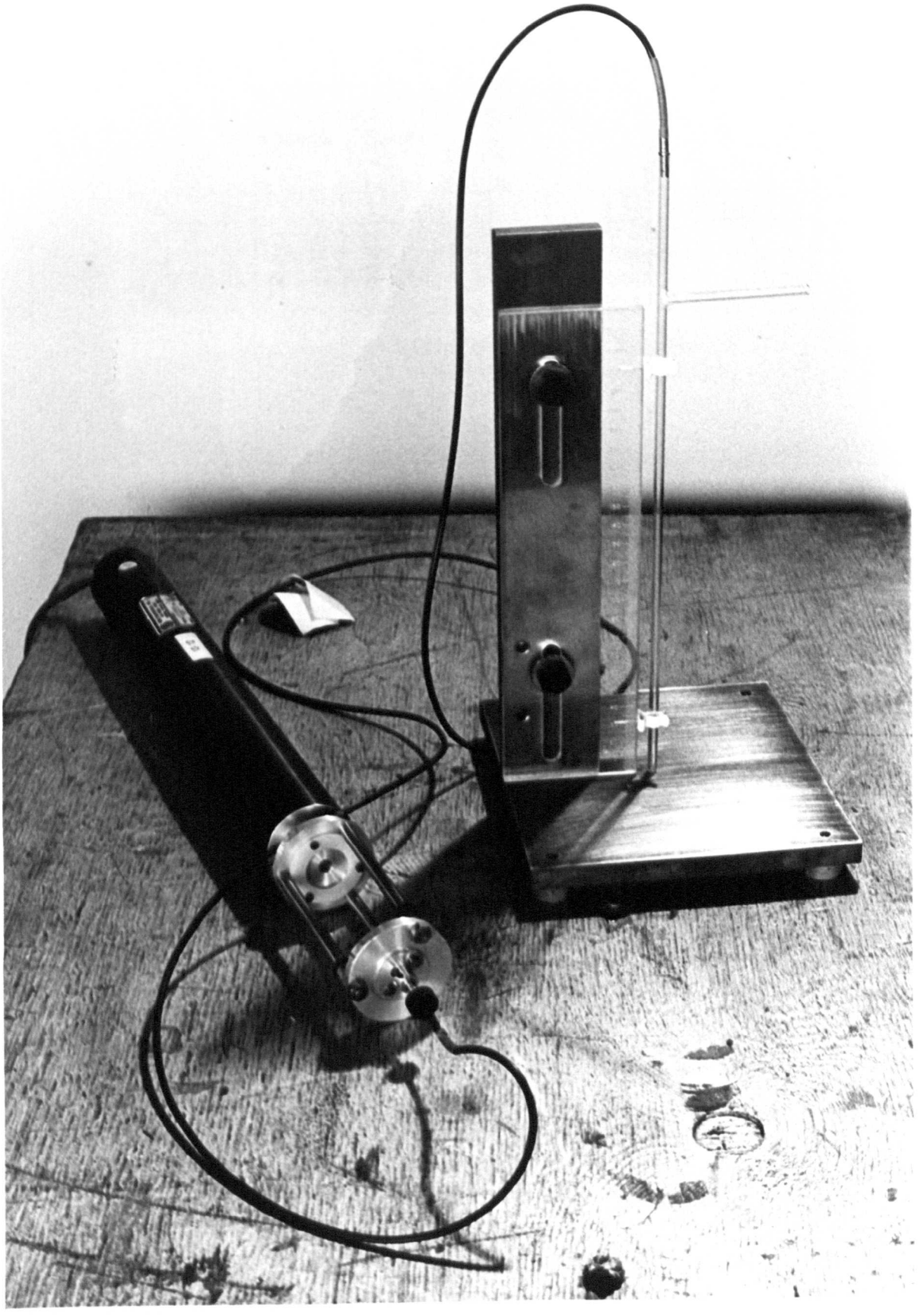
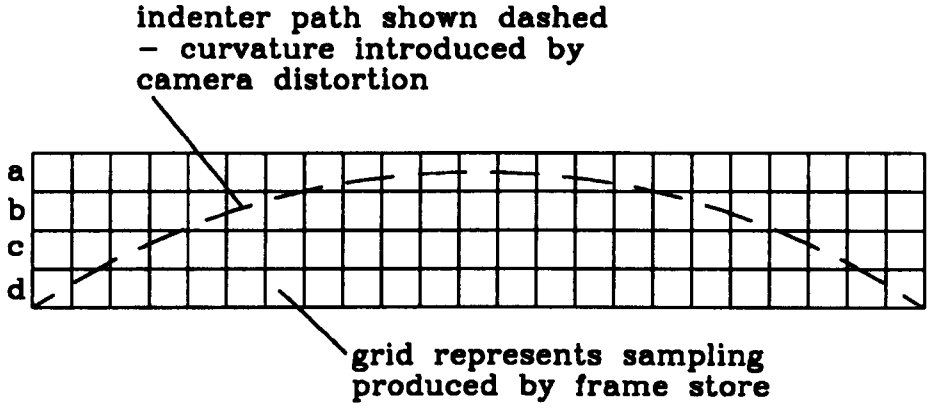


PLATE I: SHORE SYSTEM HARDWARE





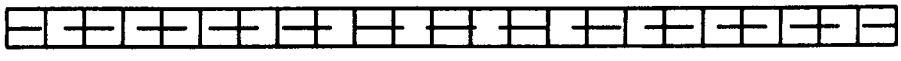
(a) original image

a	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
b	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
c	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
d	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

(b) result of thresholding image –
1's represent object pixels, 0's represent background pixels

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(c) result of OR'ing rows a,b,c,d



(d) equivalent original image to (c)

Figure 3.3: USE OF OR'ing TO COMPENSATE FOR GEOMETRIC DISTORTION

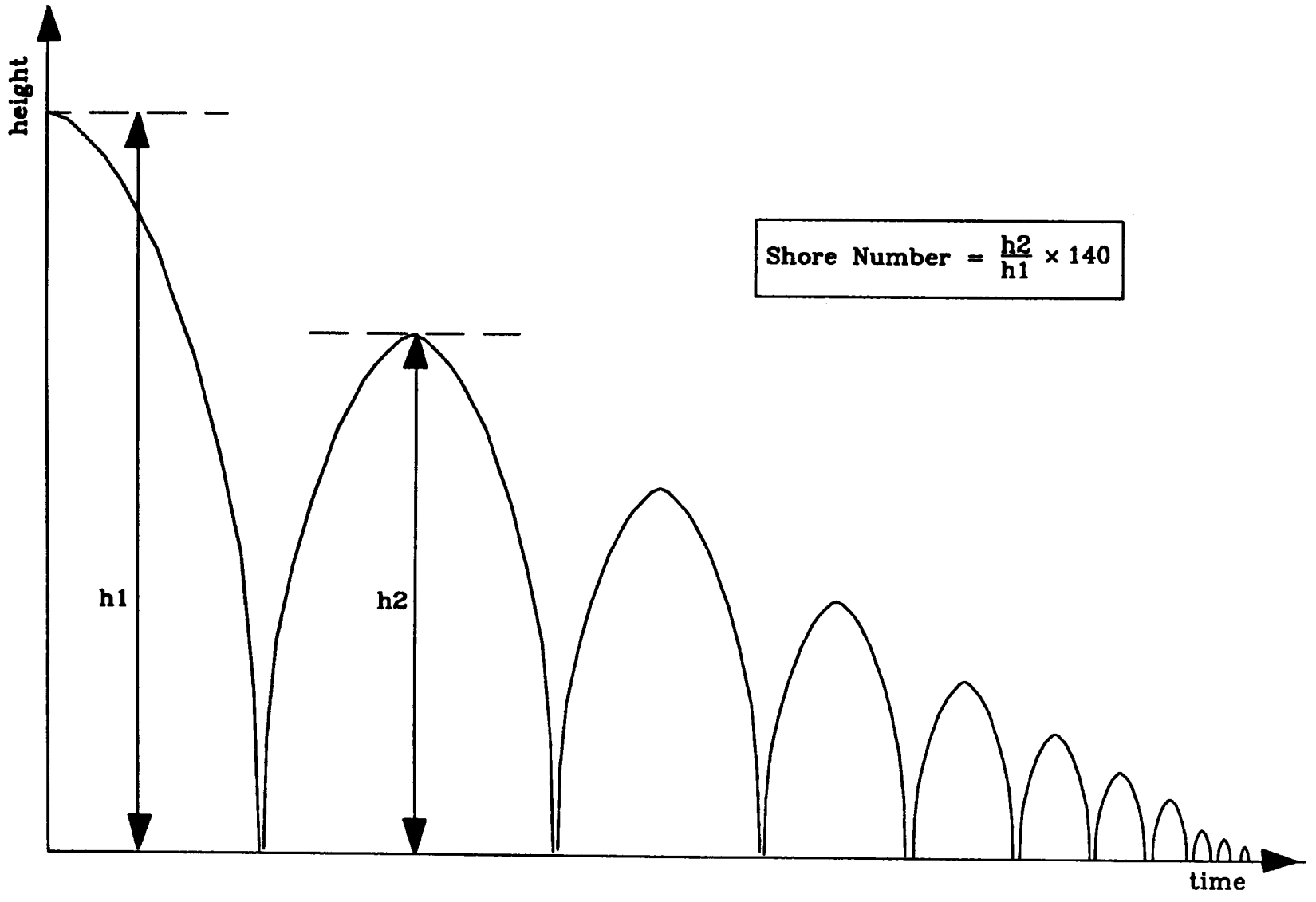


Figure 3.4: IDEALISED INDENTER MOTION

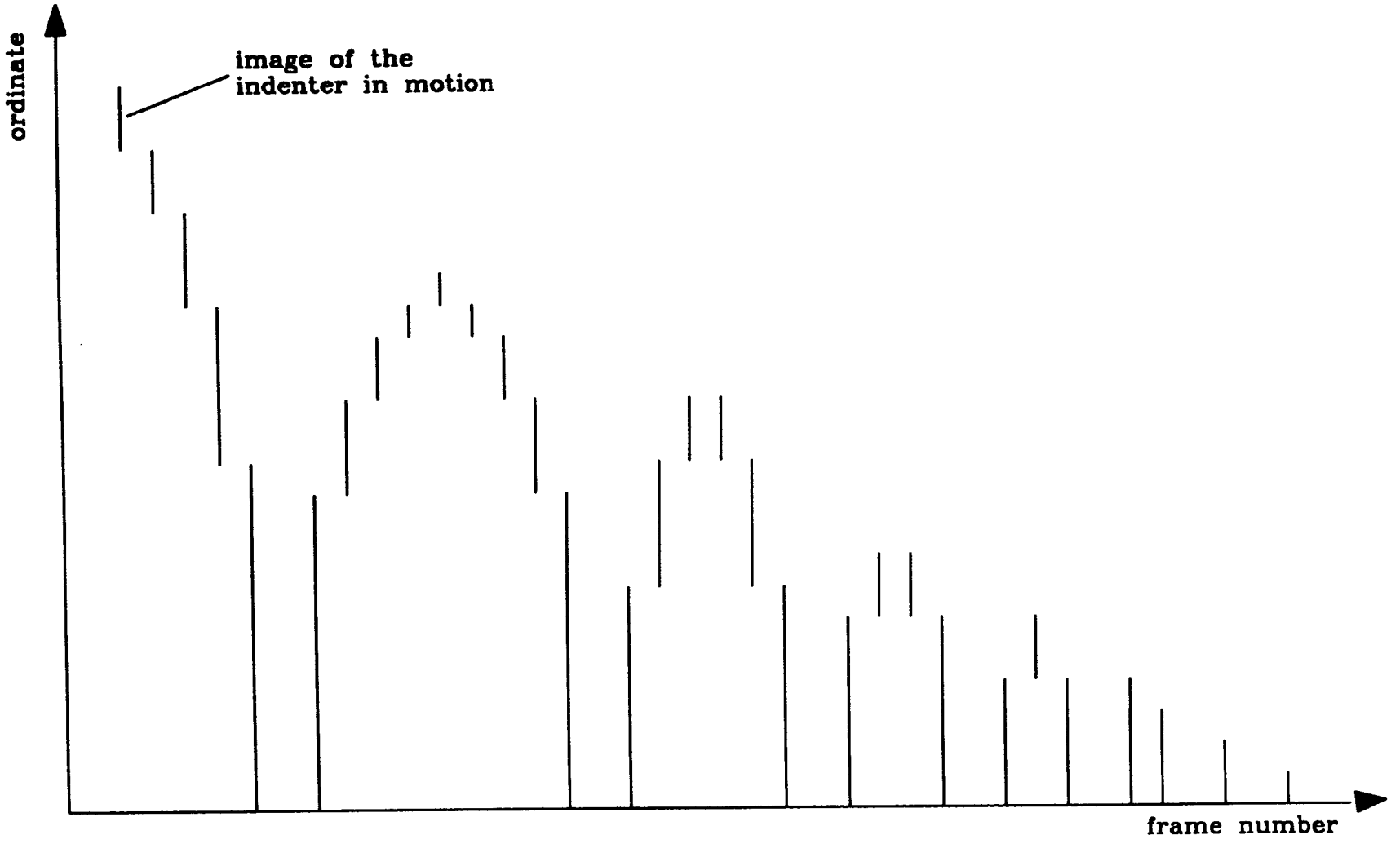


Figure 3.5: IDEAL PATH PRODUCED BY FRAME STORE

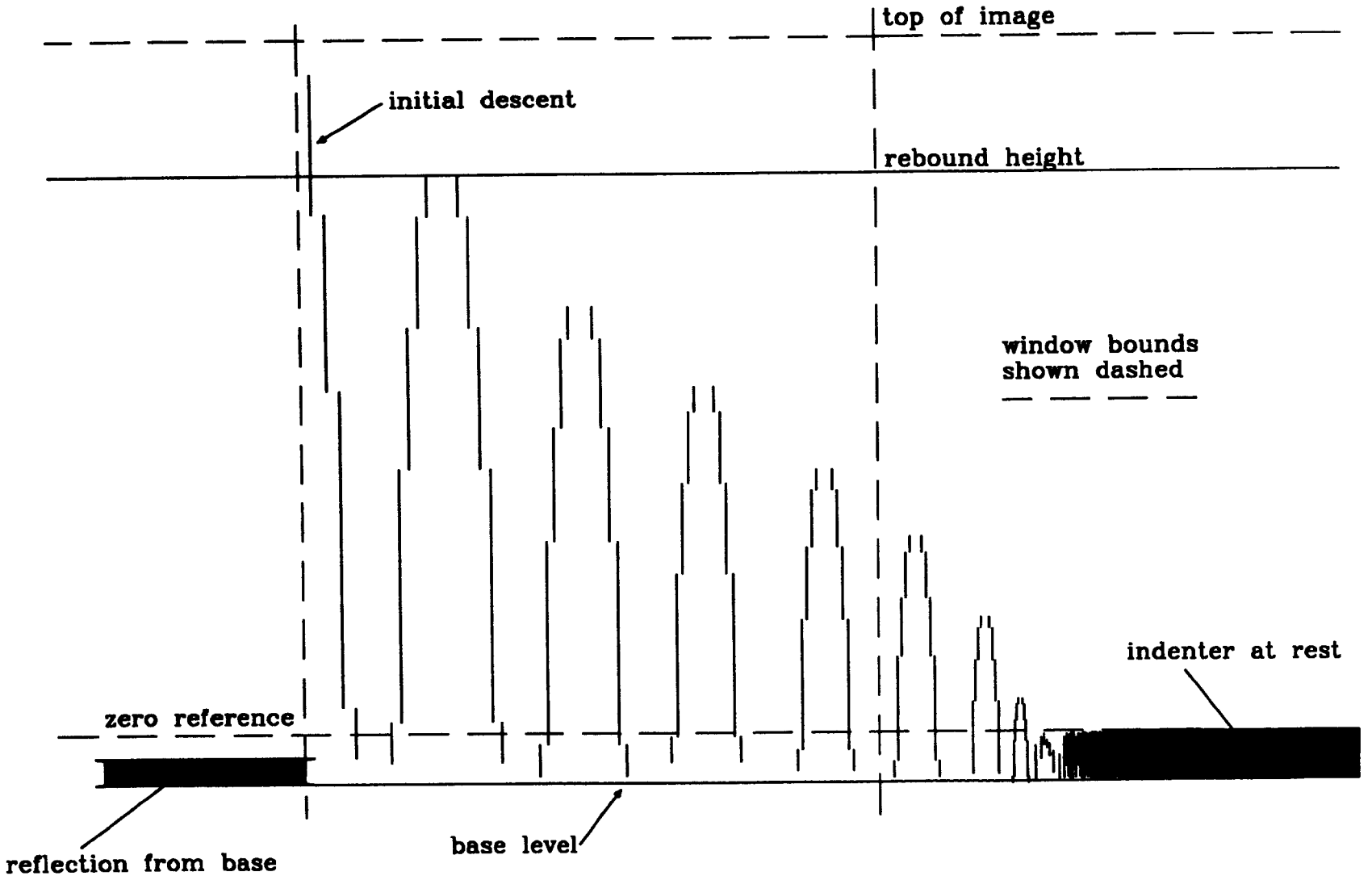


Figure 3.6: TYPICAL PATH RECORD PRODUCED BY ANALYSIS PROGRAM

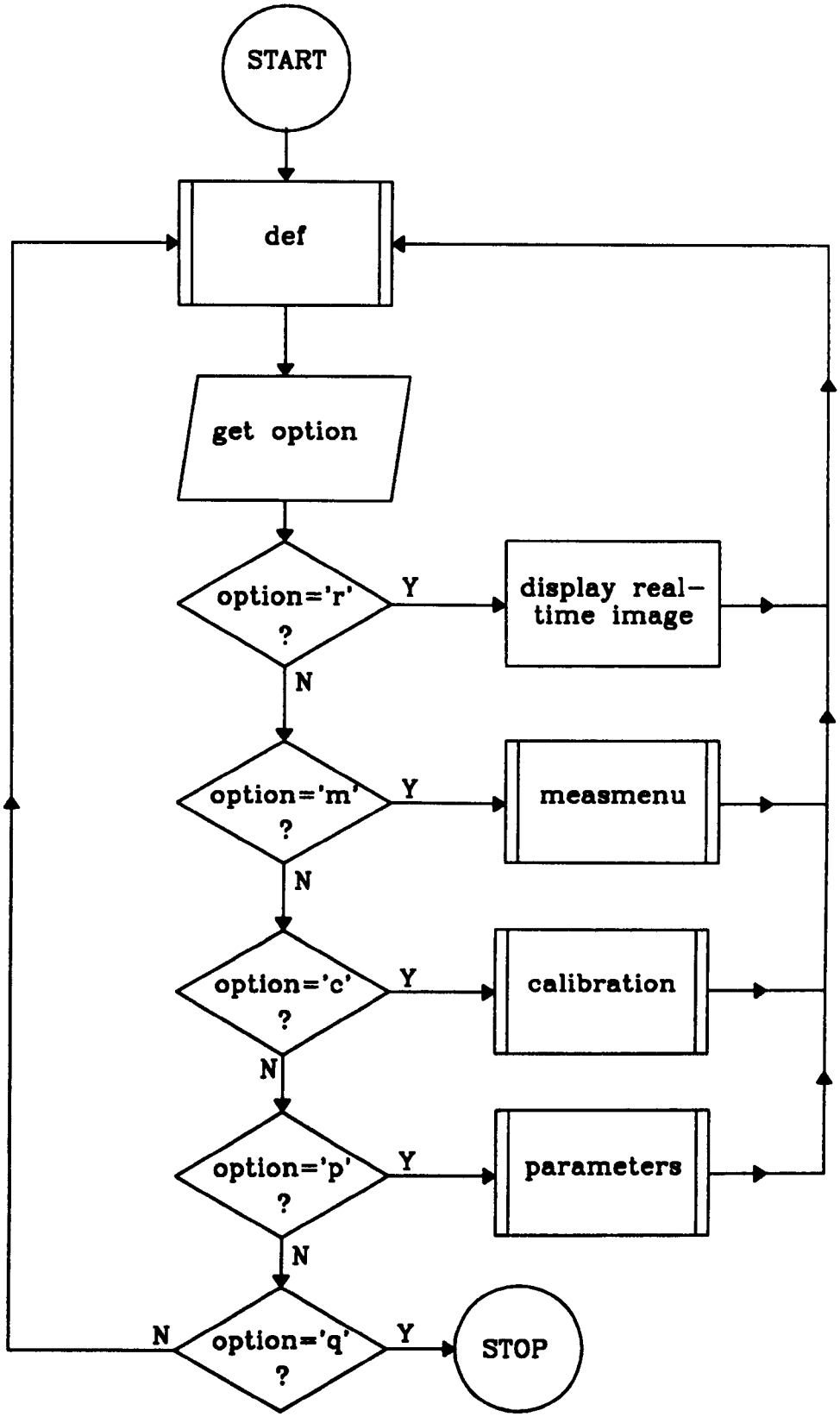


Figure 3.7: FLOWCHART FOR SHORE TEST PROGRAM

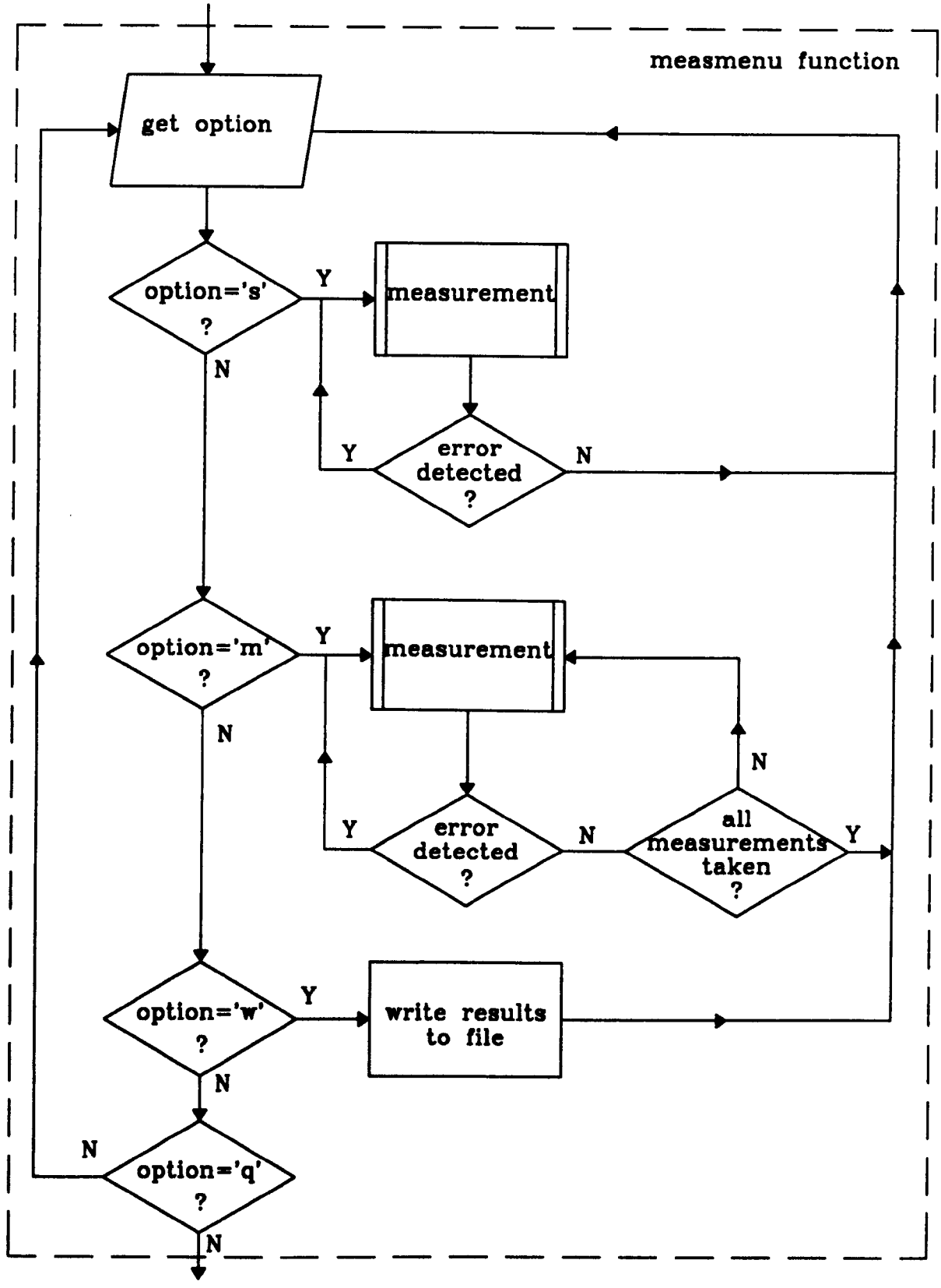


Figure 3.8: FLOWCHART FOR SHORE TEST PROGRAM /contd.

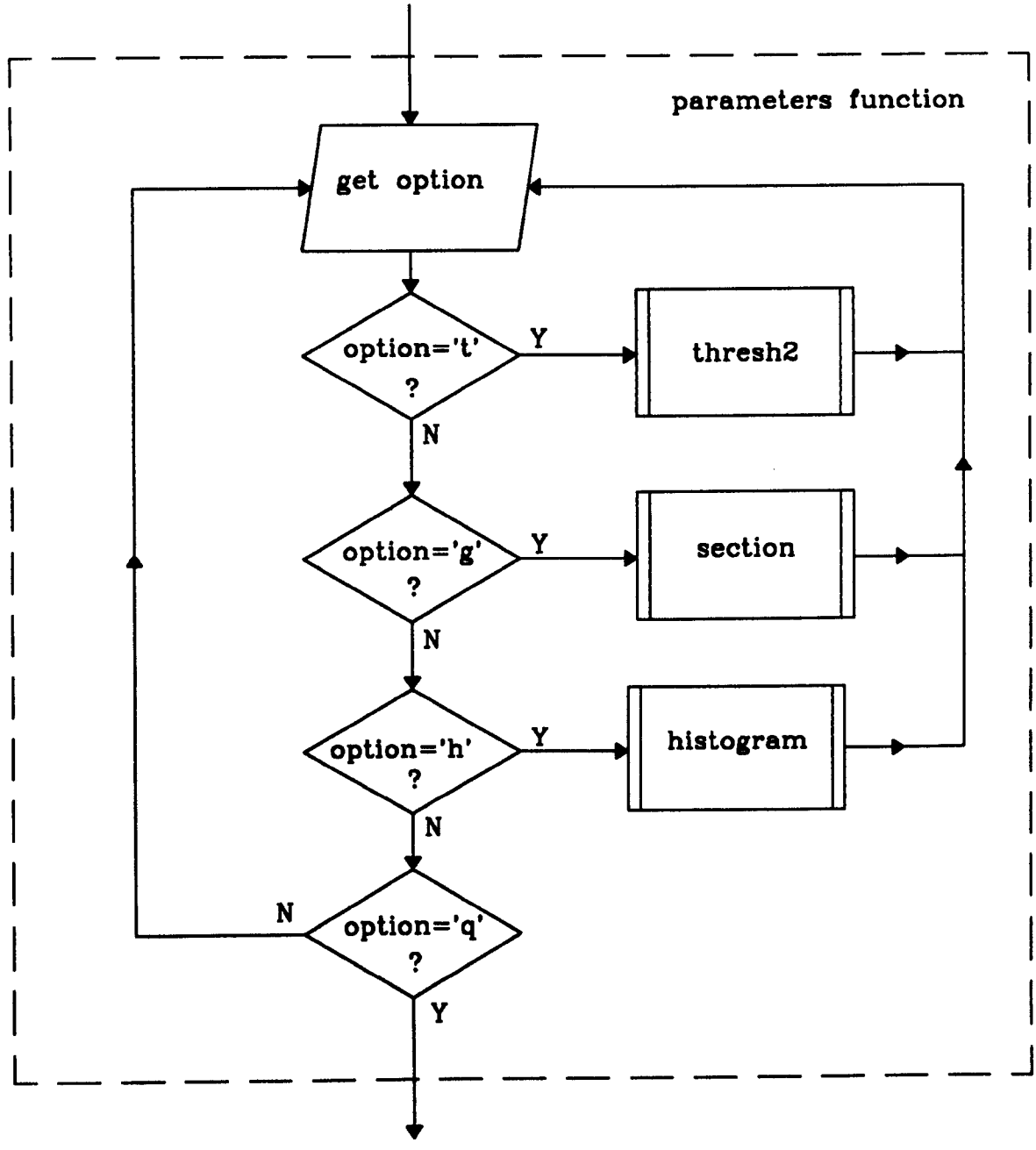


Figure 3.9: FLOWCHART FOR SHORE TEST PROGRAM / contd.

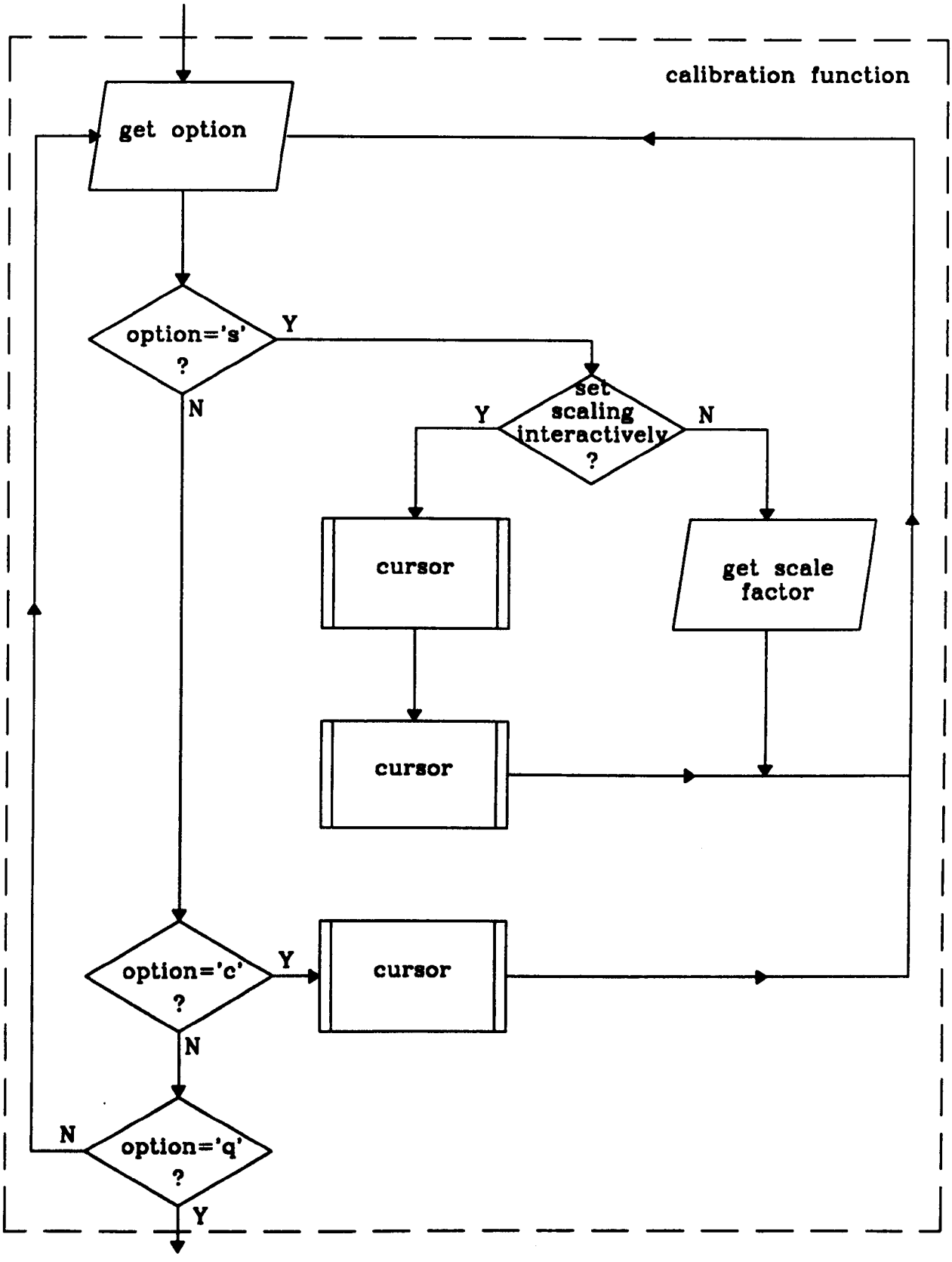


Figure 3.10: FLOWCHART FOR SHORE TEST PROGRAM / contd.

CHAPTER 4

VICKERS HARDNESS TEST

4.1 INTRODUCTION

This chapter describes the development of a vision-based system for the automation of the Vickers hardness test, outlined in Chapter 1. As with the Shore hardness, test the vision system is based on an IBM PC/AT - compatible computer, fitted with a commercially available frame store. Details of the system hardware are given in section 4.2.

The use of a 16-bit, single processor computer means that computational power is limited. To ensure reasonable cycle times, binary image processing was chosen in preference to grey scale image processing since the data processing requirements are less prohibitive (e.g. segmentation of an image can be performed in the order of seconds using binary image processing but in the order of minutes using grey scale processing with the present system). The original system software is described in section 4.3. Segmentation of the original grey scale image into a binary form is performed by a thresholding process. The required threshold bounds are found automatically using an image grey scale histogram, the software then locates the indentation and traces it's

border. From a quadrilateral approximation to the border, the size of the indentation, and hence the hardness of the test material, is calculated.

The original software was found to work well with good quality samples but less reliably with samples having poor contrast or surface blemishes. Since reliability is an important factor in automated systems, additional software was written to cope with poor quality samples; this software is described in section 4.4. It uses binary smoothing to isolate those parts of the image corresponding to the object information from any background noise. Run length image coding was used to reduce the data processing requirements and a novel algorithm was written to perform the binary smoothing of run length coded images.

The image processing software is written in the C language [35] with additional 8086 assembler code [36] used to provide a low-level interface to the frame store. A full description of the C program code is given in section 4.5. Finally, in section 4.6 a guide to setting up and operating the hardness testing system is given.

4.2 SYSTEM HARDWARE

The hardness testing system consists of seven main components (Fig. 4.1 and Plate II): the microscope, illumination source, camera, frame store, computer, monitors and software. Each of these parts is described in detail below.

Microscope

The microscope is a binocular type and incorporates a movable table on which the sample is placed. The objective lenses are mounted on a turret allowing magnifications of 1X, 3X and 10X to be selected. A semi silvered mirror directs illumination onto the sample and transmits the reflected light from the object into the eye piece. This is a dark field illumination set-up [23] since the bright parts of the image are caused by specular reflection of the incident light from the surface of the sample and the dark parts of the image result from scattering of the incident light by the indentation. The term scattering is actually a misnomer since the indentation consists of four highly reflective facets which reflect light at an acute angle to the test piece surface. The illumination set up shown in Fig 2.5(c) was not viable since the small depth of field meant that the image was not in focus over the entire field of view. Also, the bright field technique, shown in Fig. 2.5(b), could not be employed since the illumination source would need to be rotated in order that light from all four indentation facets is reflected into the camera. The depth of field of the microscope is of the order of 5 μm and the field of view approximately 1 mm. There is no significant advantage in the use of a CCD video camera in this application.

Illumination

Illumination is provided by an incandescent bulb rated at 12V/55W DC. The bulb is small enough to be mounted within the microscope turret. A.C. lamps are not suitable because of stroboscopic effects.

Camera

This is an Hitachi [38] monochrome vidicon camera of the type previously used in the Shore system (Chapter 3).

Frame Store

The frame store, a Data Translation model DT2853, [39] is of the same type as was used in the Shore system (Chapter 3).

Computer

The computer is an Opus PCV [40] and is of the same type as was used in the Shore system (Chapter 3).

Monitors

Two monitors are used in the system, one for the camera output and one for the frame store output. Both are Hitachi monochrome types [38] with composite video inputs and a 625 line raster operating at 50 Hz.

Software

The image analysis software was written in Lattice C code [41] and 8086/8088 assembler code was used for the frame store interface routines. The program was developed under the MS-DOS operating system [42]. Version 3.30 was used and it is recommended that this be used in future to avoid compatibility problems (the programs will in fact work with any version from 3.00 to 3.30). Software used during the development of the program was the same as that used for the Shore system.

4.3 SOFTWARE OVERVIEW

4.3.1 INTRODUCTION

The process by which the diagonal lengths of the indentation are extracted from the image data can be broken down into seven main stages:-

(1) The grey level image is thresholded giving a binary (black and white) representation.

(2) The rough centre of area of the indentation is calculated using the whole of the image data.

(3) The pixels forming the boundary of the object are located and their position recorded.

(4) A more accurate centre of area calculation is made based on the boundary pixels.

(5) The distance from the centre of the area to each of the boundary pixels is calculated, from this data the position of each corner and it's distance from the centre is determined.

(6) The mean lengths of the diagonals are calculated taking into account the camera aspect ratio and scale factor (pixels / mm).

(7) From the specified load the hardness is then calculated.

The following sections contain a more detailed description of each of the stages.

4.3.2 AUTOMATIC THRESHOLDING

In converting the grey scale image to a binary representation the threshold limits are

automatically calculated using a grey level histogram. Typically the histogram is bi-modal with one mode corresponding to the indentation (object) and the other to the background. The relative size of the peaks is determined by the size of the indentation.

The lower threshold is taken to be zero grey level, since the pixels corresponding to the object will always have the lowest grey levels in the image. The upper threshold is taken to be the grey level having minimum frequency between the two peaks, since this delineates the grey levels of pixels belonging to the object from those of the background.

In order to determine the upper threshold, it is first necessary to locate both histogram peaks. The first peak is found simply by calculating which grey level has maximum frequency in the histogram, to locate the second peak requires a more sophisticated scheme. A portion of the histogram, called the search interval, which extends from the grey level corresponding to the first peak to some lower grey level is examined (Fig. 4.2). The grey level having the maximum frequency and the grey level having minimum frequency within this interval are then located. If the grey level having minimum frequency is between the grey level having maximum frequency and the grey level corresponding to the first peak then a local minimum has been found, the grey level of which is used as the upper threshold. If this is not the case then the process is repeated with ever increasing search intervals until either a minimum is found or the search interval extends as far as the lowest permissible grey level. If no minimum, and hence no threshold, can be found (Fig. 4.3) then the entire algorithm is repeated but with the search interval now extending from the grey level corresponding to the first peak to some higher grey level (Fig. 4.4). This an adaptation of a scheme suggested by Rosenfeld and Kak [24].

In practise it was found that small fluctuations in the histogram in the vicinity of the first peak caused erroneous threshold selection by the algorithm. To overcome this problem a range of grey levels, called the exclusion interval, which extend from the grey level corresponding to the first peak to some lower grey level, and similarly from the first peak to some higher grey level, are not examined.

The algorithm can be expressed thus:-

(1) locate the first mode at grey level g_1 - find g_i such that:-

$$\forall g_r,\ g_i \neq g_r,\ g_{\min} \leq g_i, g_r \leq g_{\max}$$
$$n(g_i) \geq n(g_r) \qquad [4.1]$$

where n(g) is the number of pixels having grey level g

(2) initialise the size of exclusion interval, g_e, and the size of search interval interval g_s

(3) find the grey level having maximum frequency in the search interval, g_2 - find g_i such that:-

$$\forall g_r,\ g_i \neq g_r,\ g_1 - g_e - g_s \leq g_i, g_r \leq g_1 - g_e$$
$$n(g_i) \geq n(g_r) \qquad [4.2]$$

then set $g_2 = g_i$

(4) find the grey level having minimum frequency in the search interval, g_3 - find g_i such that:-

$$\forall g_r,\ g_i \neq g_r,\ g_1 - g_e - g_s \leq g_i, g_r \leq g_1 - g_e$$
$$n(g_i) \leq n(g_r) \qquad [4.3]$$

then set $g_3 = g_i$

(5) increase the size of the search interval:-

$$g_s \leftarrow g_s + \Delta g \qquad [4.4]$$

(6) repeat steps (3) to (5) while the following condition holds:-

$$[(g_3 \leq g_2) \cup (g_3 \geq g_1 - g_e)] \cap (g_1 - g_e - g_s > g_{\min}) \qquad [4.5]$$

(7) if on termination of the loop the following condition holds:-

$$g_2 < g_3 < g_1 - g_e \qquad [4.6]$$

then the upper threshold is given by g_3. However if this is not so the search must be continued on the upper side of the first mode using the following steps:-

(8) reinitialise the size of the search interval g_s

(9) locate the grey level having maximum frequency inside the search interval, g_2 - find g_i. such that:-

$$\forall g_r,\ g_i \neq g_r,\ g_1 + g_e \leq g_i, g_r \leq g_1 + g_e + g_s \qquad [4.7]$$
$$n(g_i) \geq n(g_r)$$

then set $g_2 = g_i$

(10) find the grey level having minimum frequency inside the search interval, g_3 - find g_i such that:-

$$\forall g_r,\ g_i \neq g_r,\ g_1 + g_e \leq g_i, g_r, \leq g_1 + g_e + g_s$$
$$n(g_i) \leq n(g_r) \qquad [4.8]$$

then set $g_3 = g_i$

(11) increase the size of the search interval:-

$$g_s \leftarrow g_s + \Delta g$$

(12) repeat steps (3) to (5) while the following condition holds:-

$$[(g_3 \leq g_1) \cup (g_3 \geq g_1 + g_e)] \cap (g_1 + g_e + g_s < g_{\max}) \qquad [4.9]$$

(13) if on termination of the loop the following condition holds:-

$$g_1 + g_e < g_3 < g_2$$

then the upper threshold is given by g_3., if not then the histogram is uni-modal and there is no acceptable threshold.

Note that the algorithm is only successful if the histogram is strictly bi-modal. In practise this requires that both good illumination and sample quality are present to give high contrast. A method of analysing samples exhibiting poor contrast is discussed in section 4.4.

The thresholding process maps all grey levels between the two threshold bounds to black (grey level 0) and those outside to white (grey level 255). For the Data Translation frame store used $g_{min} = 0$ and $g_{max} = 255$. Optimum values for the other parameters were found empirically to be:-

$$g_e = 10$$

$$g_s = 10 \text{ (on initialisation)}$$

$$\Delta g = 10$$

A cross section along a row or column of the image, taken through the indentation would appear as in Fig. 4.23. The steepness of the "valley" sides indicated the sharpness of indentation edges in the captured image. A numerical estimate of the error in calculating the diagonal length of the indentation may be specified as Δt, where Δt is the number of pixels falling between x_1 and x_2 , x_1 may be specified as the position where the intensity falls below 0.9 t_1 and x_2 as the position where the intensity falls below 1.1 t_2.

t_1 may be set as the average grey level of the background and t_2 the average grey level of the indentation.

4.3.3 ROUGH CENTRE OF AREA CALCULATION

Having converted the grey scale image to a binary representation the next step is to compute the centre of area of the indentation. For high contrast images the binary representation will contain just one black region (corresponding to the indentation) on a white background. Under less favourable conditions the binary image may contain black regions separate from the indentation caused by surface blemishes on the test piece. These spuriously thresholded regions are generally small in comparison to the indentation and therefore by calculating the centre of area of the image as a whole a rough estimate of the centre of the indentation can be made. Where this assumption is not valid a more sophisticated scheme is required (section 4.4).

The coordinates of the centre $(\bar{x}, \bar{y})$ are given by:-

$$\bar{x} = \frac{1}{A} \sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} x f(x,y) \qquad [4.10]$$

$$\bar{y} = \frac{1}{A} \sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} y f(x,y) \qquad [4.11]$$

where the area, A is given by:-

$$A = \sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} f(x,y) \qquad [4.12]$$

where n is the number of columns and m the number of rows of pixels in the image array and f(x,y) is a binary valued function defined by:-

$$f(x,y) = 1, \quad g(x,y) = 0$$
$$f(x,y) = 0, \quad g(x,y) = 255 \qquad [4.13]$$

where g(x,y) represents the thresholded grey scale of a pixel at location (x,y).

4.3.4 PERIMETER CALCULATION

The next stage in the process is to locate those pixels which form the boundary of the indentation. Several algorithms are available for perimeter tracing, one of the most popular being analogous to a blind man feeling his way around a building [48]. However the following scheme was used in preference for reasons given later.

Starting at the centre of area the pixels are scanned radially one by one until five contiguous white pixels are found (Fig. 4.5), the previous black pixel is then taken as a boundary point. The process is repeated with the scan rotated at 1 degree increments until an entire circle has been swept out. The perimeter is then described by a set of 360 coordinates. The algorithm is quite immune to salt and pepper noise (section 4.4.1) as can be judged by the results shown in Chapter 6. For a truly square indentation, occupying the entire field of view, the distance between successive boundary points will be 4-5 pixels; this leads to an unceratinty in the diagonal length measurement of 12-14

pixels, corresponding to 18-30 μm (since the vertical and horizontal scaling factors are 673 and 454 pixels/mm respectively). For a material of nominal hardness 220 HV30 the corresponding variation in the hardness values is 207-233 HV30. Approximately 1200 boundary pixels (i.e. a scan increment of 0.25 degrees) would be needed to ensure that all the successive boundary pixels are 8-adjacent. This number was thought excessive in terms of execution and 360 points was regarded as giving the optimum trade-off between speed and accuracy.

In several of the images it was noticed that a large black region (usually caused by spurious thresholding of the part of the image where "sinking in" had occurred) was connected to the object by just a few pixels (Fig. 4.6). Using the "blind man" perimeter tracing algorithm would have lead to the boundary of this region being recorded as part of the perimeter, causing large errors in the signature calculation (section 4.3.5). The algorithm used here records only a few erroneous boundary pixels where the connected region meets the indentation.

4.3.5 ACCURATE CENTRE OF AREA CALCULATION

The centre of area as calculated in section 4.3.3 gives only a rough estimate of the indentation centroid since it is based on the whole of the image, not just the object itself. At this stage of the processing a more accurate calculation is made by applying the centre of area formulae [4.10 - 4.13] to the boundary coordinates. This is analogous to finding the centroid of a two dimensional wire frame.

A drawback with this approach is that if the first estimate of the centre of area is wildly inaccurate (in extreme cases it is possible that the centroid can actually lie outside the

indentation) then those pixels lying furthest from the centre will be more widely spaced than those lying nearest to the centre, since the scan is always taken at fixed 1 degree increments. The pixel density will therefore be less at the most distant parts of the boundary. This has the effect of biasing the calculation toward the original centre of area. A solution to this problem is discussed in section 4.4.

4.3.6 SIGNATURE CALCULATION

A signature is a one-dimensional functional representation of the boundary of an object [49]. One method of generating the signature is to plot boundary distance from the centroid against angle (Fig. 4.7) and it is this method which has been employed here. For a quadrilateral object, to which the indentation approximates, the signature contains four distinct peaks corresponding to the corners. The program, after "smoothing" the signature to nullify any spurious boundary points, locates these peaks and records their corresponding distances from the centroid. From these distances the lengths of the diagonals and their mean value can be found.

4.3.7 HARDNESS CALCULATION

The signature calculation can be used to give the length of the diagonals in pixels, however to calculate the hardness it is necessary to find the true lengths in physical units. The aspect ratio of the camera is first taken into account by treating each diagonal as a vector and scaling the horizontal component by the aspect ratio (approximately 4:3). The true length is then found by dividing the diagonal length in pixels by the scaling factor.

The scaling factor was found experimentally as follows. An object was first moved in the horizontal then the vertical direction using a micrometer drive. The displacement in pixels was found by placing a cursor at a known position on the object (actually an indentation corner) and was plotted against physical displacement read from the barrel of the micrometer drive. The scale factor was found to be 454 pixel/mm in the horizontal direction and 673 pixel/mm in the vertical direction when the microscope had a magnification of 10X. Obviously doubling the magnification doubles this ratio. The Vickers hardness formula [2.4] can be adapted to take account of this:-

$$HV = \left(\frac{1.854\,F}{D^2}\right)\left(\frac{M^2}{100}\right) \qquad [4.14]$$

where M is the magnification, F the applied force in kgf and D the diagonal length in mm.

4.4 IMPROVED SOFTWARE

4.4.1 INTRODUCTION

The software described previously will give accurate results when using good samples under controlled conditions. For an automated system to work reliably however, the software must be capable of adapting to variations in sample quality and illumination. It must also be able to give error warnings in case of fault conditions (e.g. indentation not being within the field of view of the camera).

The most serious problem raised by the existing software is that the rough centre of area may lie outside the indentation itself. This occurs when pixels are spuriously labelled as object regions by the thresholding process under conditions of poor contrast. A solution to this is to use region labelling - this involves assigning a label to each object pixel within the image such that all pairs of pixels which are connected (section 2.4) share the same label whereas all pairs of pixels which are not connected have different labels.

For each connected region various parameters are calculated and those region(s) of interest are extracted. Here the region of interest is the indentation which will, under normal circumstances, have the largest area. By extracting the largest region the indentation can be isolated and an accurate centre of area calculation made. The processing can then continue to the perimeter calculation and following stages.

Under low contrast conditions there is poor delineation between object pixels and background pixels. The thresholding process tends to yield many isolated white pixels in predominantly black regions and vice versa. This phenomenon, often called "salt and pepper" noise, causes the region labelling process to waste large numbers of labels on erroneously thresholded pixels. Clearly this is inefficient in terms of both memory requirements and execution time, therefore some pre-processing is usually necessary.

The amount of "salt and pepper" noise may be dramatically reduced by applying, iteratively, combinations of two operators often known as "blow"(or "expand") and "shrink" [50]. Expand (Fig. 4.8) sets all 8-neighbours of a pixel black, if that pixel is black, and tends to fill in predominantly black regions. Shrink (Fig. 4.9) on the other hand sets a pixel white if it is has less than a given number of black 8-neighbours, reducing the amount of isolated black pixels.

These operators can be realised very efficiently in hardware but in software are computationally expensive - the expand operator requiring nine read operations and one write operation for each pixel. A slightly more efficient set of operators have been suggested by [28].

The first operator fills in one pixel light areas in otherwise dark areas and is accomplished by using the Boolean expression (with ref to Fig. 4.10):-

$$B_1 = p + b \bullet g \bullet (d + e) + d \bullet e \bullet (b + g) \qquad [4.15]$$

where "$\bullet$" and "$+$" represent the logical AND and OR respectively. As in [4.13] a dark pixel is represented by a logical 1 and a light pixel by a logical 0. If B_1 then is true 1 is assigned to p, otherwise 0 is assigned to p. Equation [4.15] is assigned to all pixels simultaneously i.e. the next value of each pixel location is determined before any of the other pixels have been changed.

The other operator fills in small notches in straight edge segments and eliminates isolated 1's. This is accomplished using the Boolean expression:-

$$B_2 = p \bullet [((a + b + d) \bullet (e + g + h)] + [(b + c + e) \bullet (d + f + g))] \qquad [4.16]$$

These were programmed in 'C' and used with the Data Translation frame store but were still found to be too slow to be of use. To reduce the data processing requirements use can be made of run length coding [50]. Here, instead of storing the whole binary image only the locations of black to white and white to black transitions along consecutive rows are recorded. For the frame store used each image contains 512 x 512 = 256k pixels, however typical indentation images usually contain only about 10k transitions. Thus a reduction in data of around 25:1 has been achieved. In theory this should lead

to a corresponding reduction in execution time. In practise an additional overhead is incurred with the adaptation of the expand/shrink operators to work with run length coded data. The modified operators along with the modified region labelling algorithm are described in sections 4.4.3 to 4.4.5.

The algorithms underlying these are original to this work. Although run length coding has been applied to the detection and and characterisation of connected components [51] it has not been applied to binary smoothing. The algorithms used here reduced the execution time from approximately 5 min (for the pixel based method in equations [4.15]-[4.16]) so approximately 30 sec. The position of the indentation centroid found after pixel based smoothing did not differ by more than one pixel from that found using run length coded smoothing.

4.4.2 RUN LENGTH CODING

The run length coding algorithm records the locations of black to white and white to black transitions in consecutive pixels along each row of the image, row by row. The first and last columns of the image are set white so that each row starts with a white to black transition and ends with a black to white transition - this simplifies the expand and shrink operations. There are three types of codeword:-

RN - contains the number of the row on which the transitions lie

TW1 - contains the column number of a white to black transition

TW2 - contains the column number of a black to white transition

A typical part of the code might look like:-

... RN / TW1 / TW2 / TW1 / TW2 / RN / TW1 / TW2 / RN ...

The codewords are defined in such a way that the software can distinguish between different types, the actual representation is given in section 4.5.2.

4.4.3 EXPAND OPERATION

The expand operation, when combined with the shrink operation, is used to provide binary smoothing of images. This was necessary to reduce the number of black to white and white to black transitions in an image so that the region labelling process could be performed more efficiently. The expand operation expands the border of an object by one pixel. This can be achieved in three stages: an expansion in the x direction, an expansion in the -y direction and an expansion in the +y direction. A separate algorithm for the +y direction is not necessary; by reversing the code, applying the -y algorithm then reversing the code again the same result can be effected. In practise an expansion in the +y direction was not found to be necessary.

In the following discussion extensive use will be made of the notation developed by [51]. Here a run is defined as a compact part of a line (row) not belonging to the background (Fig. 4.11) and can be represented by a 3-tuple of integers (i,k,l) such that:-

$$(1 \leq i \leq n) \cap (1 \leq k \leq l \leq m) \cap (f(i,k-1) = 0) \cap (f(i,l+1) = 0) \cap (\forall j, k \leq j \leq l, f(i,j) = 1) \quad [4.17]$$

where [m,n] is the order of the image array and f(x,y) is as defined in [4.13]. Furthermore for a given run s = (i,k,l)

i is called the line (or row number) of the run, denoted by lin(s)

k is called the left of the run, denoted by lft(s)

l is called the right of the run, denoted by rgt(s)

from the description of the run length code clearly:-

$$i \equiv RN$$

$$k \equiv TW1$$

$$l \equiv TW2$$

Let S_i be the set of runs on the ith line ($1 \leq i \leq m$) and s_j be the jth run on the ith line, i.e.:-

$$\forall j,\ s_j \in S_i, s_{j+1} \in S_i, lft(s_j) < lft(s_{j+1}) \qquad [4.18]$$

Let r_j be the jth run on the (i+1)th line ($r_j \in S_{i+1}$) and $\{r\}$ is similarly ordered. Where the ordering is unimportant the subscripts will be dropped.

The x direction expansion is achieved by moving the left of each run one pixel to the left and the right of the run one pixel to the right. If as a result any two runs overlap then they are condensed into one run by removing the right of the first run and the left of the second run. Strictly:-

for each run:-

$$lft(s) \leftarrow lft(s) - 1 \qquad [4.19]$$

$$rgt(s) \leftarrow rgt(s) + 1 \qquad [4.20]$$

if as a result for any pair of runs:-

$$rgt(s_j) \geq lft(s_{j+1}) \qquad [4.21]$$

then remove $rgt(s_j)$, $lft(s_{j+1})$ and

$$rgt(s_j) \leftarrow rgt(s_{j+1}) \qquad [4.22]$$

To perform the y expansion it is necessary to examine the runs on the (i + 1)th row to see the effect of a one pixel expansion into the ith row. There are five basic configurations between runs of two consecutive rows (Fig. 4.12). These are defined thus:-

Configuration 1

$$\exists r, \exists s, \ (lft(r) \leq lft(s) \leq rgt(r)) \cup (lft(r) \leq rgt(s) \leq rgt(r)) \qquad [4.23]$$

call this relation $CON1(s,r)$

Configuration 2

$$(\exists r, \exists s, \ (lft(r) \leq lft(s) \leq rgt(r)) \cap (lft(r) \leq rgt(s) \leq rgt(r))) \cap (\forall v, v \neq s{:}CON1(v,r)) \qquad [4.24]$$

call this relation $CON2(s,r)$

Configuration 3

$$(\exists r, \exists s \ (lft(r) < lft(s) \leq rgt(r)) \cap (rgt(r) < rgt(s))) \cap (\forall u, u \neq r{:}CON1(s,u)) \cap (\forall v, v \neq s{:}CON1(v,r)) \qquad [4.25]$$

call this relation $CON3(s,r)$

Configuration 4

$$(\exists r, \exists s,\ (lft(r) \leq rgt(s) < rgt(r)) \cap (lft(s) \leq lft(r))) \cap (\forall u, u \neq r{:}CON1(s,u)) \cap (\forall v, v \neq s{:}CON1(v,r)) \quad [4.26]$$

call this relation $CON4(s,r)$

Configuration 5

$$\exists r, \exists s,\ (lft(r) \geq lft(s)) \cap (rgt(s) \geq rgt(r)) \cap (\forall u, u \neq r{:}CON1(s,u)) \quad [4.27]$$

call this relation $CON5(s,r)$

The rules for these configurations are defined so that a run can only overlap either zero or one and one only other runs. Expansions corresponding to each of these basic configurations are detailed below (Fig. 4.13):-

Configuration 1

create a new run with:-

$$lft(s) \leftarrow lft(r) \quad [4.28]$$

$$rgt(s) \leftarrow rgt(r) \quad [4.29]$$

Configuration 2

$$lft(s) \leftarrow lft(r) \qquad [4.30]$$

$$rgt(s) \leftarrow rgt(r) \qquad [4.31]$$

Configuration 3

$$lft(s) \leftarrow lft(r) \qquad [4.32]$$

Configuration 4

$$rgt(s) \leftarrow rgt(r) \qquad [4.33]$$

Configuration 5

no expansion possible

It is also possible to build up more complicated configurations using the five basic configurations given above. A block (Fig. 4.14) can be defined as a set of runs $T \cup U$ where $T \subset S_i$ is a subset of the set of runs on the ith line and $U \subset S_{i+1}$ is a subset of the set of runs on the (i + 1)th line.

Let $t_j \in T,\ t_j = s_j,\ n \leq j \leq m$ [4.34]

Let $u_j \in U,\ u_j = r_j,\ k \leq j \leq l$ [4.35]

Then $T \cup U$ is a block if:-

$$\nexists t, \nexists u:\ CON1(t,u) \cup CON2(t,u) \cup CON3(t,u) \cup CON4(t,u) \cup CON5(t,u) \qquad [4.36]$$

This simply means that a block is a set of runs to which none of the previous configurations apply. The expansion is as follows (Fig. 4.15):-

replace all runs of the block on the ith line (i.e. T) by a single run s

if

$$lft(u_k) < lft(t_n) \qquad [4.37]$$

then

$$lft(s) \leftarrow lft(u_k) \qquad [4.38]$$

else

$$lft(s) \leftarrow lft(t_n) \qquad [4.39]$$

if

$$rgt(u_l) > rgt(t_m) \qquad [4.40]$$

then

$$rgt(s) \leftarrow rgt(u_l) \qquad [4.41]$$

else

$$rgt(s) \leftarrow rgt(t_m) \qquad [4.42]$$

and in all cases

$$lin(s) \leftarrow lin(t) \qquad [4.43]$$

4.4.4 SHRINK OPERATION

As with the expand operation the shrink operation is first performed in the x direction then in the +y direction. The x direction shrink is achieved by moving the left of each run one pixel to the right and the right of each run one pixel to the left. All runs of one or two pixels in length are removed. Strictly:-

for all runs

if

$$rgt(s) - lft(s) \leq 2 \qquad [4.44]$$

then remove s

otherwise:

$$lft(s) \leftarrow lft(s) + 1 \qquad [4.45]$$

$$rgt(s) \leftarrow rgt(s) - 1 \qquad [4.46]$$

To perform the y shrink it is necessary to examine runs on the ith row to see the effect of a one pixel shrink on the (i+1)th row (Fig. 16). The shrink operations, as applied to the five basic configurations given before, are:-

Configuration 1

remove the run on the (i+1)th line; r

Configuration 2

$$lft(r) \leftarrow lft(s) \qquad [4.47]$$

$$rgt(r) \leftarrow rgt(s) \qquad [4.48]$$

Configuration 3

$$lft(r) \leftarrow lft(s) \qquad [4.49]$$

Configuration 4

$$rgt(r) \leftarrow rgt(s) \qquad [4.50]$$

Configuration 5

no shrink possible

In applying the shrink operation to blocks it is necessary to "explode" each run of the block on the (i+1)th line according to those runs on the previous line (Fig. 4.17). In other words each run,$u \in U$, on the (i+1)th line (see [4.35]) is replaced by a set of runs

W . It is convenient to define a subset of runs, $V \subset T$, on the ith line, each of which overlaps u i.e.

$$\forall v,\ v \in V, (lft(u) < lft(v) < rgt(u)) \cup (lft(u) < rgt(v) < rgt(u)) \qquad [4.51]$$

If V is an empty set then no shrink operation is possible and u is retained, otherwise the members of W are given as follows:-

if

$$\exists v, v \in V,\ (lft(u) < rgt(v) < rgt(u)) \cup (lft(v) < lft(u)) \qquad [4.52]$$

define LEFT such that if v satisfies the above condition $v = \text{LEFT}(V,u)$

then $w \in W$ with:-

$$lft(w) \leftarrow lft(u) \qquad [4.53]$$

$$rgt(w) \leftarrow rgt(v) \qquad [4.54]$$

$$lin(w) \leftarrow rgt(u) \qquad [4.55]$$

if

$$\exists v,\ v \in V,\ (lft(u) < lft(v) < rgt(u)) \cap (rgt(u) < rgt(v)) \qquad [4.56]$$

define RIGHT such that if v satisfies the above condition $v = \text{RIGHT}(V,u)$

then $w \in W$ with

$$lft(w) \leftarrow lft(v) \qquad [4.57]$$

$$rgt(w) \leftarrow rgt(u) \qquad [4.58]$$

$$lin(w) \leftarrow rgt(u) \qquad [4.59]$$

the other members of W are given by:-

$$lft(w) \leftarrow lft(v') \qquad [4.60]$$

$$rgt(w) \leftarrow rgt(v') \qquad [4.61]$$

$$lin(w) \leftarrow lin(v') + 1 \qquad [4.62]$$

$$\text{where } \{v'\} = V - \text{LEFT}(V,u) - \text{RIGHT}(V,u) \qquad [4.63]$$

4.4.5 REGION LABELLING

The region labelling algorithm assigns groups of connected object pixels with a unique identifier [51,53]. The process starts by assigning each group of object pixels (i.e. each run) on the first line a different label. Subsequent lines are then scanned from left to right and labels applied to those pixels which are 8-adjacent on the previous line:-

line i	|P|P|P|
line i + 1	| |X| |

If X is an object pixel and any of the points P are labelled then X is given the same label. If none of the points P are labelled and X is an object pixel then X is assigned a new label. A set of pixels sharing the same label is called a blob. If two branches, having different labels, from the previous line converge (i.e. it turns out that they belong to the same 8-connected region) then the label of the leftmost branch is taken to be the one which survives (Fig. 4.18). A note of each convergence is made in a separate table. In the case of a divergence, i.e. where a blob fans out into branches on successive lines, the old label is retained for each branch.

The above discussion applies to a pixel based algorithm however, the algorithm can easily be adapted to work with run length coded data:-

In case of configuration 1 the run r is assigned a new label.

For simple configurations (i.e. numbers 2-5):-

$$\text{LABEL}(r) \leftarrow \text{LABEL}(s) \qquad [4.64]$$

In the case of blocks (using same notation as for [4.51]), if V is an empty set then:-

$$\text{LABEL}(u) \leftarrow \text{LABEL}(t) \qquad [4.65]$$

such that

$$\exists u, \exists t, \quad (lft(u) > lft(t)) \cap (rgt(t) > rgt(u)) \qquad [4.66]$$

otherwise

$$\text{LABEL}(u) \leftarrow \text{LABEL}(v) \qquad [4.67]$$

such that if v is not unique (i.e. V contains more than one member)

$$\forall z, \; z \in V, \; z \neq v, \; lft(v) < lft(z) \qquad [4.68]$$

Following the region labelling step it is necessary to analyse the convergence list (CONVLIST) to group together connected blobs - this is known as label collection (Fig. 4.19). The algorithm [50] works with a set of stacks into which equivalent label from CONVLIST is pushed. The stack address (SA) field is an array of pointers which contain the stack number of each label. Every time a pair of labels (i,j) is taken from CONVLIST a test is made to see whether one of the corresponding SA cells already contains a pointer to a stack. The following cases may occur:-

(1) None of the labels is in a stack (SA(i) = SA(j) = NONE), then both labels are pushed onto the next empty stack X and SA(i) = SA(j) = 'pointer to stack X' are set.

(2) One of the labels, say i, is already in a stack; then j is pushed onto the same stack and SA(j) - SA(i) is set.

(3) The two labels are in different stacks (SA(i) ≠ SA(j)); then one of the labels is pushed onto the other to collect all the equivalent labels. At the same time, all SA cells that pointed to the empty stack must be updated to the common stack and the empty stack is free again.

After one run through CONVLIST each stack contains a set of equivalent labels.

The area of each blob is stored in a separate array indexed by label. Once the label collection routine has been performed then the areas of connected blobs are summed to give the corresponding region area. In the Vickers hardness test application the largest region is taken to be the indentation and is identified by a set of labels produced by the region labelling algorithm.

Note that for run length coded data the centroid equations can be simplified. Using equation [4.10] the equation for the x co-ordinate of the centroid can be rewritten:-

$$\bar{x} = \frac{1}{A}\sum_{x=0}^{x=n-1}\sum_{y=0}^{y=m-1} x f(x,y)$$
$$= \frac{1}{A}\sum_{S_i}\sum_{x=lft(s)}^{x=rgt(s)} x \qquad [4.69]$$

The inner summation represents the sum of an arithmetic series with $rgt(s) - lft(s) + 1$ terms. The first term is $lft(s)$ and the last term is $rgt(s)$. The sum of such a series is given by:-

$$\frac{1}{2}\,(\text{ number of terms })\,(\text{ first term } + \text{ last term }) \qquad [4.70]$$

hence

$$\sum_{x=lft(s)}^{x=rgt(s)} x \;=\; \frac{1}{2}\,(lft(s) + rgt(s))\,(rgt(s_i) - lft(s_i) + 1) \qquad [4.71]$$

therefore

$$\bar{x} = \frac{1}{2A}\sum_{S_i}\sum_{s}(lft(s) + rgt(s))\,(rgt(s) - lft(s) + 1) \qquad [4.72]$$

The equation for the y co-ordinate of the centroid can be rewritten as:-

$$\bar{y} = \frac{1}{A}\sum_{S_i}\sum_{s} i\,(rgt(s) - lft(s) + 1) \qquad [4.73]$$

where the area, A is given by:-

$$A = \sum_{S_i}\sum_{s}(rgt(s) - lft(s) + 1) \qquad [4.74]$$

4.4.6 PROGRAM ERROR TRAPPING

As indicated earlier, for the vision system to work reliably the software must be able to cope with a range of fault conditions. A list of these conditions are given in table 4.1.

Symptom	Cause	Action Taken
grey level histogram bi-modal but poorly defined	poor illumination or sample quality	use binary smoothing followed by region labelling to isolate the indentation
histogram not bi-modal	poor illumination, sample quality, or lack of indentation	FAIL - inform user
too many black/white transitions	analogue pick-up, poor sample quality, dust on lenses	FAIL - inform user
centroid outside indentation or too close to border	poor sample quality - crystal defects	use binary smoothing and region labelling to isolate the indentation
indentation not fully in view	mechanical fault	FAIL - inform user

Table 4.1 Summary of Program Error Traps.

The software first assesses the quality of the captured image by analysing the grey level histogram. If a threshold cannot be found by the algorithm given in section 4.3.2 then execution cannot continue further and an error is signalled. This will occur if the histogram is uni-modal, indicating poor illumination, poor sample quality or even lack of an indentation in the field of view.

Even if a threshold is found the image quality can still be poor. The degree of contrast in the image can be judged by calculating the ratios $n(g_1)/n(g_3)$ and $n(g_2)/n(g_3)$ (section 4.3.2). The higher these values are, the sharper the maxima and minimum in the

histogram and hence the higher the contrast. If either of these values is found to be less than 2, then the contrast is assumed to be too poor for the indentation to be reliably segmented by thresholding alone. In this case the image is thresholded and the smoothing and region labelling operations then applied. Otherwise execution continues as normal.

The next stage of the processing is to calculate the centre of area based on the entire image. In cases where the indentation area is small and/or large areas of background pixels are spuriously thresholded as object regions, the centroid may lie so close to the border of the indentation that the signature calculation is invalidated. In extreme cases the centroid may even lie outside the indentation.

The software checks whether the centroid is inside the indentation and is sufficiently far away from the border before the signature calculation is performed. The criterion used is that the maximum distance from the centroid to the border should be no greater than twice the minimum distance. If an error is detected, the smoothing and region labelling routines are used to isolate the indentation region, from which an exact centre of area calculation can be made.

Errors may also arise in the region labelling and run length coding routines. The maximum number of transitions which can be stored by a single code is 32k. If this figure is exceeded an error is signalled and execution stops. Conventional (pixel based) smoothing could be used as a fall-back position however, this was considered to be too time consuming. It should be noted that such a large number of transitions would tend to suggest that the image quality was so poor that any smoothing operations would have little effect in any case. The maximum number of labelled blobs which can be accommodated is 256. If this figure is exceeded an error is flagged and execution stops.

Following the border tracing routine, a check is made to see if the indentation is entirely within the field of view. This may not be case in an automated system because of mechanical failure. If any part of the indentation perimeter touches the image border then the indentation is not fully in view an error. Subsequent processing is then curtailed.

4.5 DESCRIPTION OF THE PROGRAM CODE

As with the Shore hardness test software (Chapter 3) the Vickers software is divided into two parts, a set of routines written in 8086 assembler code which provide a low level interface to the frame store, and a C program which implements the image analysis functions. The 8086 code is the same as for the Shore test while the C program (Fig. 4.20-4.22) is outlines below (for a detailed description see Appendix VII).

The main() function of the C program consists of a large switch construct which transfers control according to options entered by the user. The 'm' and 'v' cases prompt the user for the new magnification and weight respectively. The 'v' case calls the verify() function so that the size of the indentation can be verified manually by the user. Details of this function are given later.

The 'p' case forms the heart of the program in which all the image processing takes place. In the 'p' case the binary() function is first called. This thresholds the image according to the algorithm given in section 4.3.2. This function returns one of three integer values: FATAL if the threshold bounds cannot be found, NON_FATAL if the image has been thresholded but the original contrast was poor and FALSE if the original

contrast was good and the image has been thresholded. If an error is not found then the centre_of_area() function is called. If a FATAL error is signalled control transfers back to the original menu selection. If a NON_FATAL error is signalled then the smooth() and blob() functions are called which perform the binary smoothing and region labelling operations respectively. If an error is signalled by either of these functions (indicating in the first instance an excessive number of runs and in the second case an excessive number of labels) control transfers back to the original menu selection. The centre_of_area() function returns the co-ordinates of the centroid of the indentation based on the image as a whole. The perimeter() function then traces out the boundary of the indentation. If any part of the indentation boundary touches the image border, an error is signalled and control subsequently transfers to the main menu selection.

The sig_check() function ensures that the centre of area is within the indentation boundary, if this is not so then an error is signalled and the smooth() and blob() functions are called. The error trapping for these routines is as above. If no error is signalled by the sig_check() function then the new_centre() function is called in order to calculate the centre of area based on the boundary points of the indentation. The sig() function then calculates the signature as an array of distance values versus angle and returns the angles of the corners from the horizontal. Finally the vickers() function computes the Vickers hardness for the given corner angles (and hence diagonal lengths), magnification and weight.

4.6 USING THE SYSTEM

4.6.1 SETTING UP THE HARDWARE

The first task in setting up the system is to attach the camera to the microscope. This can be achieved by removing the microscope eye piece and replacing the camera lens by an extension tube giving a sliding fit onto the eye piece mounting. The cables linking the frame store to the camera and monitors can then be attached.

The bulb providing the illumination is placed in the microscope and connected to a variable voltage power supply capable of providing up to 12V at up to 5A. The test piece is then placed on the microscope table and the bulb energised at 12V. By adjusting the height of the table and the angle of the microscope mirror a clear focused image can be formed on the monitor. Once this has been achieved the power supply voltage should be reduced so that the light level is just below that which causes saturation of the camera.

4.6.2 RUNNING THE SOFTWARE

The program is entirely menu driven and is invoked by the operator entering

vickers <enter>

The operator is then faced with the main options menu. There are five choices, each selected by pressing the letter next to the desired option on the keyboard. These options are:

m - alter Magnification power

By selecting this option the operator is able to enter the new magnification power of the microscope if it has been changed (default x10).

l - alter the applied Load

Allows the operator to tell the program the size of the load which was used to produce the indentation (default 10 Kgf).

p - Process a sample

Calculates the hardness of a material from an indentation using the above settings. Having selected this option the operator is prompted to position the indentation in the field of view of the camera then press a key. The analysis and calculation are carried out automatically. The results given include not only the hardness but also the lengths of the indentation diagonals and their average.

v - Verify the hardness manually

This allows the operator to check the accuracy of the system by measuring the size of the indentation directly. The user is instructed to position a set of software generated cursors at the corners of the indentation as seen on the output monitor. The length of the diagonals, their mean value and the corresponding hardness of the sample are then calculated by the program. An option is provided to "stretch" the grey scale of the image to improve visibility.

x - eXit

Terminate the program and return to the operating system (DOS).

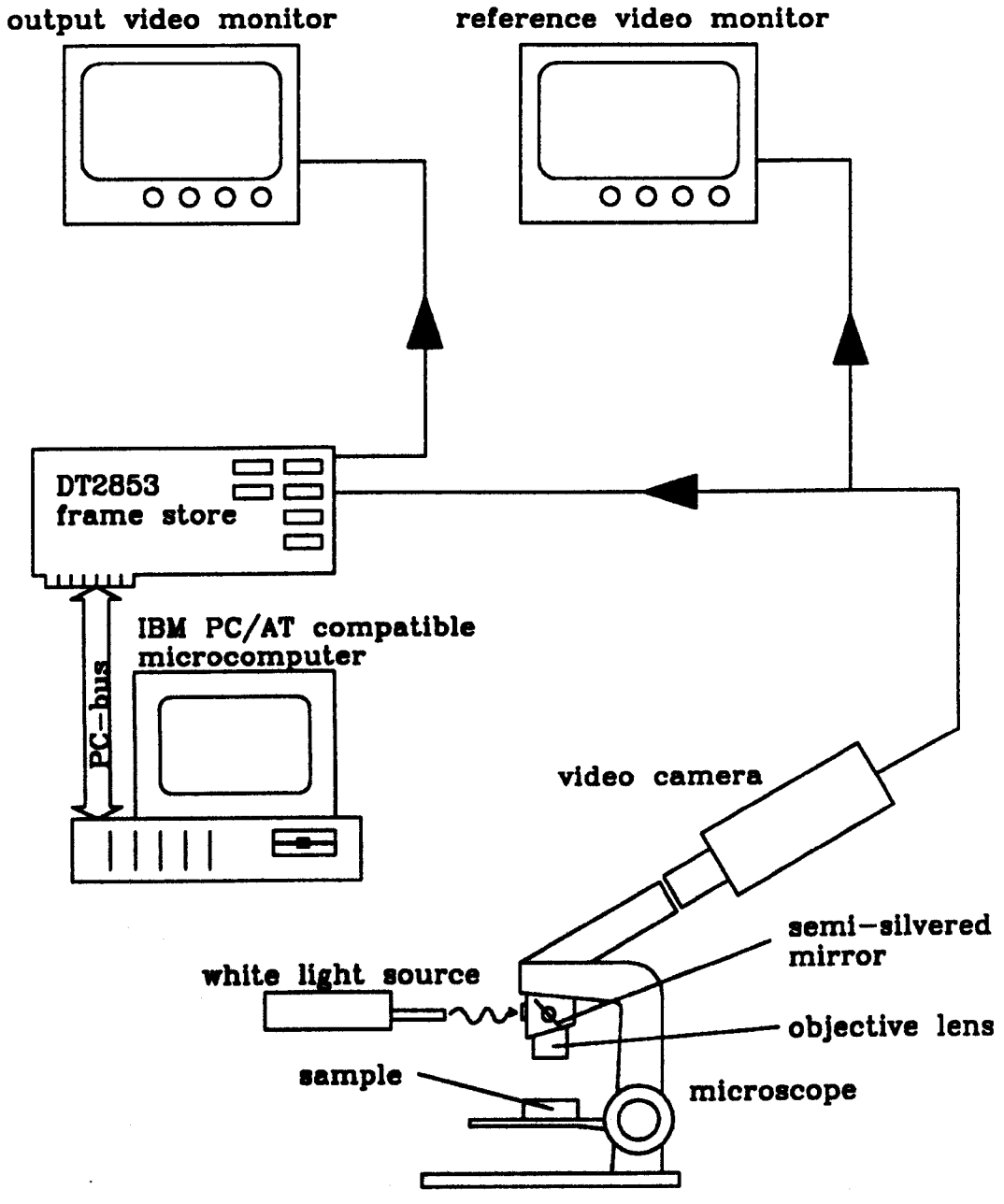


Figure 4.1: VISION SYSTEM HARDWARE

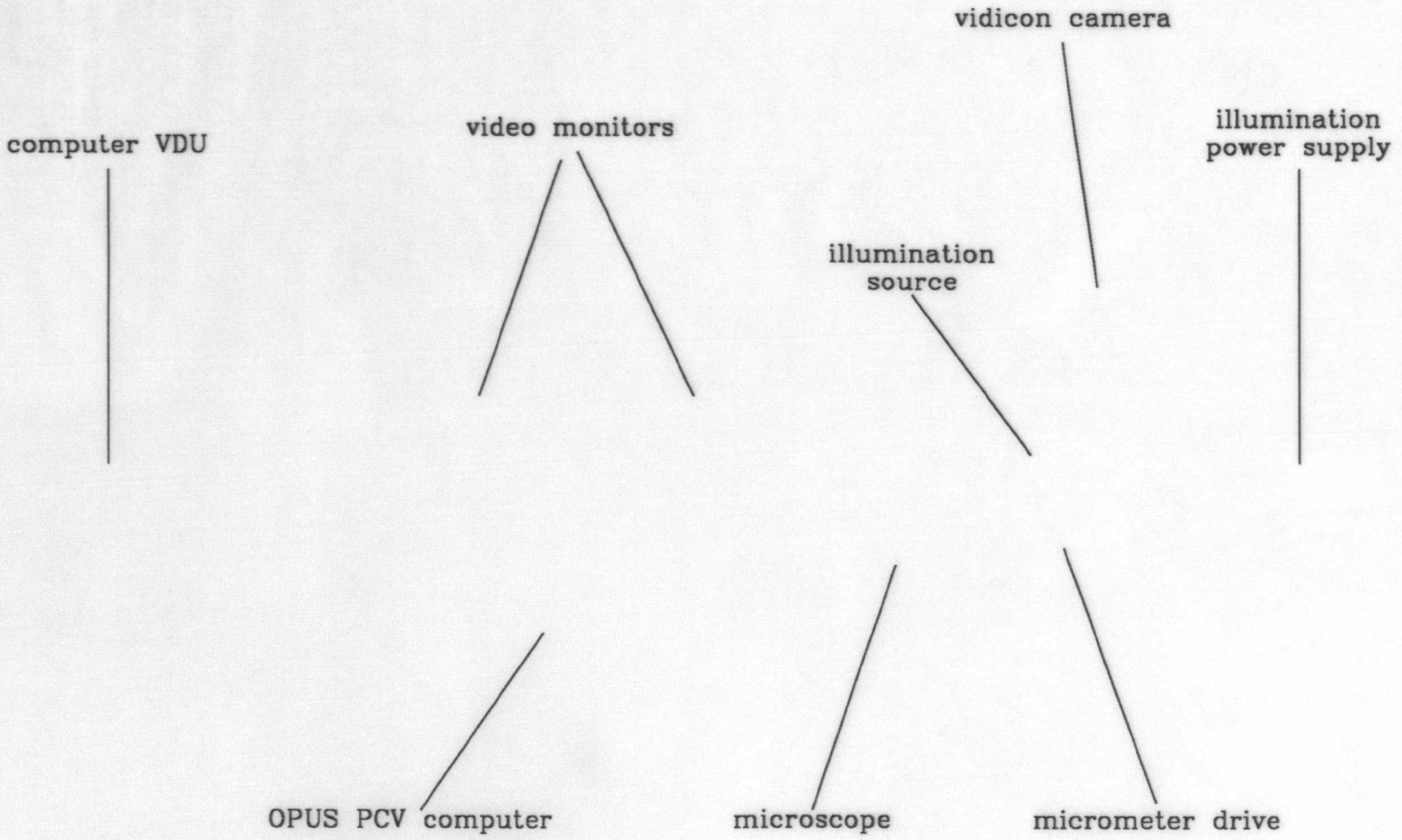


PLATE II: VICKERS SYSTEM HARDWARE

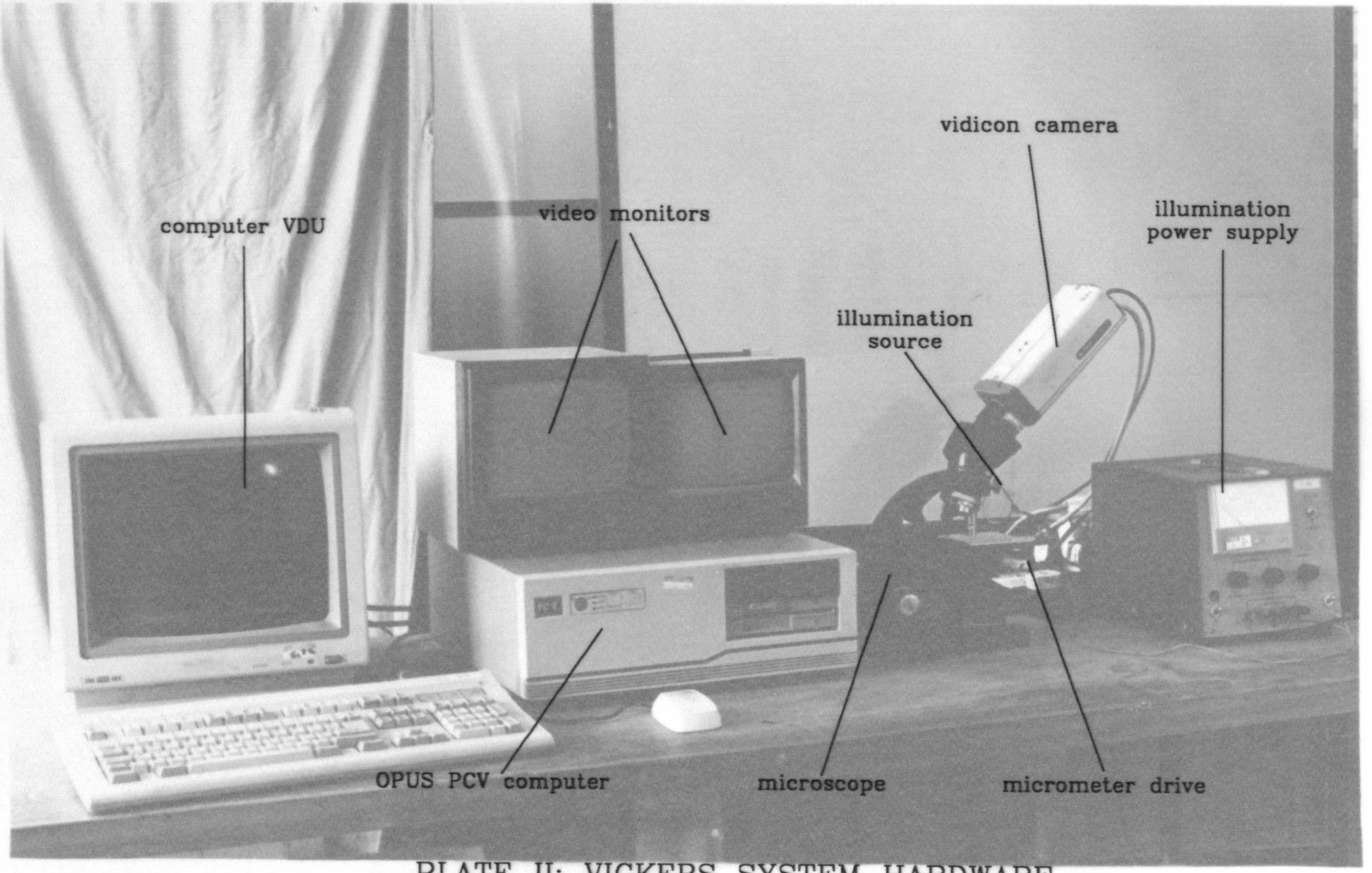
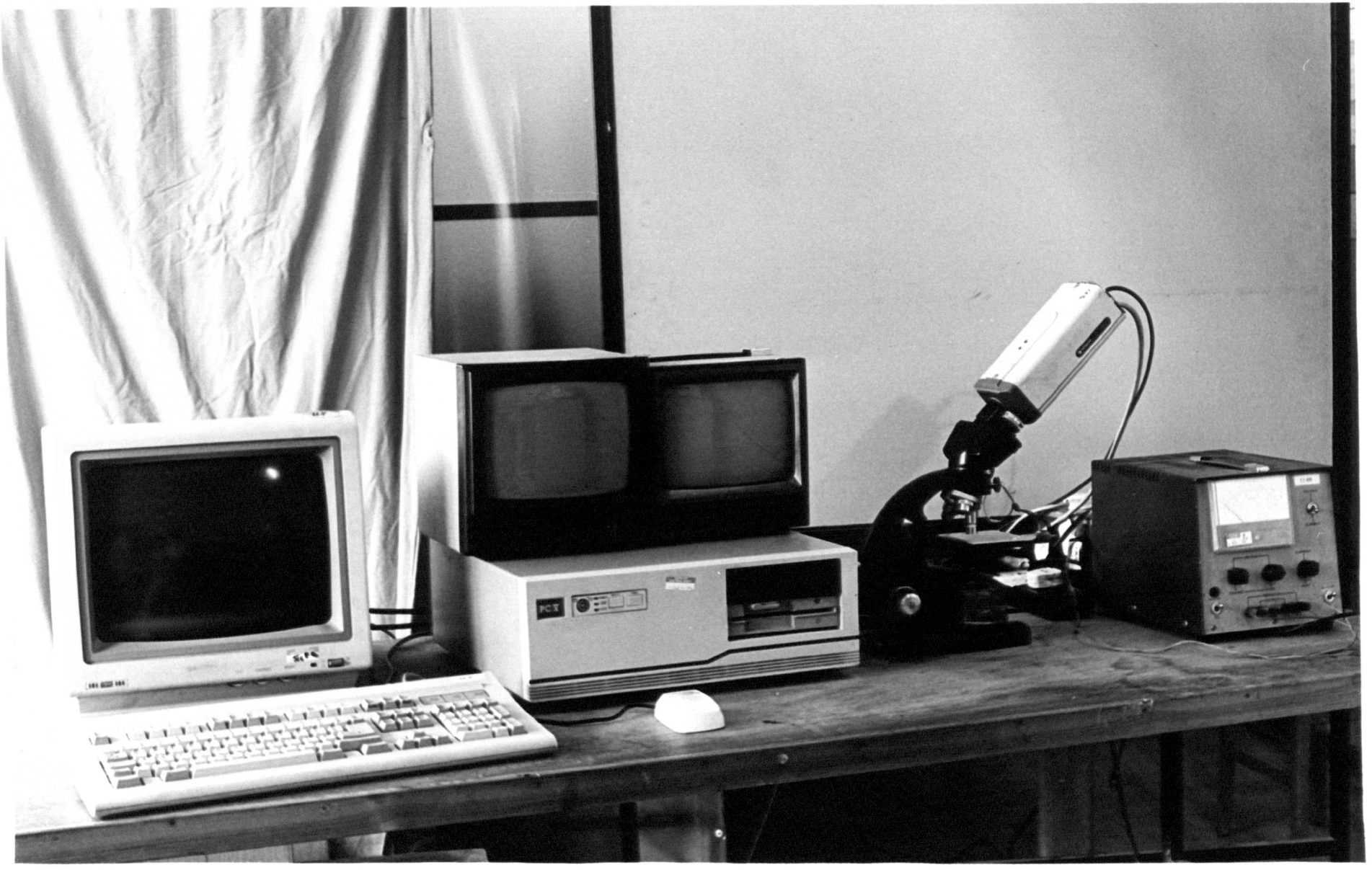


PLATE II: VICKERS SYSTEM HARDWARE


PC-T

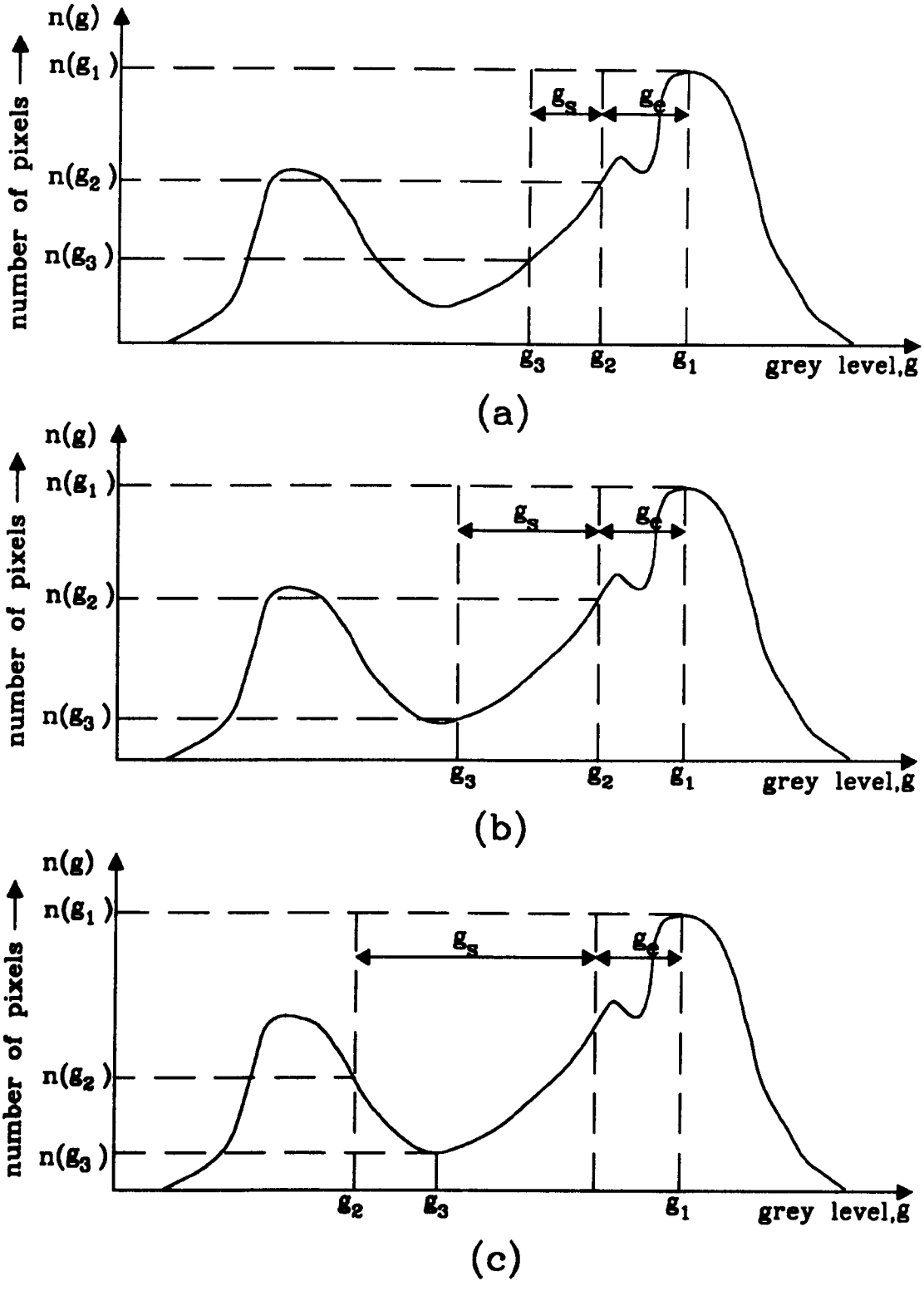


Figure 4.2: AUTOMATIC THRESHOLDING ALGORITHM

location of a minimum on the lower side of the first peak, algorithm (a) after one iteration, (b) after two iterations, (c) on termination.

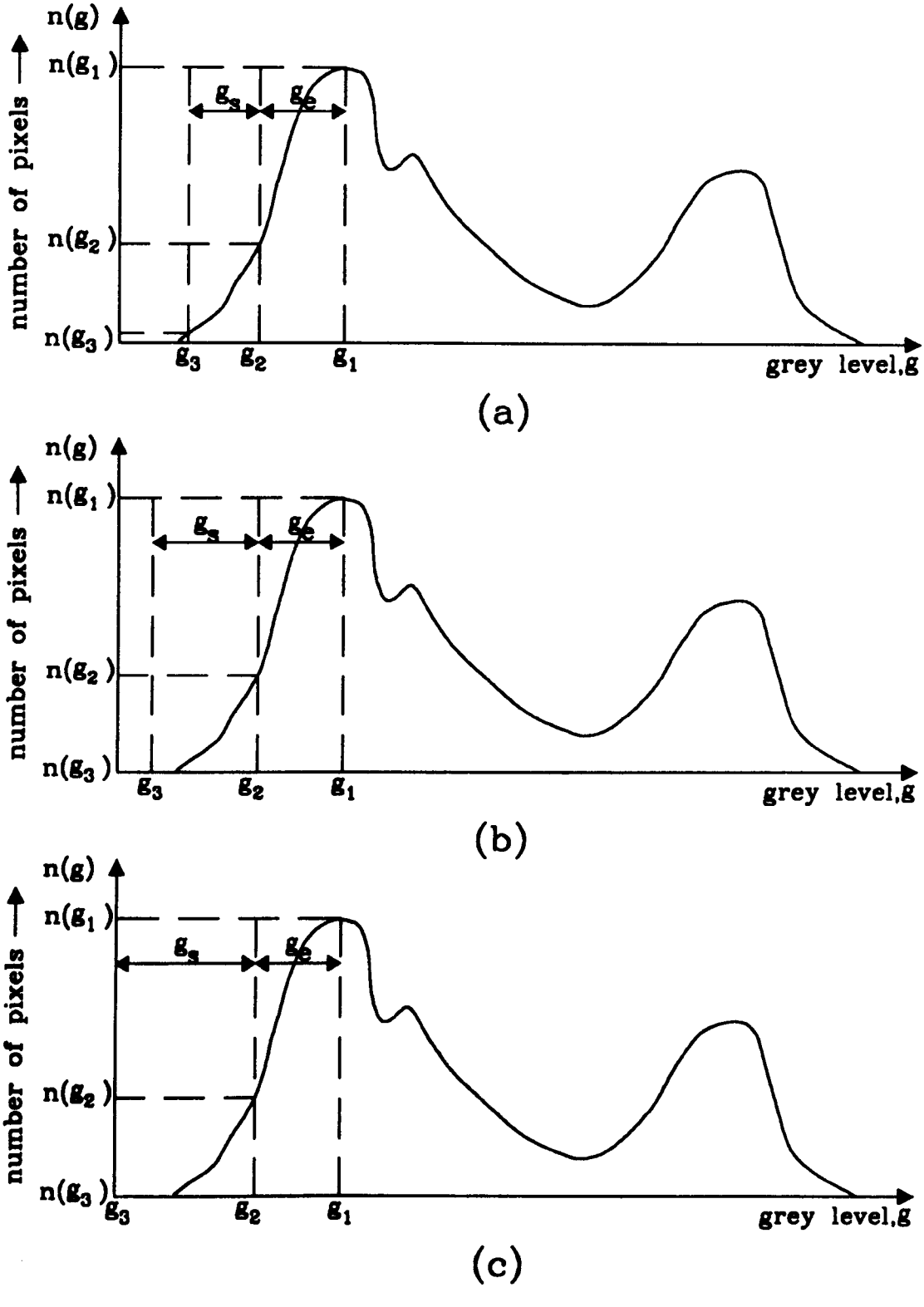


Figure 4.3: AUTOMATIC THRESHOLDING ALGORITHM

case in which minimum is absent on lower side of first peak, algorithm (a) after one iteration, (b) after several iterations, (c) on termination.

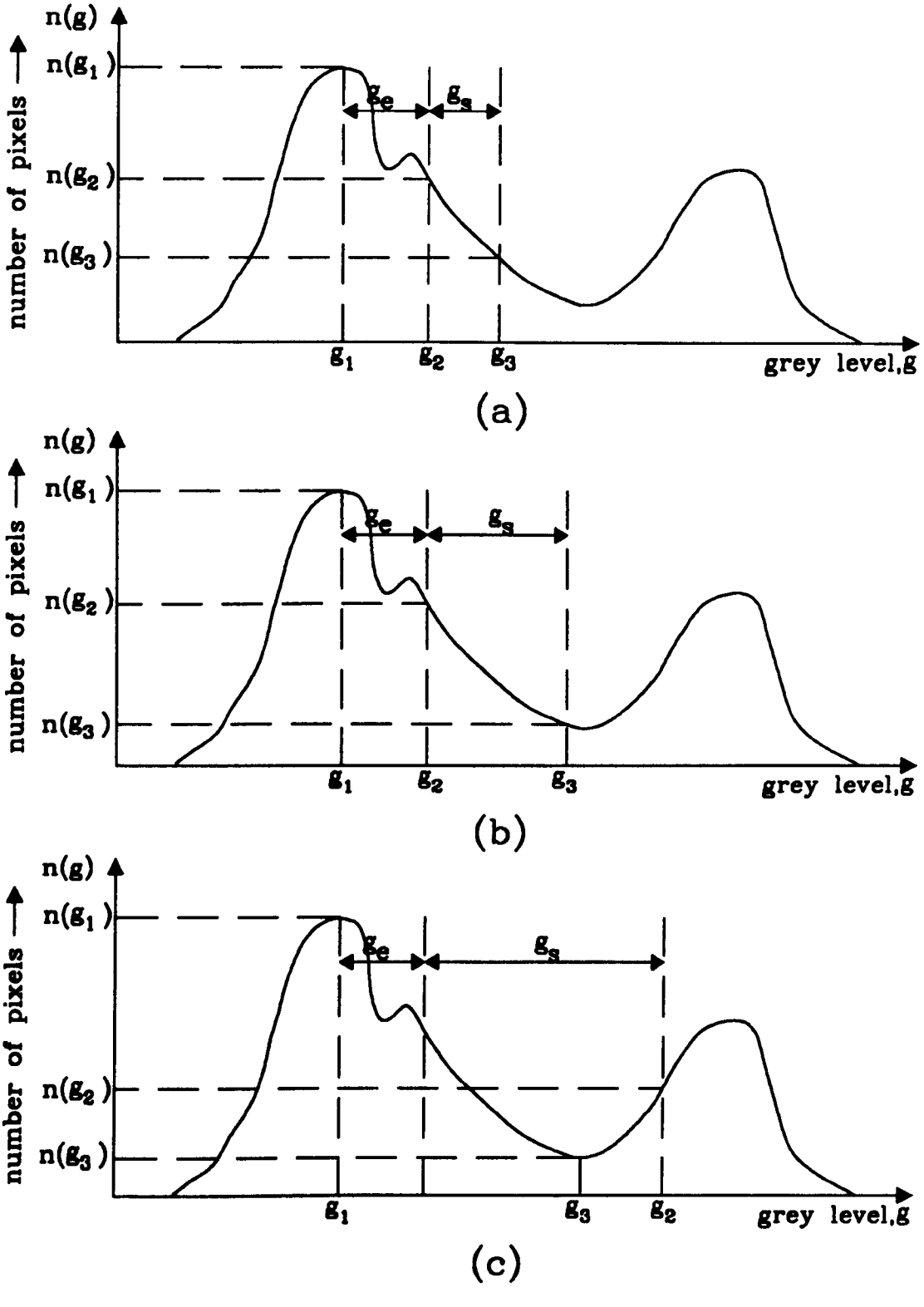


Figure 4.4: AUTOMATIC THRESHOLDING ALGORITHM

location of a minimum on the upper side of the first peak, algorithm (a) after one iteration, (b) after several iterations, (c) on termination.

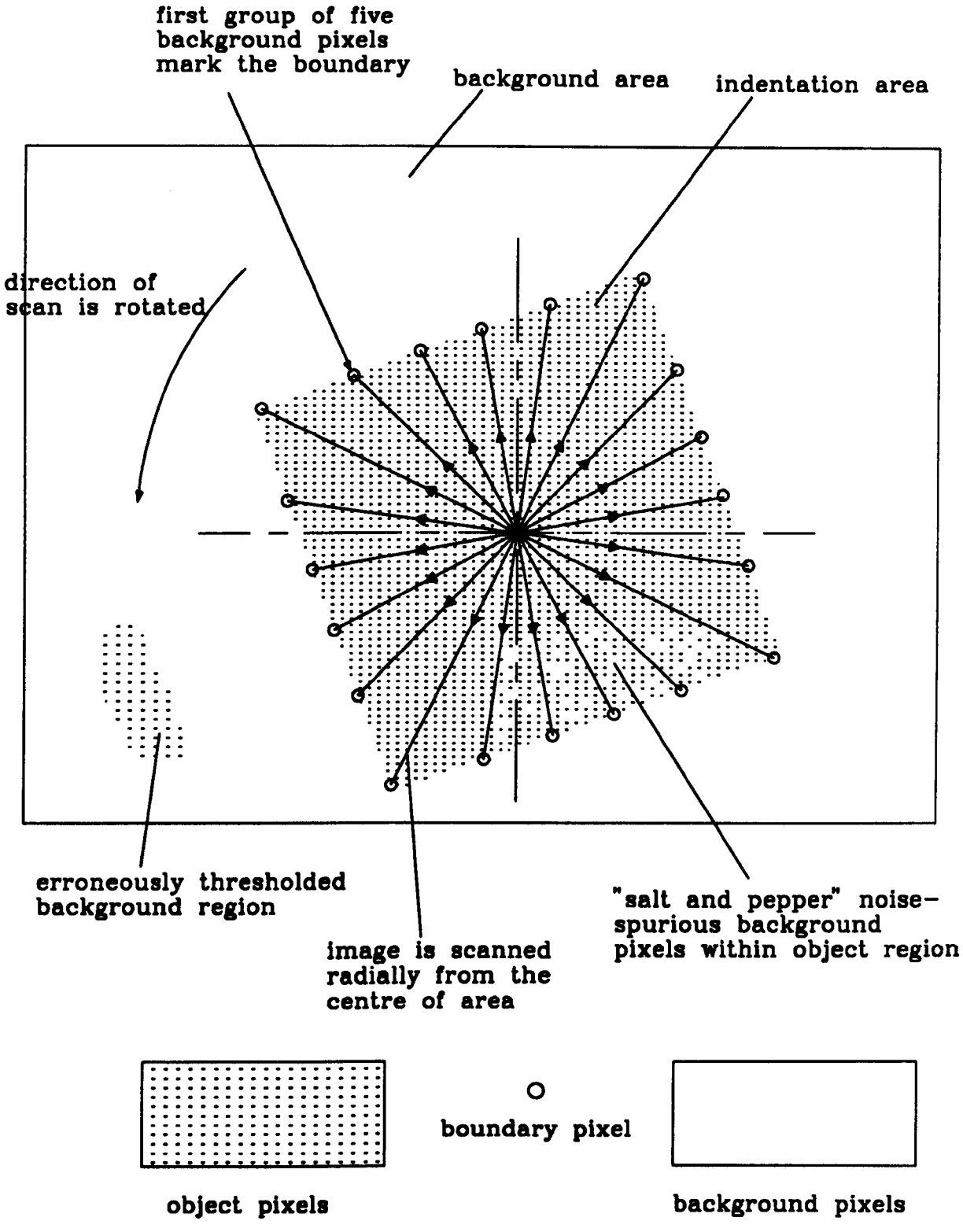


Figure 4.5: BOUNDARY FINDING ALGORITHM

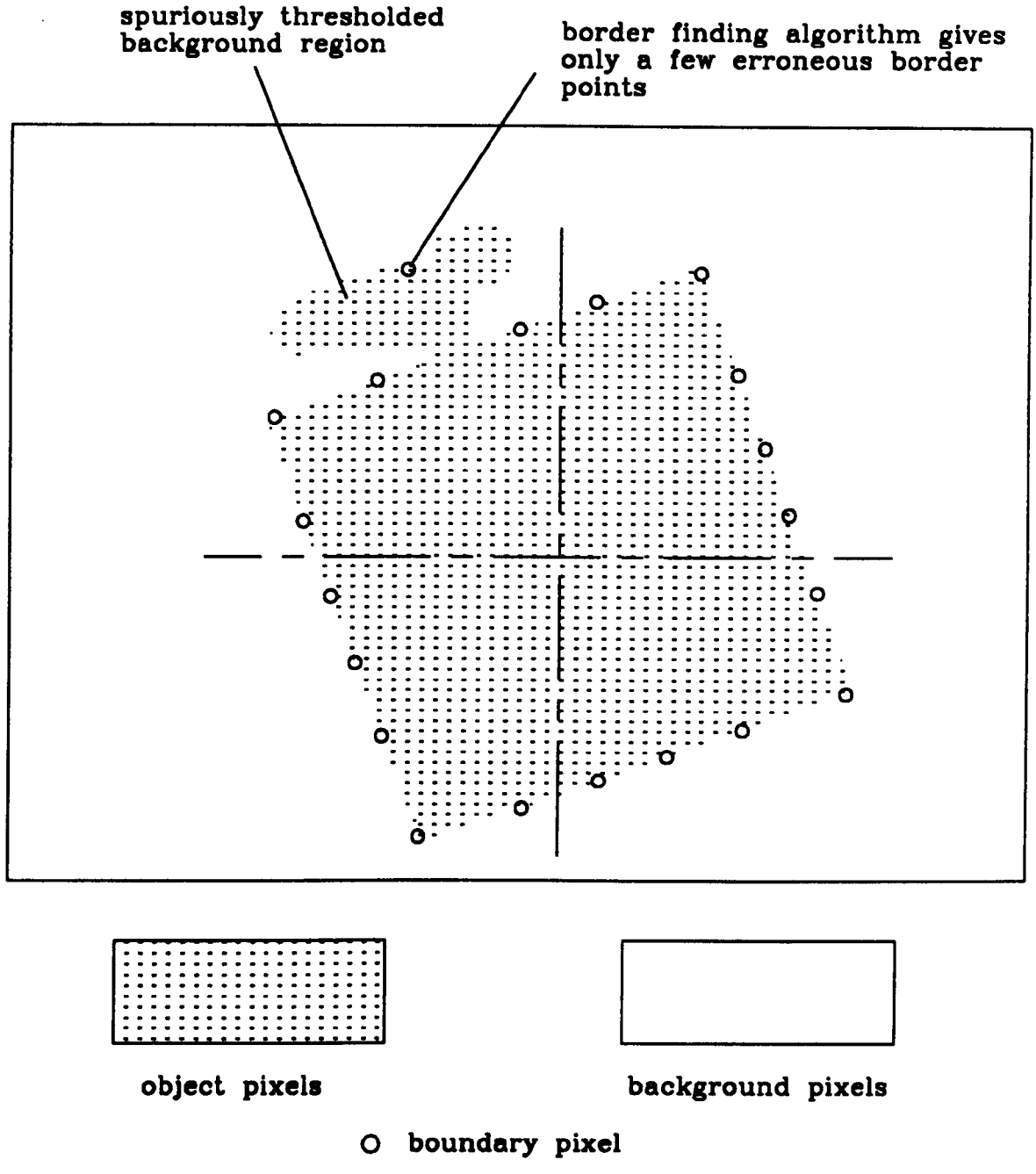


Figure 4.6 ADVANTAGE OF THE BORDER TRACING ALGORITHM

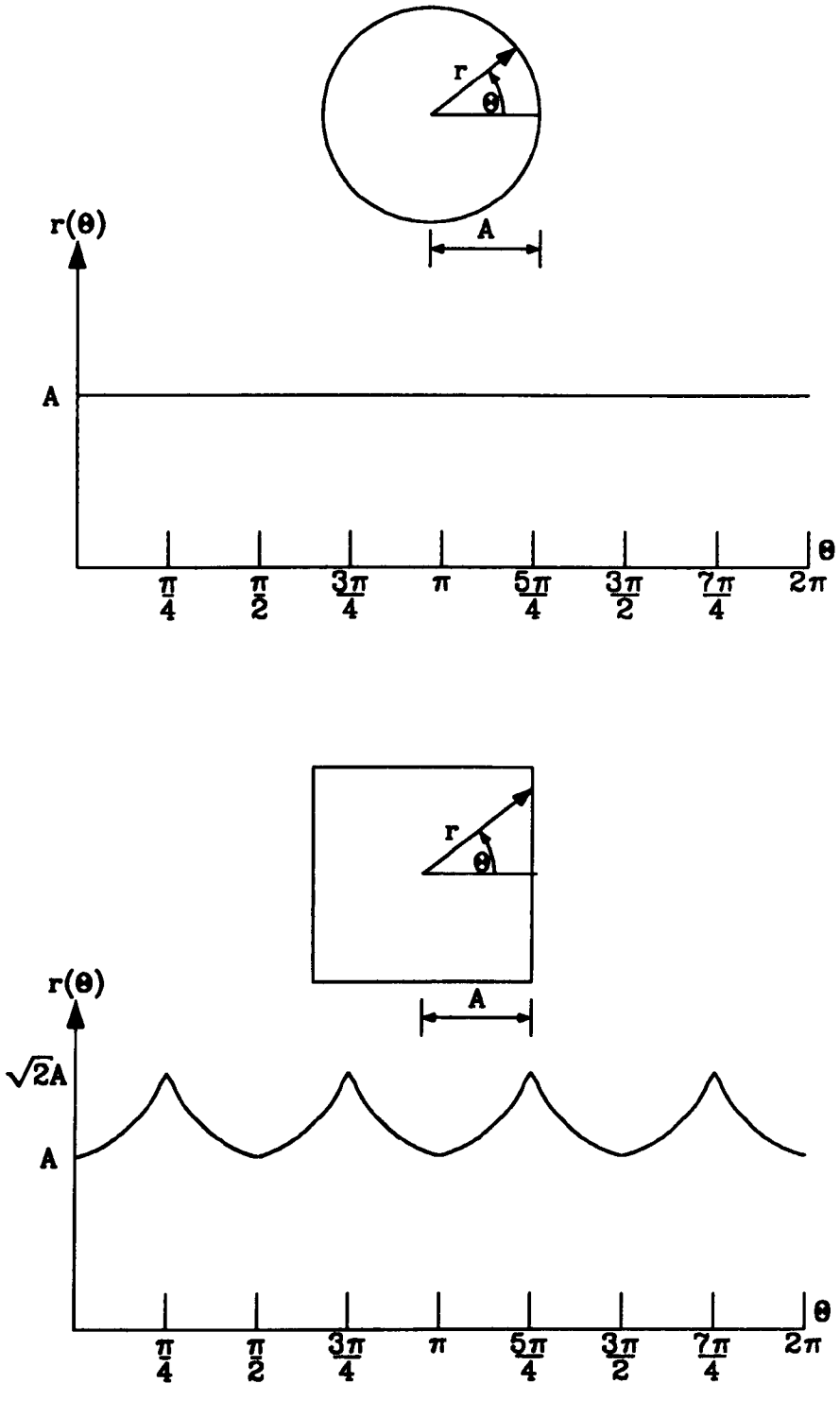
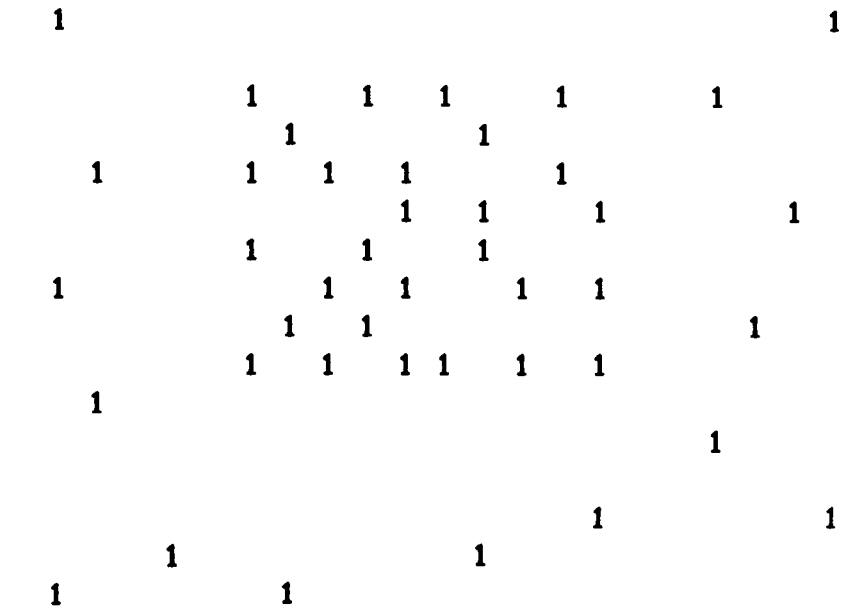
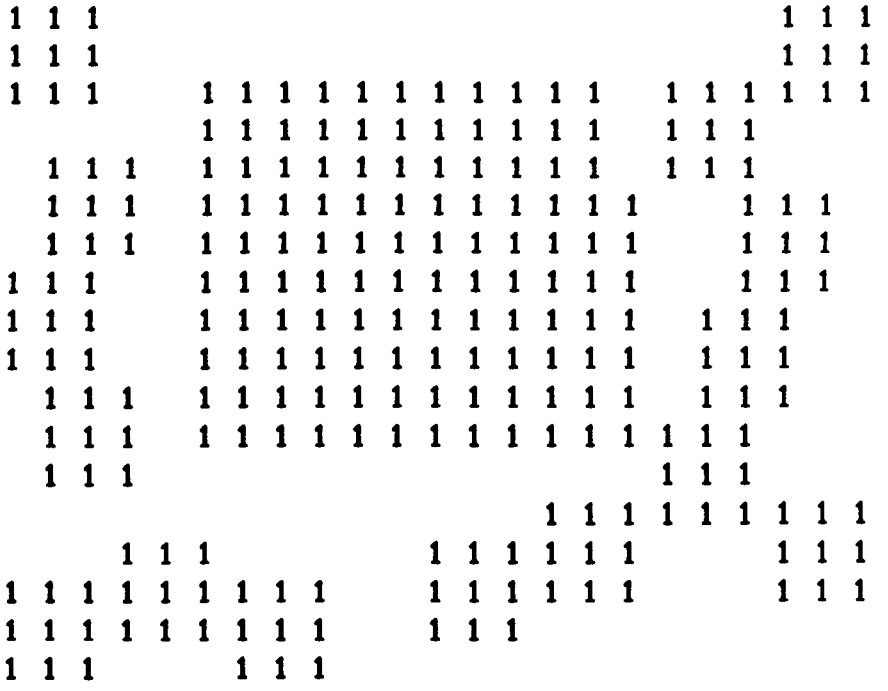


Figure 4.7: DISTANCE VERSUS ANGLE SIGNATURES FOR TWO SIMPLE OBJECTS



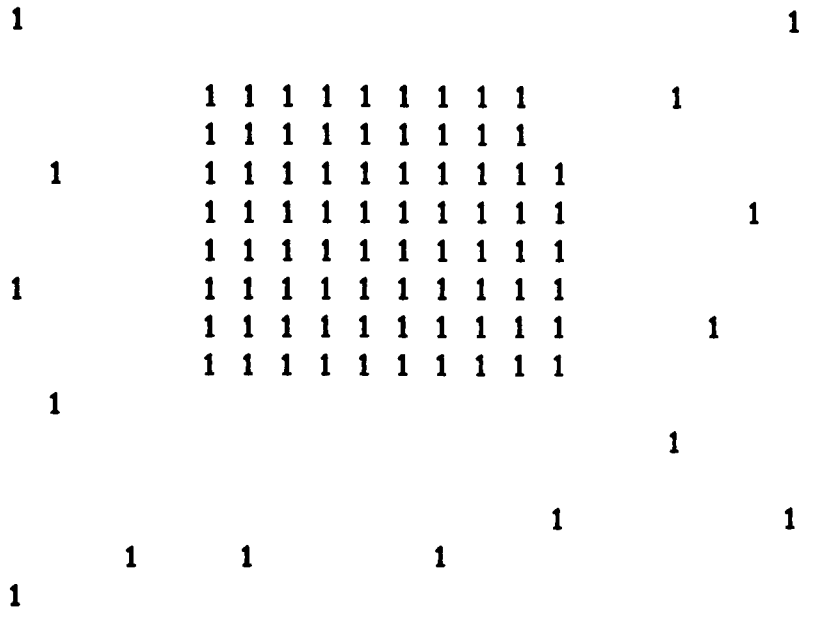
(a)



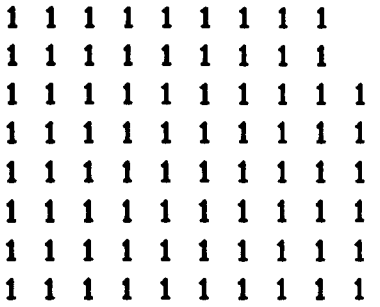
(b)

Figure 4.8: BINARY SMOOTHING BY EXPANDING AND RESHRINKING

(a) original image, (b) results of 8–neighbour expansion of (a) / contd...



(a)



(b)

Figure 4.9: BINARY SMOOTHING BY EXPANDING AND RESHRINKING

(a) result of 8-neighbour shrinking of 4.8(a), (b) result of removing isolated 1's from (a).

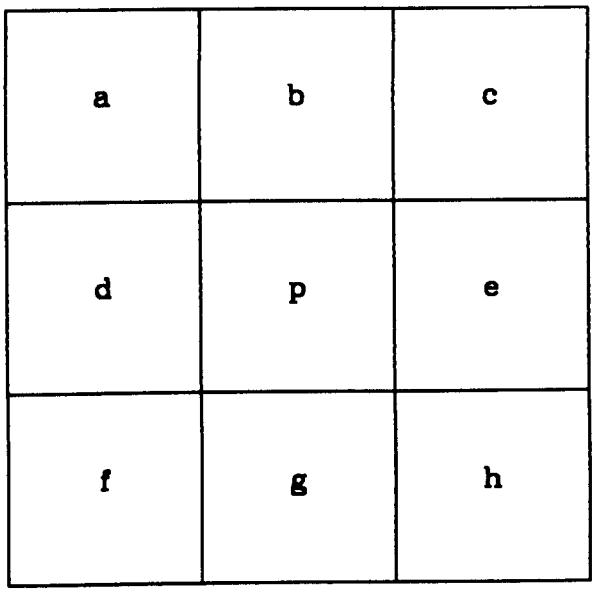


Figure 4.10: NEIGHBOURS OF p USED FOR SMOOTHING BINARY IMAGES

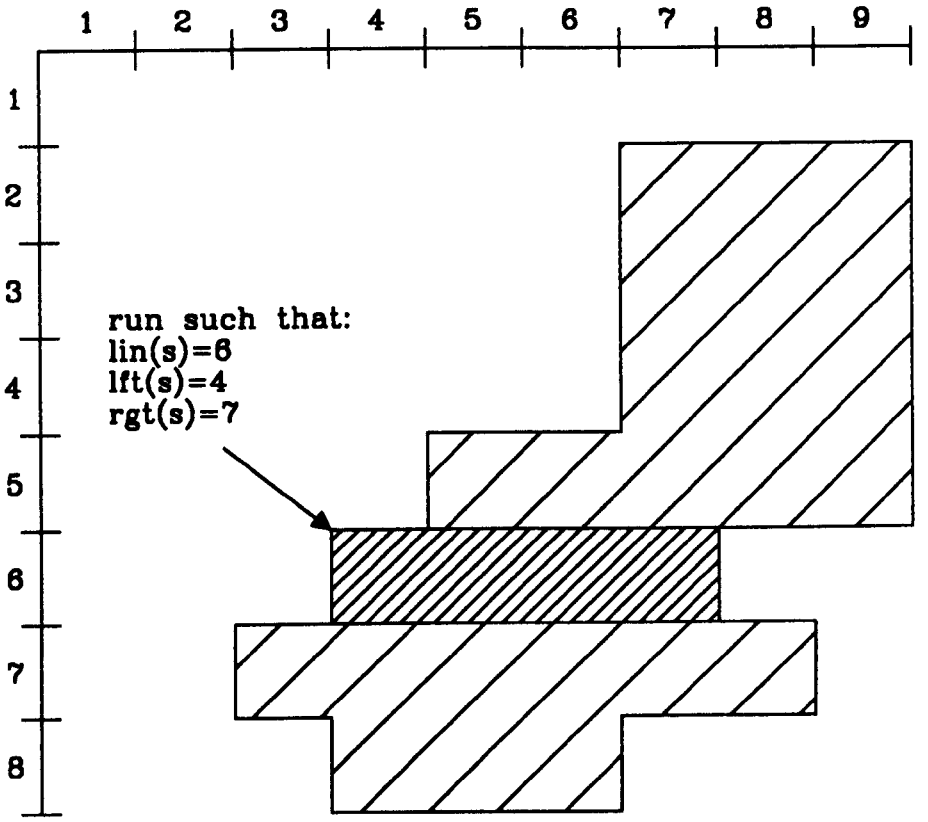


Figure 4.11: EXAMPLE OF A RUN

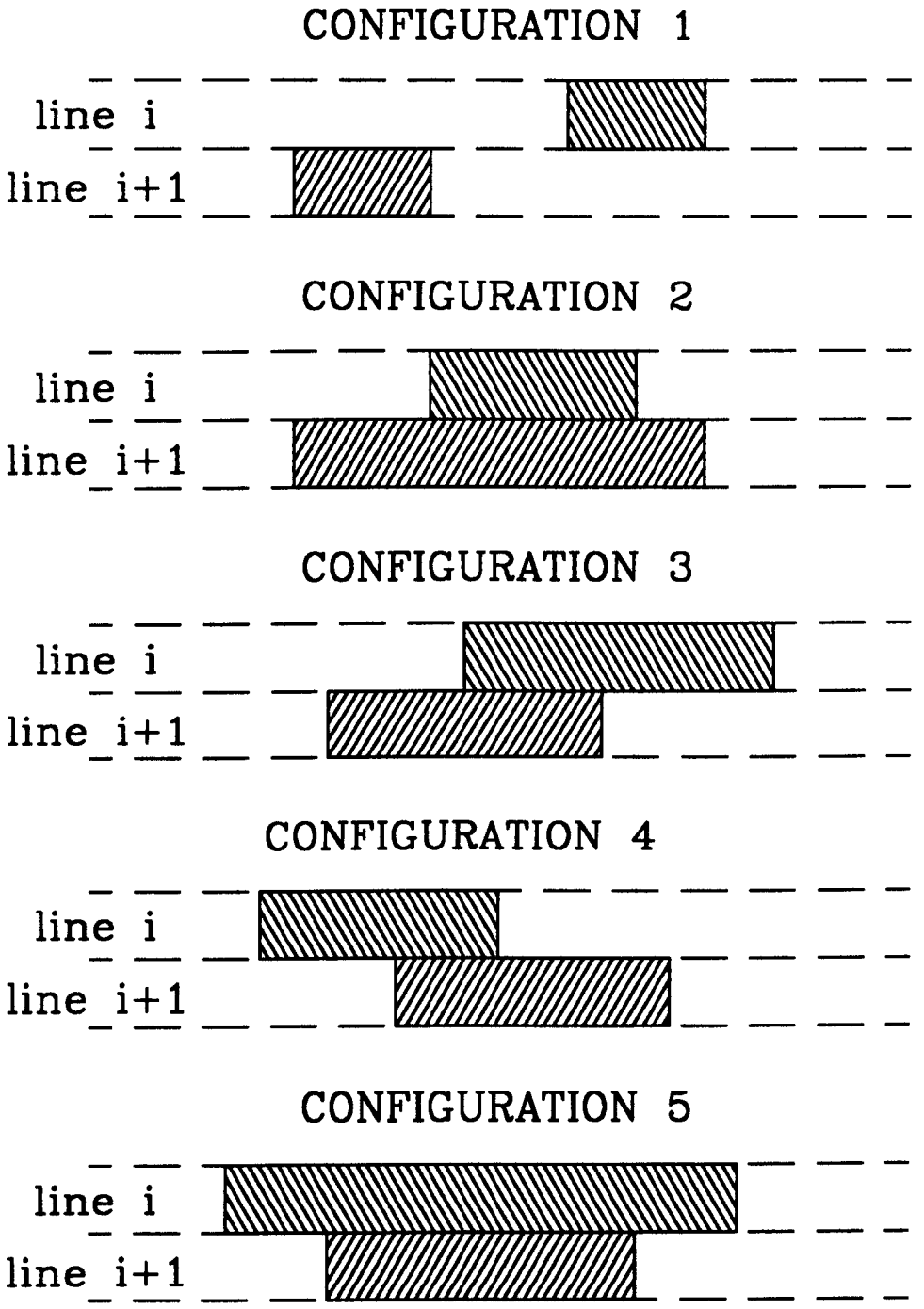


Figure 4.12: EXAMPLE RUN CONFIGURATIONS

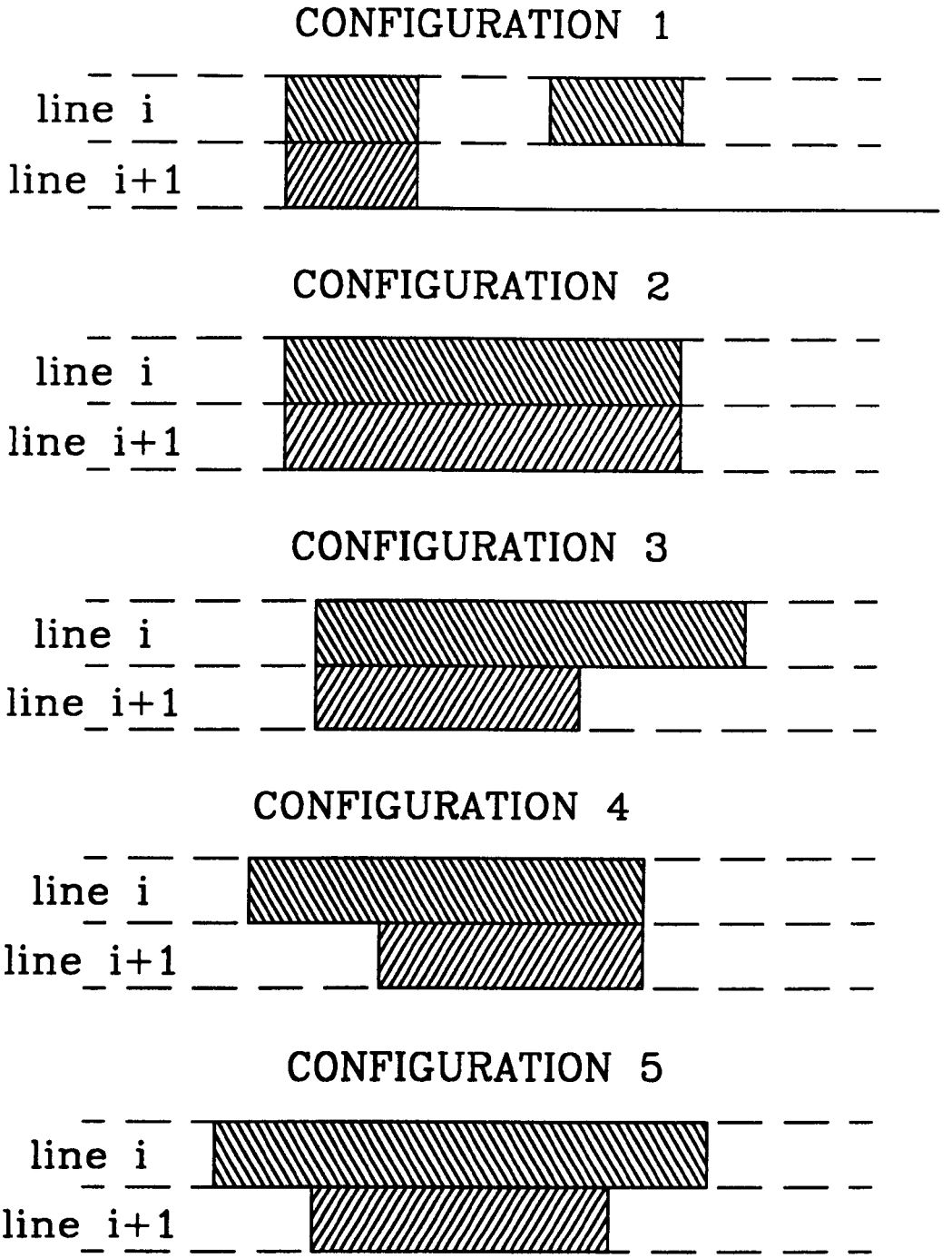


Figure 4.13: EXPANSION OF THE BASIC RUN CONFIGURATIONS

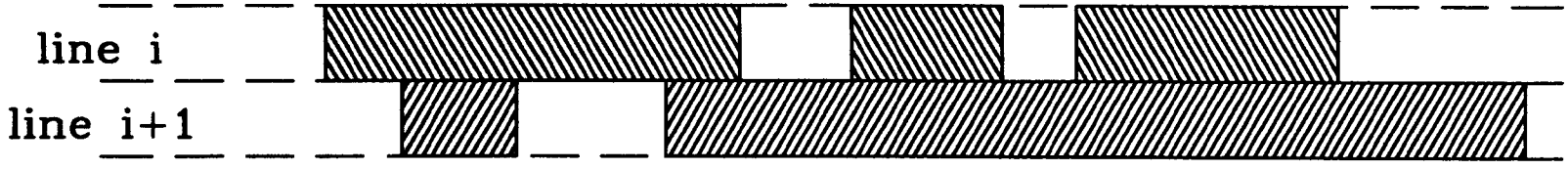


Figure 4.14: EXAMPLE BLOCK CONFIGURATION

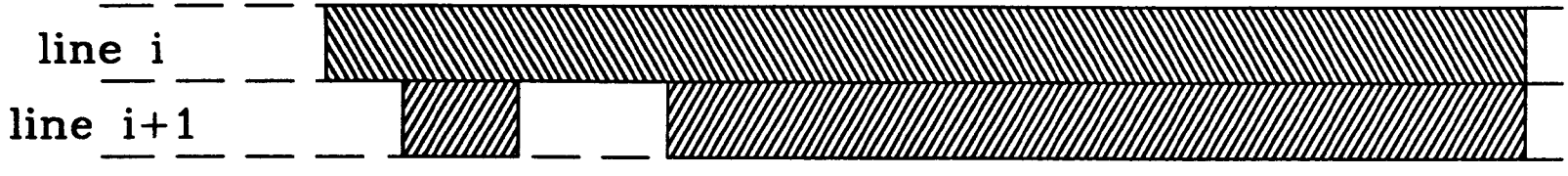


Figure 4.15: EXPANSION OF EXAMPLE BLOCK

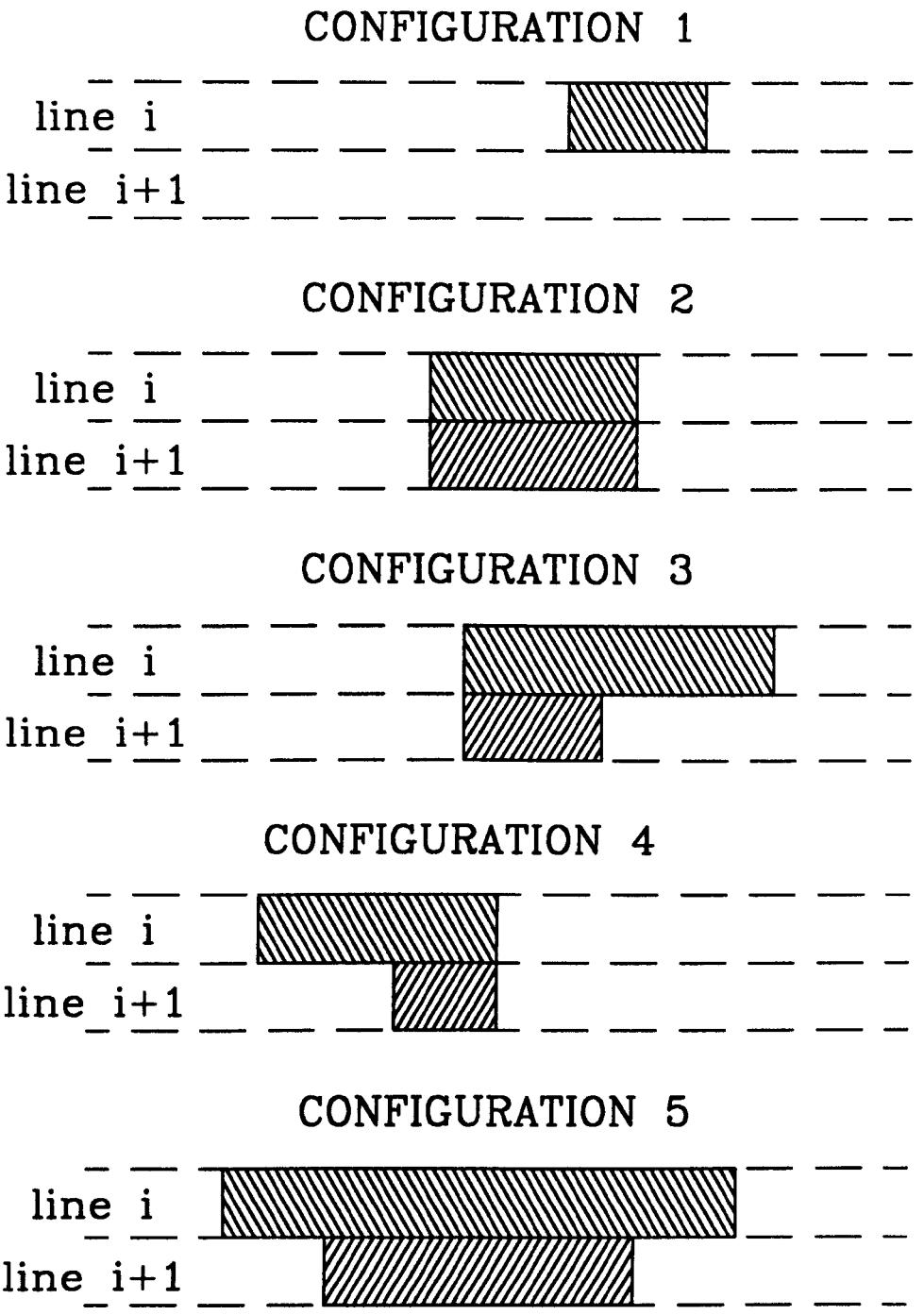


Figure 4.16: SHRINK OPERATION APPLIED TO BASIC RUN CONFIGURATIONS

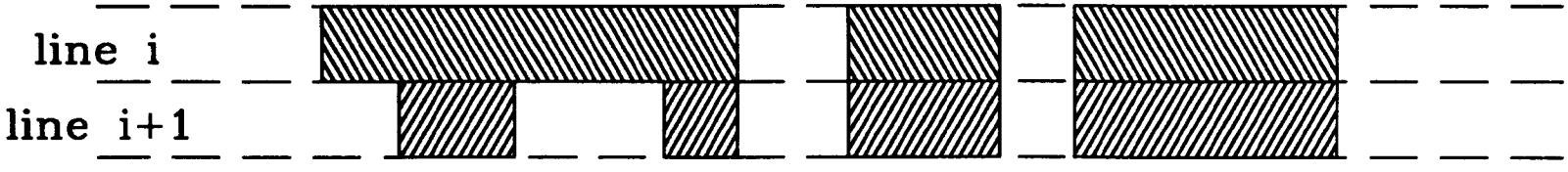


Figure 4.17: SHRINK OPERATION APPLIED TO EXAMPLE BLOCK

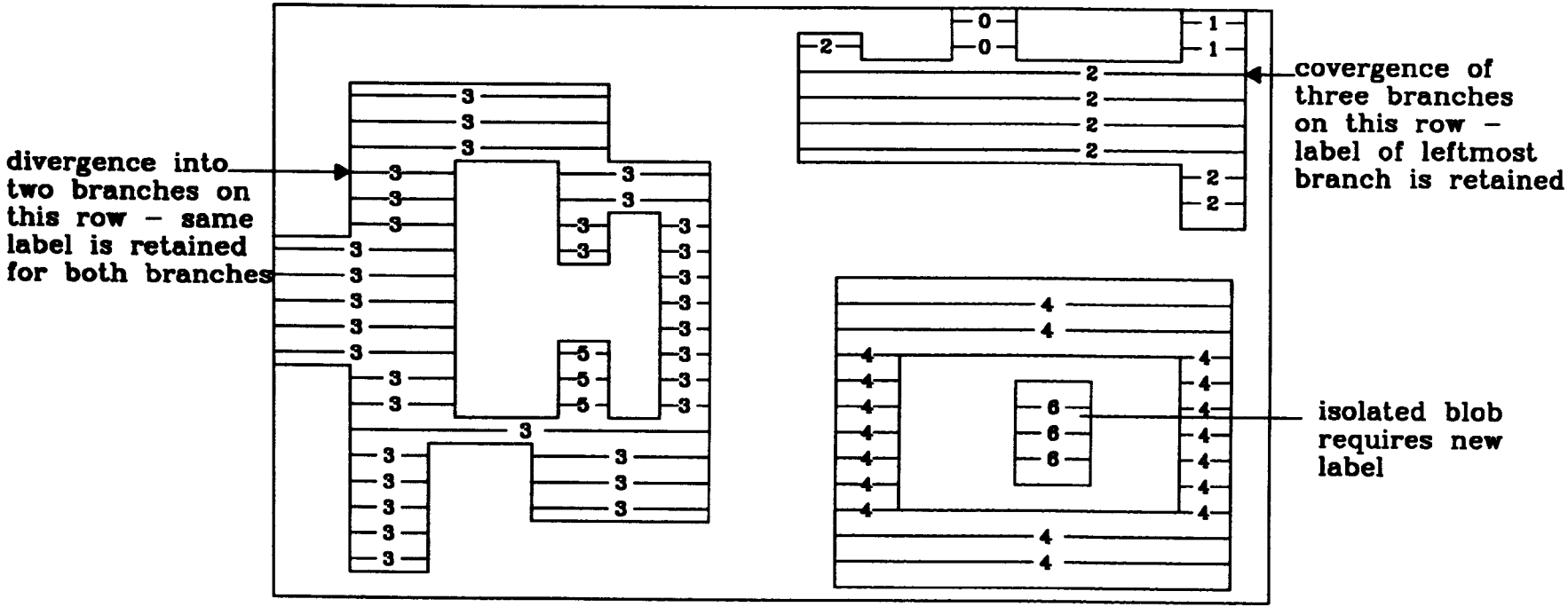


Figure 4.18: REGION LABELING IN ORDER OF APPEARANCE DURING A RASTER SCAN

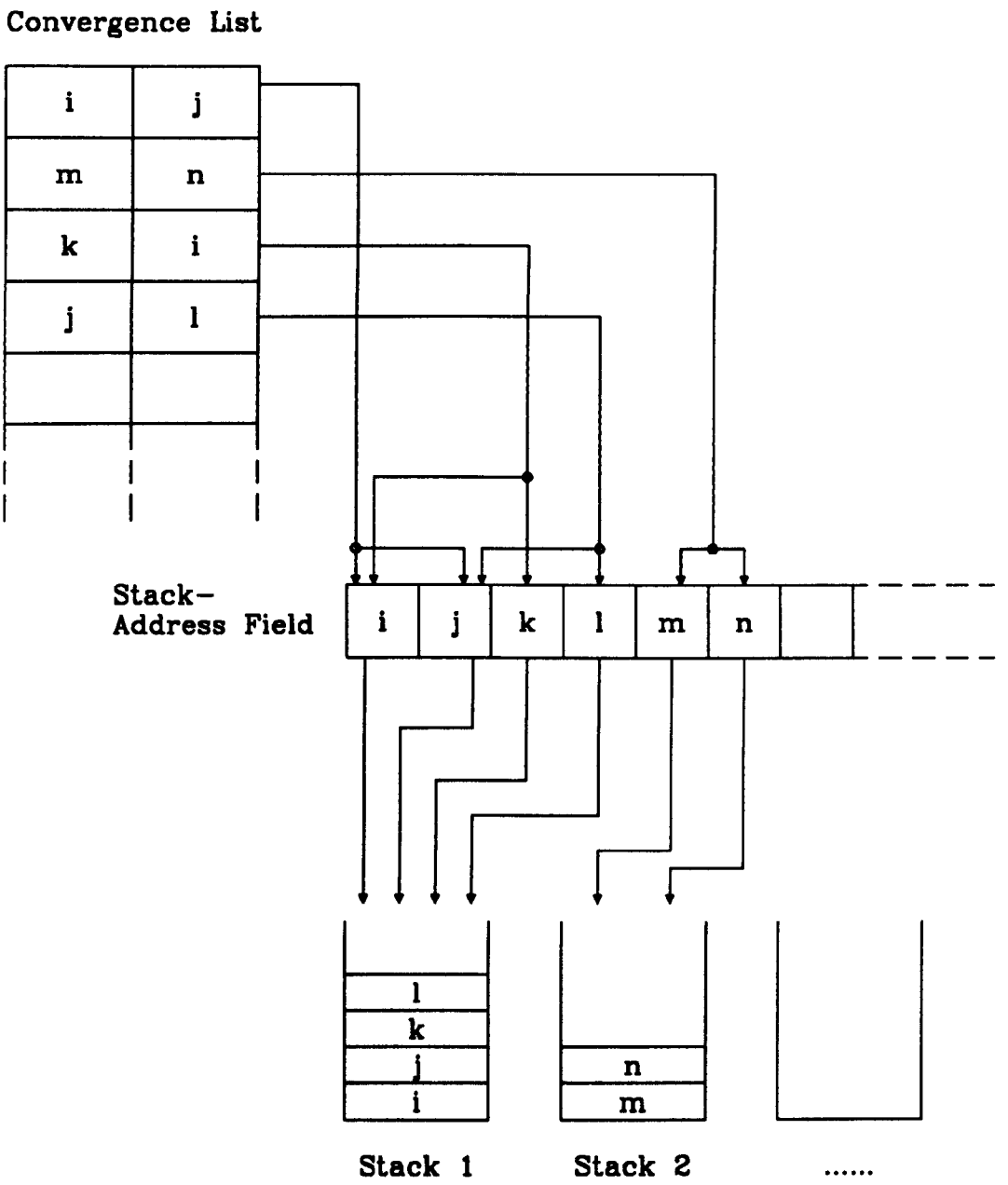


Figure 4.19: LABEL COLLECTION WITH THE AID OF A STACK–ADDRESS FIELD

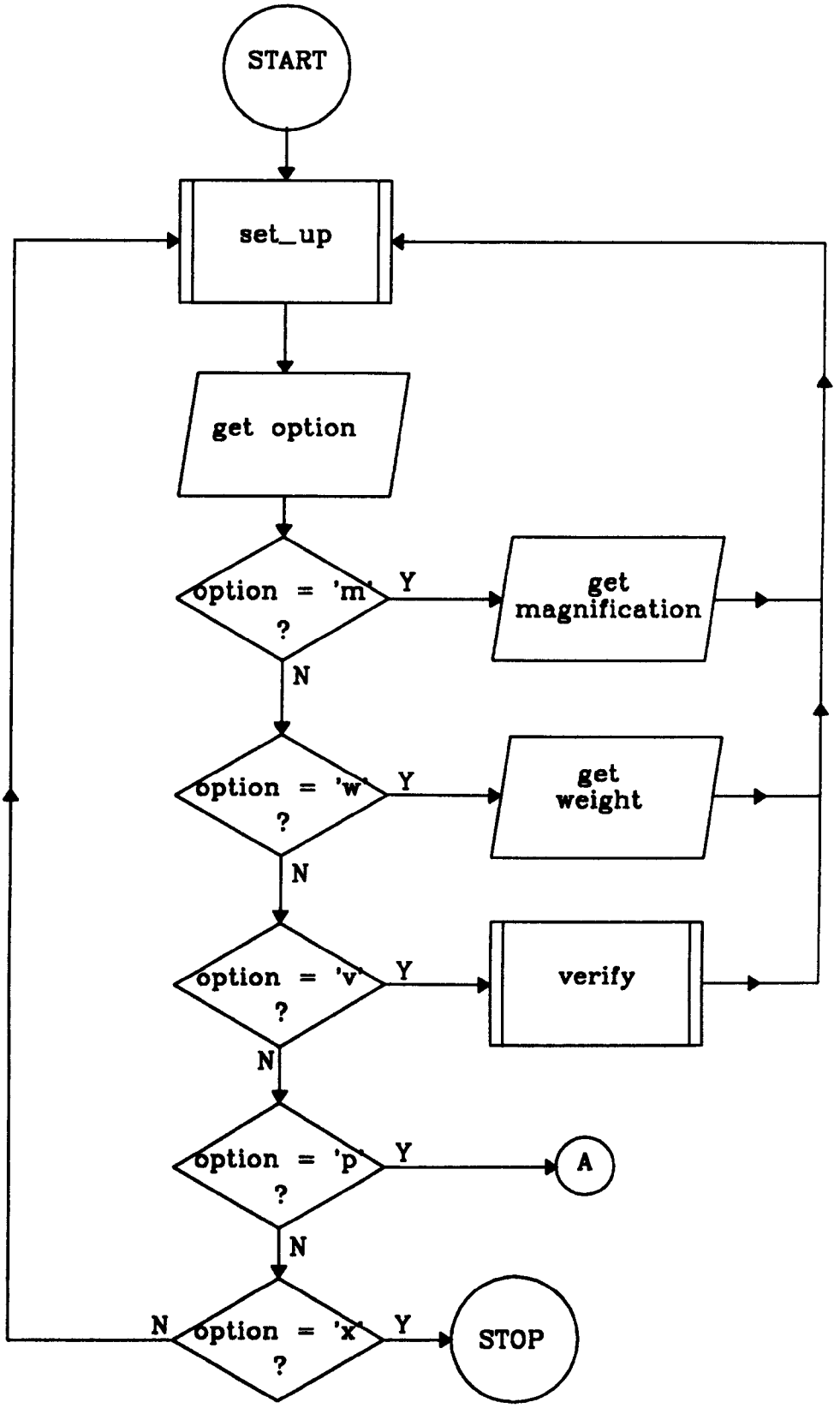


Figure 4.20 FLOWCHART FOR VICKERS TEST PROGRAM

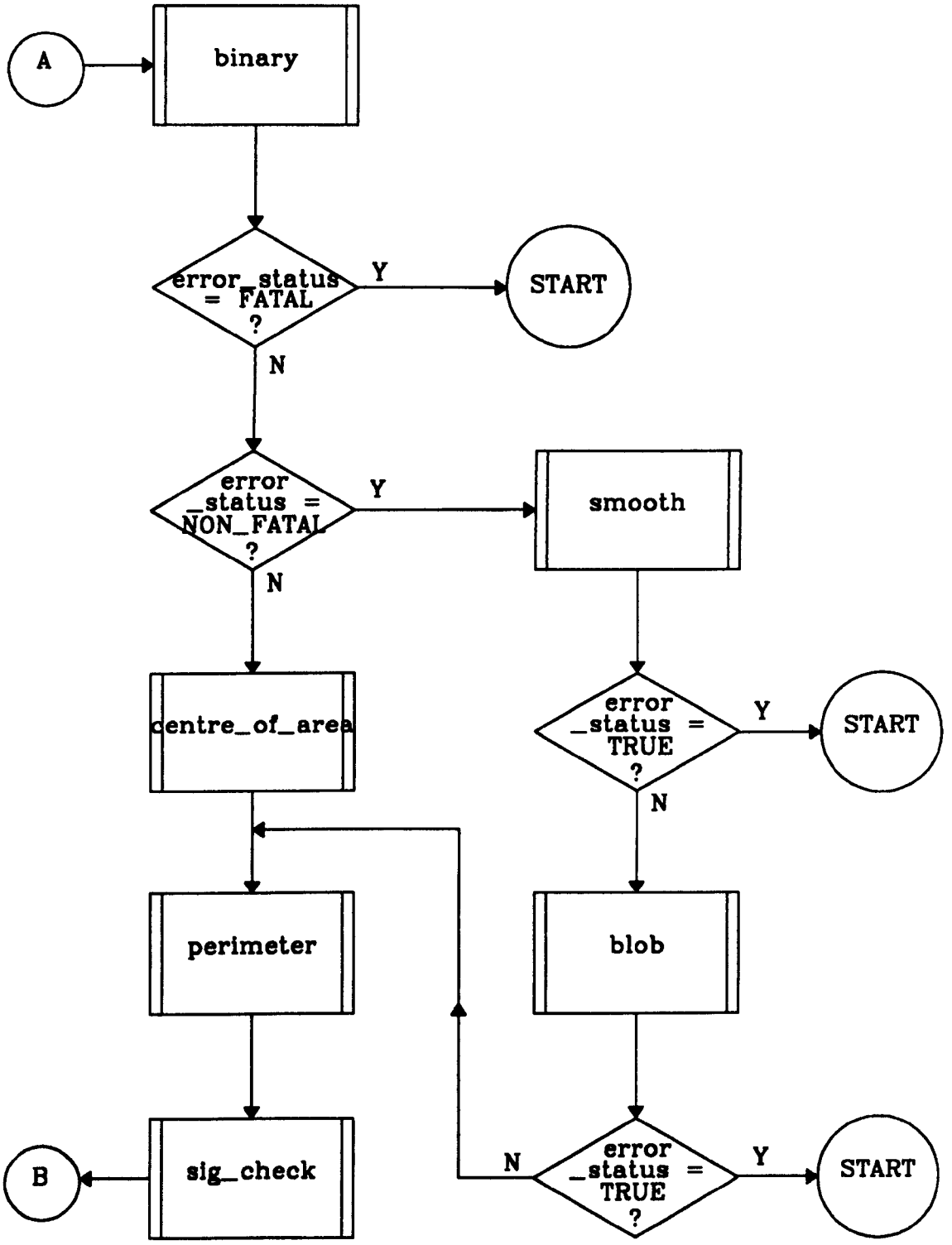


Figure 4.21: FLOWCHART FOR VICKERS TEST PROGRAM /contd.

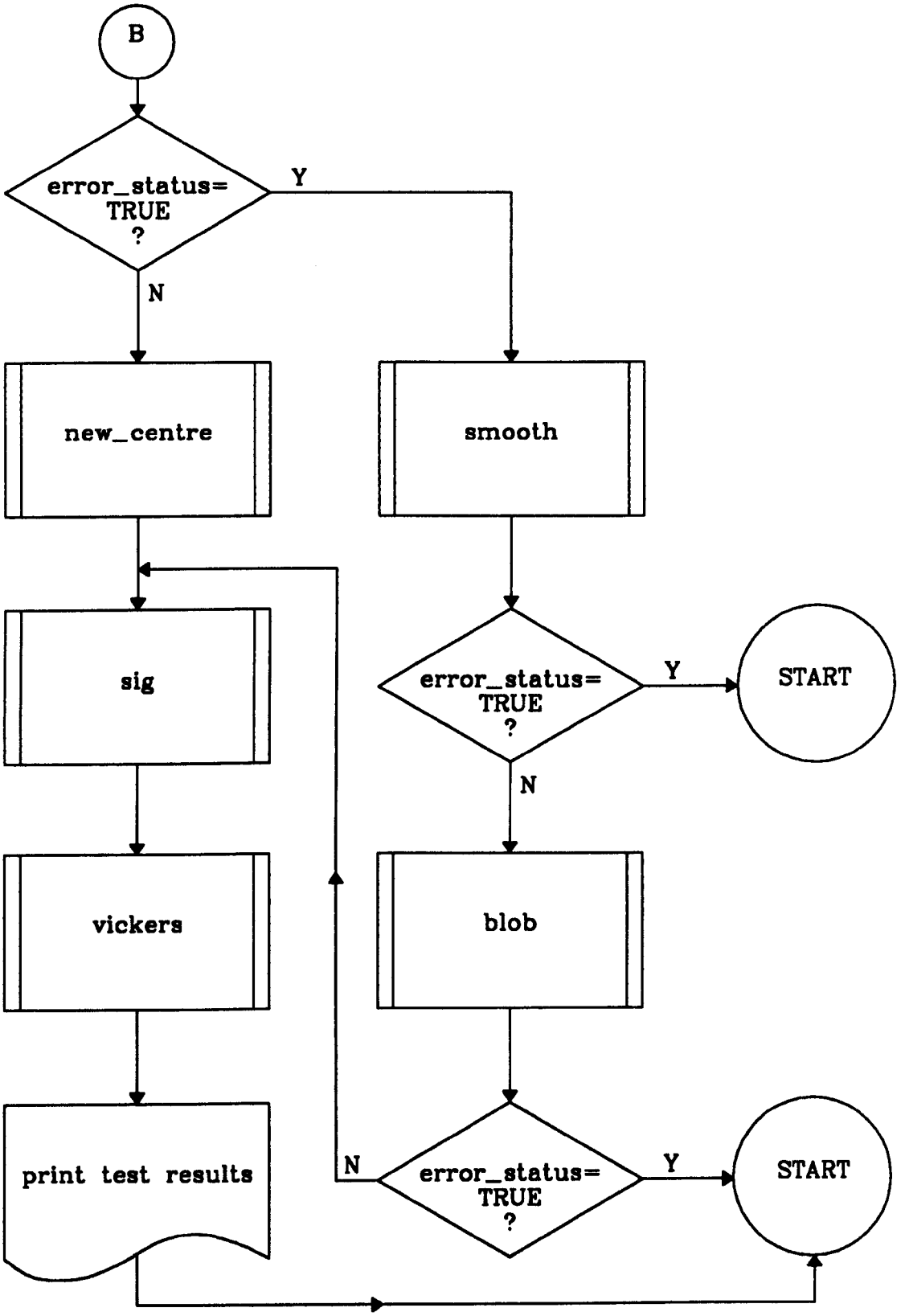


Figure 4.22: FLOWCHART FOR VICKERS TEST PROGRAM / contd.

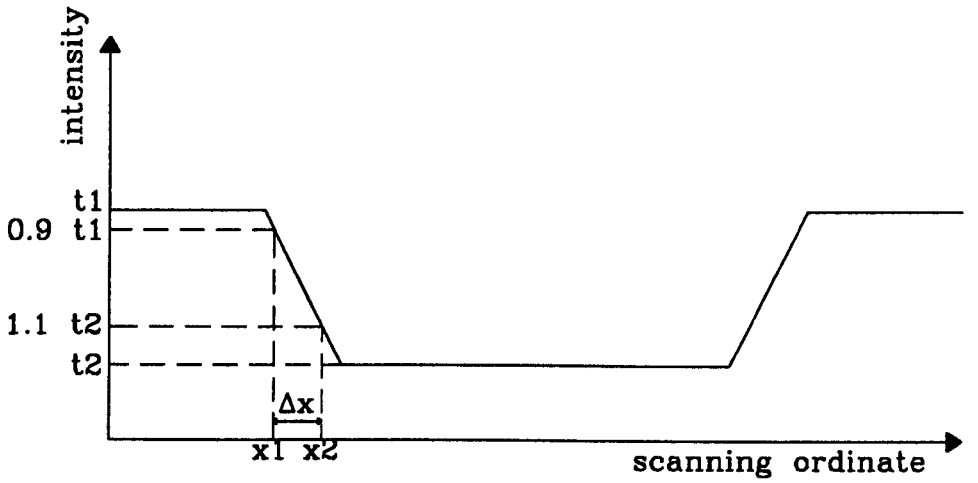


Figure 4.23: GREY LEVEL CROSS-SECTION THROUGH AN INDENTATION

CHAPTER 5

QUALITY ASSESSMENT OF FERRITE COMPONENTS

5.1 INTRODUCTION

The vision system described in this chapter is to be used in the quality assessment of machined components. The components, which are to be examined off-line, consist of ferrite discs, approximately 8 mm in diameter, coated with aluminium. Image data is to be captured via a video camera and frame store with subsequent analysis performed by a personal computer. The system is required to produce a summary of geometric defects found in individual components as well as an indication of other defects such as surface cracks and chips.

A summary of the geometric defects found in the components is given in Fig. 5.1 (the scale here is grossly exaggerated). The degree of eccentricity in both the inner and outer edges of the disc is to be ascertained as well as the offset between the hole centre and the disc centre, should the hole be drilled off-centre. The system is required to indicate the presence of any chip and crack defects on the surface of the component. The length, width, area and position of such defects is to be given and a distinction made between

cracks and chips. Here cracks are defined as fissure like defects, which are long and thin, whereas chips appear as compact regions (see section 2.4.4). Cracks and chips extending to the edges of the disc are to be classified separately. A warning is also to be given if a crack extending from the inner edge of the disc to the outer edge is present since such a defect signals immediate failure of the component.

5.2 SYSTEM HARDWARE

The vision system (Fig. 5.2,5.3) consists of six main components, namely: illumination, camera, frame store, monitors, computer and software. Each of these components is described below.

Illumination

Illumination is provided by a helium-neon laser [37] having peak spectral output at 632 nm wavelength (visible red). To ensure that the whole of the sample is uniformly illuminated, the laser beam is expanded by a confocal lens arrangement to give a spot approximately 10 mm in diameter. Light from the beam expander passes through a 50% reflecting mirror, placed at 45 ° to the medial axis of the beam, onto the sample and background surface. The background surface used was translucent tracing paper which tended to either absorb or scatter incident light, hence only light incident on the sample is reflected back to the mirror and then reflected into the camera. Since the camera effectively views the sample at the specular angle this is a dark field arrangement (see section 2.3.1).

Camera

The camera used to record the image is a Hitachi CCD type AP-120E [38]. This has advantages of less susceptibility to saturation and greater burn-out immunity over the vidicon type. These properties are particularly important when using cameras in conjunction with laser light. To ensure compatibility with the frame store, the camera needs to provide a 625 line, interlaced field, raster operating at 50 Hz. The camera is fitted with a 50 mm focal length lens mounted on a 20 mm extension tube. This arrangement gives maximum coverage of the sample within the field of view. A red filter is fitted to the lens to reduce the effects of changes in ambient light.

Frame Store

The frame store, a Data Translation DT2853 [38], is of the same type as was used in the Shore system (Chapter 3).

Monitors

Two Hitachi video monitors [39] are used in the system; one to monitor the camera output and the other to monitor the frame store output. Both are of the same type as were used in the Shore system (Chapter 3).

Computer

The computer is an Opus PCV [40] and is of the same type as was used in the Shore system (Chapter 3).

Software

The image analysis software was written in Lattice C code [41] and 8086/8088 assembler code [36] was used for the frame store interface routines; these routines are the same as those used in the Shore system (Chapter 3). The program was developed under the MS-DOS operating system [42]. Version 3.30 was used and it is recommended that this be used in future to avoid compatibility problems (the programs will in fact work with any version from 3.00 to 3.30). Software used during the development of the program was the same as that used for the Shore system (Chapter 3).

5.3 SOFTWARE OVERVIEW

5.3.1 INTRODUCTION

The process through which a description of the defects present within a component is arrived at can be broken down into eight main stages, namely:-

(1) An image is captured from the vision camera and thresholded to give a binary (black and white) representation.

(2) An estimate of the centroid of the disc component is made, based on all of the pixels within the image.

(3) The inner border of the disc is traced.

(4) The hole region of the disc is filled and an ellipse fitted to the region so that the eccentricity of the inner edge can be determined.

(5) The outer border of the disc is traced.

(6) The annular region of the disc is filled and an ellipse fitted to the combined hole and annular region so that the eccentricity of the outer edge can be determined.

(7) A plot of deviation between the actual border and the fitted ellipse is produced for both the inner and outer edges of the disc; this allows crack and chip defects extending to the edge of the disc to be identified.

(8) Chip and crack regions are located and the border of each region is traced. The regions corresponding to these defects are then filled and size and shape parameters are calculated for each defect.

A detailed description of the algorithms employed in each of the processing steps is given in the following sections.

5.3.2 THRESHOLDING

The image analysis software for the vision system is based on algorithms specific to binary images, so that the grey scale image produced by the video camera and frame store first needs to be segmented by thresholding. In this application, both the illumination and optical properties of the object and background are essentially constant so that a fixed threshold can be used. All pixels having grey levels between 0 and 20 are labelled as background pixels and those pixels having grey levels outside this range labelled as object pixels. The background pixels (labelled as peak white i.e. grey level 255) form the surface on which the component is placed as well as any chip or crack regions present, while the object pixels (labelled as peak black i.e. grey level 0) form the reflective parts of the component.

5.3.3 ESTIMATION OF THE DISC CENTROID

The first stage in the image analysis process is to locate, roughly, the centre of the disc using the same process as in the the Vickers system (Chapter 4). This is to enable starting points for the inner and outer border following processes (see section 5.3.4) to be determined. The centroid equations are repeated here for convenience:

The coordinates of the centre of area $(\bar{x}, \bar{y})$ are given by:-

$$\bar{x} = \frac{1}{A} \sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} x f(x,y) \qquad [5.1]$$

$$\bar{y} = \frac{1}{A} \sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} y f(x,y) \qquad [5.2]$$

where the area, A is given by:-

$$A = \sum_{x=0}^{x=n-1} \sum_{y=0}^{y=m-1} f(x,y) \qquad [5.3]$$

where n is the number of columns and m the number of rows of pixels in the image array and f(x,y) is a binary valued function defined by:-

$$\begin{aligned} f(x,y) &= 1, \quad g(x,y) = 0 \\ f(x,y) &= 0, \quad g(x,y) = 255 \end{aligned} \qquad [5.4]$$

where g(x,y) represents the thresholded grey scale of a pixel at location (x,y).

5.3.4 BORDER FOLLOWING ALGORITHM

One of the requirements of the vision system is to determine the eccentricity of both the inner and outer edges of the disc component. In order to achieve this, is necessary to identify which pixels in the image belong to the hole region of the disc and which pixels belong to the annular region of the disc so that the ellipse fitting equations (section 5.3.6) may be applied. The delineation is performed by tracing both the inner and outer edges of the disc to form two closed contours; this is referred to as border following. All pixels lying within the inner edge contour are assigned the same unique grey level to identify them as belonging to the hole region whilst all pixels falling inside the outer edge contour, but outside the inner edge contour, are assigned another unique grey level to identify them as belonging to the annular region. The process of identifying and labelling pixels inside a contour is referred to as filling.

The border following algorithm requires as inputs the locations of two pixels one of which forms the starting point of the contour. The pixels need to be adjacent on a row of the image array with one pixel (referred to as c in the algorithm) in the interior of the region to be enclosed by the contour and one pixel (referred to as d in the algorithm) in the exterior. In the case of the inner edge contour the input pixels are located by scanning the image array in the +x direction, from the centre of area, until five contiguous black pixels are found. The first black pixel of the group is then taken as an exterior point whilst it's horizontally adjacent white neighbour is taken as an interior point. In the case of the outer edge contour, the input pixels are located by scanning the image array along a row containing the centroid in the +x direction, starting at the first column of the array until five contiguous black pixels are found. The first black

pixel of the group is then taken as an interior point whilst it's adjacent white pixel is taken as an exterior point.

Two sets of 360 co-ordinates are used to describe the border of the disc, one for the inner edge and one for the outer edge. Each set is stored in an array indexed by angle of rotation. The sets are compiled by first calculating the distance, r, of each border pixel from the centroid as well as it's angle of rotation, θ, about the centroid from the horizontal using the following formulae:-

$$r = \sqrt{(x-\bar{x})^2 S_x^{\,2} + (y-\bar{y})^2 S_y^{\,2}} \qquad [5.5]$$

$$\theta = \arctan\left\{ \frac{S_y(y-\bar{y})}{S_x(x-\bar{y})} \right\} \qquad [5.6]$$

where (x,y) are the co-ordinates of the border pixel, $(\bar{x},\bar{y})$ are the co-ordinates of the centroid and S_x and S_y are the horizontal and vertical scaling factors, respectively.

The co-ordinates are not immediately written to the array; if this were so parts of the outer border having significant concavity would not be recorded (Fig. 5.4) whilst parts of the inner border exhibiting significant convexity would similarly be missed. Instead, in the case of the inner border, the distance value, r, is compared with the distance from centroid of the co-ordinates having a corresponding angle in the border pixel co-ordinate array. The co-ordinates having the largest distance from the centroid are then placed in the array. This is because any chips or cracks extending to the inner edge of the disc increase the size of the border from the centroid and since it is just such defects that are of interest, it is the position of those pixels furthest from the centroid that needs to be retained.

For the outer border the process is repeated with the co-ordinates having the smaller of the two distance values from the centroid being retained. This is because any chips or cracks extending to the outer edge decrease the distance of the border from the centroid so that it is the position of those pixels nearest the centroid that needs to be recorded.

It should be noted that at the stage of processing when the borders of the disc are traced, accurate values for the centroids of the hole region and combined hole and annular region are not available. It is therefore necessary to assume that the centroid of the disc based on the entire image is a fair approximation to the other two centroids. For this reason it is also necessary to record the co-ordinates of border pixels and not their distance from the centroid with respect to angle of rotation. Only when the hole and annular regions have been filled can accurate values of distance from the centroid against angle of rotation be arrived at. In this way an accurate assessment of the edge deviation from the fitted ellipse be arrived at. The border following algorithm (later referred to as BF) was originally developed by Rosenfeld and Kak and is described briefly below (for a comprehensive discussion see [24]).

Let C be a connected component (section 2.4.1) having pixels from the set of object pixels and D be a connected component having pixels from the set of background pixels. The D-border of C is defined as the set of points of C which are adjacent to points of D and it is the location of pixels forming the D-border which is needed in this application. Here 4-adjacency is used and it is assumed that C is 8-connected whilst D is 4-connected. The algorithm requires an initial pair of adjacent pixels as a starting point, c in C and d in D. In this application, when the inner edge of the disc is to be traced C is taken to be the set of background pixels forming the hole region of the disc whilst D is taken to be the set of object pixels forming the annular region of the disc. When the outer edge of the disc is to be traced D is taken to be the set of background

pixels falling outside the disc whilst C is taken to be the set of object pixels forming the annular part of the disc. The algorithm (shown by example in Figs. 5.5 and 5,6) proceeds as follows:-

(1) Change the values of c and d to 3 and 2, respectively.

(2) Let the 8-neighbours of c, in clockwise order, starting with d and ending with the first occurrence of 1, 3, or 4, be $e_1, \ldots, e_k$.

   (a) If $c = 3$ and $e_h = 2$ for some $h < k$, change the 3 to 4, the 2 to 0 and stop.

   (b) Otherwise, change the value of c to 4 (if it was 1). Take e_k as the new c and e_{k-1} as the new d and return to step (2).

When BF terminates, those pixels assigned grey level 4 are the points of the D-border of C.

Once the inner edge of the disc has been traced a check can be made to see if a crack extending from the inner to the outer edge of the disc is present; if this is the case BF will have traced both the inner and outer edges of the disc (Fig. 5.7). Since the positions of those pixels furthest from the centroid are retained in the border pixel co-ordinate array, the array will contain only the positions of pixels forming the outer edge of the disc. If a crack is not present the array will only contain the positions of pixels forming the inner edge of the disc.

The image analysis program calculates the average distance of the border from the centroid, r_{av}, using the border co-ordinates held in the array and compares this value with the mean of the nominal inner edge radius, r_{inner}, and outer edge radius, r_{outer}.

If

$$r_{av} > \frac{1}{2}(r_{inner} + r_{outer}) \qquad [5.7]$$

then the inner border following algorithm will have traced both the inner and outer edges of the disc and therefore the disc contains a crack extending from the inner to the outer edge. Otherwise only the inner edge of the disc has been traced and therefore the disc does not contain a such crack.

5.3.5 CONTOUR FILLING ALGORITHM

There are two methods through which the region encircled by a closed contour can be filled; parity filling and connectivity filling. Parity filling is based on the principle that a straight line will always intersect a closed contour an even (or zero) number of times (Fig. 5.8). The contour may be filled by performing a scan of all of the rows or columns in the image array and recording the number of times a contour pixel is found on each line. If the number of intersections of the scan line with the contour is odd then filling is started whilst if the number of intersections is even filling is stopped. Care must be taken in places where the scan line forms a tangent to the contour; here the point of contact must be counted as two transitions. Parity filling can only be used to fill the contours of "full" regions [48] since in the case of regions which are not "full" filling continues past the contour into the exterior region (Fig. 5.9). In this application, the contour of such a region could be produced by a thin crack extending to the edge of the disc. This problem does not occur with connectivity filling and for this reason a connectivity based algorithm was used here.

In the connectivity filling method an initial "seed" pixel is given and all pixels which have a path, not crossing the contour, to this pixel are located. The simplest way of performing connectivity filling is to scan from left to right across the image array, filling each new pixel, until a contour pixel is reached. The direction of the scan is then reversed on the next lower line and filling continued until the contour is again reached; the process is then continued on subsequent lines. Once the lower part of the region has been filled a similar upward scan is performed to fill the upper part of the region. The main drawback with this approach is that it can only be used to fill convex regions; for non-convex regions (Fig. 5.10) the location of local maxima and minima need to be recorded by placing the positions of pixels marking turning points on a stack. The algorithm used in this application is based on this approach and was originally developed by Pavlidis. The algorithm is discussed briefly below; for a detailed discussion see [48].

As noted earlier, when filling the interior region of the contour care must be exercised with parts of the contour forming local maxima or minima. In the connectivity filling algorithm a function, LINK, is used to give information about the contour curvature. The argument of the function is the location of the leftmost pixel, on a given line, which is to the right of the contour and the function returns two values *above* and *below*. *above* gives the number of 8-connected runs of contour pixels (see section 4.4.3) between the present line and the next upper line, whilst *below* gives the number of 8-connected runs between the present line and the next lower line (Fig. 5.11). The set *(above,below)* is known as the degree of the contour and can take one of the following values: (1,1) for a simple curve, (2,0) for a local minimum, or (0,2) for a local maximum. Other values are only possible for regions which are not full but do not cause problems in the main region filling algorithm. The LINK algorithm is given below:-

0. Input is a pixel on the contour at (x,y). Outputs are the values *above* and *below*.

1. Initialise *above* and *below* to zero.
2. IF (x-1,y+1) belongs to the contour, THEN increment *above*.
3. IF (x-1,y-1) belongs to the contour, THEN increment *below*.
4. WHILE (x,y) belongs to the contour DO steps 5-7.

   BEGIN.

5. IF (x,y+1) belongs to the contour and (x-1,y+1) does not, THEN increment *above*.
6. IF (x,y-1) belongs to the contour and (x-1,y-1) does not, THEN increment *below*.
7. Increment x.

   END.

8. IF (x-1,y+1) does not belong to the contour and (x,y+1) does, THEN increment *below*.
9. IF (x+1,y-1) does not belong to the contour and (x,y+1) does, THEN increment *above*.
10. RETURN location of pixel (x,y) and the values *above* and *below*.
11. END OF ALGORITHM.

The location of a pixel is later referred to as it's address, denoted by p. The address of the pixel to its left is denoted by p-1 whilst the address of the pixel to its right is denoted by p+1. It is convenient to define a function LEFT(p) which returns the address of the leftmost pixel in a given run containing p as well as a function LRIGHT which returns the address of the rightmost pixel to the left of p having the same grey level, but with the property that there is at least one pixel between the two having a different grey level (Fig. 5.12). The latter function can be written as:-

$$\text{LRIGHT}(p) = \text{LEFT}(\text{LEFT}(p)-1)-1$$

Before discussing the main algorithm it is also necessary to modify LINK to specify the addresses e_1 and e_2 in Fig. 5.12. If any of these addresses is not defined then LINK should return an illegal value; here this is taken to be -1. Furthermore, the argument of link is taken to be any pixel in the interval $[p_1, p_2]$

The main algorithm fills lines from left to right starting at a pixel immediately to the right of the contour. On the first line the starting point is given by $p = \text{LEFT}(p_{seed})$ where p_{seed} is the address of the seed pixel; LINK(p-1) is then used to find the degree of the left contour. The address of the starting pixel on the next line, p_{next} is set to e_1 if the scan direction is downwards or e_2 if the scan direction is upwards. If the degree is (1,1) a simple arc has been found and filling continues until the right contour is encountered; the address of the last pixel before the contour being p_{right}. LINK($p_{\text{right}} + 1$) is then used to find the degree of the contour; if this is (1,1) a simple arc has been found and p is replaced by p_{next} and the process is repeated. The above process forms the basis of steps 5-17 of the main algorithm which is given below:-

0. Inputs are the image array, the address of an interior pixel *seed* and a second pixel $seed_{ontop}$.
   LEFT(*seed*) is placed in stack S and the direction *down* in stack S_d.
   LEFT($seed_{ontop}$) is also placed in S and direction *up* in S_d
1. WHILE the stack, S is not empty REPEAT steps 2-19.
   BEGIN.
2. $p_{next} = \text{POP}(S)$, $dir = \text{POP}(S_d)$, $p_{\text{right}} = X_{\max}$.
3. IF *dir* equals *down*, THEN set $u = 2$, $other = under$.
   ELSE $u = 1$, $other = ontop$.

4. REPEAT steps 5-16

BEGIN.

5. IF p_{next} equals -1, OR IF p_{next} is filled, THEN EXIT from the loop.

6. $p = p_{next}$.

7. LINK($p - 1$)

8. IF [(*dir* equals down and *above* exceeds 1) or (*dir* equals *up* and *below* exceeds 1)] and p is to the right of p_{right} THEN EXIT from the loop.

9. IF both *above* and *below* exceed zero, THEN set $p_{next} = e_u$.

ELSE DO:

BEGIN.

10. IF p is not already filled, THEN place in S the address LEFT(LRIGHT(p)) and in S_d the value *dir*.

11. IF (*above* equals zero and *below* does not and *dir* equals down) or (*below* equals zero and *above* does not and *dir* equals up), THEN set p_{next} = LEFT(p_{other}) .

12. ELSE IF p_{other} is not filled, THEN set p_{next} = LEFT(p_{other}).

13. ELSE set $p_{next} = -1$.

END.

14. Fill the line starting from p and let p_{right} be the last pixel before the contour.

15. LINK($p_{\text{right}} + 1$)

16. If either *above* or *below* equals 0 and if p_1 is not filled,

THEN DO:

BEGIN.

17. Place p_1 in stack S.

18. IF *above* > 0, then place *up* in S_d.

19. ELSE place down in S_d.

END.

END.

END.

20. END OF ALGORITHM.

Step 5 checks if the next starting pixel has an illegal address or if the pixel is already filled. If either is the case the next seed pixel is popped from the stack and the scanning process is repeated. If the stack is empty the algorithm is terminated. Step 7 checks if the bottom of the contour has been reached, if the scan direction is downwards, or if the top has been reached, if the direction is upwards. In either case the inner loop is exited and the next seed popped from the stack. Steps 9-13 are used to check for local maxima and minima. If the scan direction is downwards and a local maximum is found (*above* = 0) a seed pixel must be placed on the stack whose address is LEFT(LRIGHT(p)), unless p has already been filled. If the scan direction is upwards and a local minimum is found (*below* = 0) a seed pixel must be placed on the stack whose address is LEFT(LRIGHT(p)), unless p has already been filled. Step 14 performs the actual filling of the interior region.

5.3.6 ELLIPSE FITTING

Once the inner and outer edges of the disc have been traced, the next stage is to fit an ellipse to the hole region so that the degree of eccentricity in the inner edge can be determined, this process is called recovering the approximating ellipse [32]. The ellipse equations are then applied to *all* pixels falling within the outer edge contour so that the

centroid of the combined hole and annular region is found. The offset between the hole centre and the disc can the be calculated, taking into account the scaling factors.

An ideal ellipse may be specified by five parameters, two in size (major and minor axis length) , two in position (centroid in x and y) and one in orientation (rotation). The centroid co-ordinates are given in equations [5.1],[5.2] while the other parameters are given by:-

$$A = \frac{4}{\pi} S_x^3 S_y \left[\sum x^2 - \frac{\left(\sum x\right)^2}{\sum f(x,y)} \right] \qquad [5.8]$$

$$B = \frac{4}{\pi} S_x S_y^3 \left[\sum y^2 - \frac{\left(\sum y\right)^2}{\sum f(x,y)} \right] \qquad [5.9]$$

$$C = \frac{4}{\pi} S_x^2 S_y^2 \left[\sum xy - \frac{\sum x \sum y}{\sum f(x,y)} \right] \qquad [5.10]$$

$$E = \sqrt{(A - B)^2 + \frac{4}{\pi} C^2} \qquad [5.11]$$

$$F = (AB - C^2)^{\frac{1}{4}} \qquad [5.12]$$

$$\text{major axis} = b = \sqrt{\frac{A + B + E}{2F}} \qquad [5.13]$$

$$\text{minor axis} = a = \sqrt{\frac{A + B - C}{2F}} \qquad [5.14]$$

$$\text{rotation} = \phi = \frac{1}{2}\arctan\left(\frac{2C}{A - B}\right) \qquad [5.15]$$

$$\text{area of region} = S_x S_y \sum f(x,y) \qquad [5.16]$$

where S_x and S_y are the horizontal and vertical scaling factors of the vision system in physical units/pixel and the summations are taken over all the pixels within the region to which the ellipse is to be fitted. It may seem anomalous that the area has been normalised with respect to the scaling factors whereas the co-ordinates of the centroid have not. However, in the following analysis the origin of the image array is effectively shifted to the centroid so that when the distance of a pixel from the centroid is required the scaling factors are *then* taken into account rather than transforming the co-ordinates of *all* pixels according to the scaling factors *initially*.

In the vision system software, the amount of time required for computation of the ellipse parameters is reduced by taking account of the position of the x co-ordinates of the left and right contours, x_l and x_r respectively, during the region filling algorithm. The summations over x, y, and f(x,y) may be simplified, as in section 4.4.5, to:-

$$\sum_{x=x_l}^{x=x_r} x = \frac{1}{2}(x_r - x_l + 1)(x_r - x_l) \qquad [5.17]$$

$$\sum_{x=x_l}^{x=x_r} y = y(x_r - x_l + 1) \qquad [5.18]$$

$$\sum_{x=x_l}^{x=x_r} f(x,y) = (x_r - x_l + 1) \qquad [5.19]$$

The other summations may be simplified as follows:-

Consider $\sum x^2$:-

$$\sum_{1}^{n} r^2 = \frac{1}{6} n(n + 1)(2n + 1) \qquad [5.20]$$

(the derivation of this identity is given in [54])

$$\sum_{1}^{N} r^2 = \sum_{1}^{n} r^2 + \sum_{n+1}^{N} r^2 \qquad [5.21]$$

$$\sum_{n+1}^{N} r^2 = \sum_{1}^{N} r^2 - \sum_{1}^{n} r^2 \qquad [5.22]$$

$$\begin{aligned} \sum_{x=x_l}^{x=x_r} x^2 &= \sum_{x=1}^{x=x_r} x^2 - \sum_{x=1}^{x=x_l-1} x^2 \\ &= \frac{1}{6}\{x_r(x_r + 1)(2x_r + 1) - (x_l - 1)(x_l + 1 - 1)(2(x_l - 1) + 1)\} \\ &= \frac{1}{6}\{x_r(x_r + 1)(2x_r + 1) - x_l(x_l - 1)(2x_l - 1)\} \end{aligned} \qquad [5.23]$$

Consider $\sum y^2$:-

$$\sum_{x=x_l}^{x=x_r} y^2 = y^2 \sum_{x=x_l}^{x=x_r} 1 \qquad [5.24]$$
$$= y^2 (x_r - x_l + 1)$$

(from [5.17]). Consider $\sum xy$

$$\sum_{x=x_l}^{x=x_r} xy = y \sum_{x=x_l}^{x=x_r} x \qquad [5.25]$$
$$= \frac{1}{2}(x_r - x_l + 1)(x_r - x_l)\, y$$

(From [5.17])

5.3.7 PROFILE COMPUTATION

The next stage is to determine if any chips or cracks extending to the edge of the disc are present. This is achieved by producing a plot (here referred to as the edge profile) of deviation of the border from the fitted ellipse against angle of rotation for both the inner and outer edges. In the profile, chips appear as relatively small deviations extending over wide angular ranges while cracks are shown by large deviations present over small intervals. The maximum, minimum and RMS deviation are also calculated at this stage. Calculation of the profile is complicated by the fact that the ellipse parameters are calculated in physical units, taking into account the scaling factors of the system, whereas the disc borders are defined by set of pixel co-ordinates. This problem is resolved as follows.

Let the distance, in pixels, of a border point located at (x,y) from the centroid, located at $(\bar{x},\bar{y})$, be $r(\theta)$, where θ is the angle of rotation of the point about the centroid, from the horizontal. Clearly:-

$$r(\theta) = \sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} \qquad [5.26]$$

$$\theta = \arctan\left\{ \frac{y - \bar{y}}{x - \bar{x}} \right\} \qquad [5.27]$$

Let $r_{phys}(\theta)$ be the physical distance corresponding to $r(\theta)$, so that:-

$$r_{phys}(\theta) = \sqrt{r_x(\theta)^2 + r_y(\theta)^2} \qquad [5.28]$$

where:-

$$r_x(\theta) = S_x r(\theta) \cos\theta \qquad [5.29]$$

$$r_y(\theta) = S_y r(\theta) \sin\theta \qquad [5.30]$$

where S_x and S_y are the horizontal and vertical scaling factors of the vision system, respectively, in physical units / pixel. Equations [5.29],[5.30] produce an implicit change in the angle of rotation of the border point such that the new angle α is given by:-

$$\alpha = \arctan\left\{ \frac{r_y}{r_x} \right\} \qquad [5.31]$$

The distance, $r_{ellipse}(\theta)$, from the centroid of a point on the fitted ellipse, at an angle θ from the horizontal, in physical units is given by:-

$$r_{ellipse}(\theta) = \sqrt{(r_{xellipse}(\theta)^2 + r_{yellipse}(\theta)^2)} \qquad [5.32]$$

where:

$$r_{xellipse}(\theta) = b\cos(\theta - \phi) \qquad [5.33]$$

$$r_{yellipse}(\theta) = a\sin(\theta - \phi) \qquad [5.34]$$

so that the deviation between the actual border and the fitted ellipse r_{dev} is given by:-

$$r_{dev}(\theta) = r_{ellipse}(\alpha) - r_{phys}(\theta) \qquad [5.35]$$

5.4 DESCRIPTION OF DEFECTS

The final part of the analysis is to locate any chip or crack defects on the surface of the disc. The image is scanned row by row until a pixel labelled as a crack or chip is located, a pair of pixels suitable as a starting point for the border following algorithm are then found and the outline of the defect is traced using the algorithm given in section 5.3.4. The perimeter length of the defect is also calculated at this stage using the following empirical formula given by Agin [32]:-

$$P \approx 0.948(S_x N_h + S_y N_V - \frac{1}{2}(S_x + S_y - \sqrt{S_x^2 + S_y^2})N_c) \qquad [5.36]$$

where S_x and S_y are the horizontal and vertical scaling factors, respectively, and N_h is the number of horizontal, N_v the number of vertical segments and N_c the number of corner (or diagonal) segments in the contour.

The perimeter length of an object can vary widely according to it's orientation if care is not taken in the calculation. Consider a square object, 10 pixels across (Fig. 5.13); when the object is aligned with the image array it's perimeter length is 40 pixels, however if the object is rotated by 45 ° the apparent perimeter length is 54 pixels. By cutting the

corners on the perimeter a better approximation to the perimeter length can be obtained. However in the case of an ideal circle the calculated perimeter length will still be too high by a factor of $(8/\pi)(\sqrt{2}-1) = 5.48\ \%$ [32]. Equation [5.36] takes account of both the corner cutting and the expected error.

Once the border of the defect has been traced the interior region is then filled using connectivity based algorithm given in section 5.3.5. The border filling algorithm requires two "seed" pixels, one vertically above the other, so that there is a minimum size condition for regions which can be filled. The condition is that for a region to be filled there must exist two, vertically 4-adjacent, defect labelled pixels each having eight 8-adjacent pixels labelled as defect pixels (Fig. 5.14).

The starting point for the border tracing algorithm is found by first locating a pair of pixels satisfying the above condition then scanning the image in the +x direction until a pair of dissimilar, horizontally 4-adjacent, pixels are found. The location of these pixels is then used as the starting point.

It should be noted that an error can occur if the two defect labelled regions are 4-connected by a single pixel (Fig. 5.14). Here the boundary following routine produces a contour which does not encircle the seed pixels. A solution to this is to scan the image, from the lower seed pixel, in the -x direction until a pixel not labelled as a defect is found. If this pixel is a boundary pixel then the contour encircles the seed pixels and the region can be filled, otherwise the seed pixel is not encircled by the contour and the original image scanning process continues until a suitable pair of seed pixels are found. Note that in the latter case, as in the case of two defect labelled regions 4-connected by a pair of vertically 4-adjacent pixels (Fig. 5.15), the border of the region not containing a seed pixel can be traced (and the region filled) at a later stage of the image scan.

For each defect region, size and shape parameters are calculated and presented to the user. These parameters are: maximum and minimum border contour radius from the centroid, ratio of maximum to minimum radius (*height/width*), length, area and compactness (*perimeter²/area*). Crack features have high values of *height/width* and *perimeter²/area* whereas these ratios are small for chip like features. In this analysis, a delineation has not been made between chips and cracks due to lack of a rigorous specification for the application.

5.4 DESCRIPTION OF THE PROGRAM CODE

As with the Shore hardness test (Chapter 3) the software for the quality assessment system is divided into two parts; a set of routines written in 8086 assembler code which provide a low level interface to the frame store and a C program which implements the image analysis functions. The 8086 code is the same as for the Shore test, the C program (Fig. 5.16 - 5.19) is outlined below (for a detailed description see Appendix VIII).

The main() function of the program provide two options to the user; 'x' to exit the the program or 'p' to process a sample. In the latter case a frame is captured from the video camera and thresholded using the thresh() function. The centre_of_area() function is then called to determine, roughly, the centroid of the disc.

The inner edge of the disc is then traced by calling the inner_border_follow() function which returns one of two values: TRUE if a crack from the inner to the outer edge of the disc has been found, or FALSE otherwise. In the former case control returns to the main option selection, while in the latter case the fill_hole() function is called to fill the

interior of the edge contour. A similar process is used to fill the annular region of the disc; the outer_border_follow() function is called follows by the fill_disk() function. The region parameters returned by the fill_disk() and fill_hole() functions are then used as arguments for the ellipse results function which calculates ellipse parameters using the fit_ellipse() function and presents them to the user.

Finally the chip_fill() function is called to locate any crack and chip regions within the disc. This function scans the image for pixels corresponding to chip and crack regions and once a suitable pair of seed pixels have been found the border of the defect region is traced using the chip_border_follow() function. This returns one of two values; TRUE if the border contour encircles the seed pixel or FALSE otherwise. In the former case scanning continues until another suitable pair of seed pixels are found. In the latter case the region is filled using the fill_chips() function and size and shape parameters are calculated and presented to the user. Scanning then continues until another pair of seed pixels have been found or the whole of the image has been scanned.

5.5 USING THE SYSTEM

5.5.1 SETTING UP THE HARDWARE

In order to obtain accurate results from the system care needs to be taken in the alignment of the illumination source, optics and vision camera. The first step in setting up the hardware is align the laser so that it is perpendicular to the surface on which the test sample is placed. This can be achieved by placing a mirror on the surface and adjusting the inclination of the laser so that the incident and reflected beams are

co-axial. The semi-silvered mirror can then be aligned by placing it at approximately 45 ° to the medial axis of the laser and noting the heights of the two reflected beams, introduced by the semi-silvered mirror, above the reference surface. By rotating the mirror, the heights of the spots produced by these beams can be made equal and at this point the semi-silvered mirror will be at exactly 45 ° to the laser beam axis.

The next step is to set up the confocal lenses. This can be achieved by first placing a cross hair at the centre of the laser spot as seen on the reference surface. The first lens should then be introduced and it's position and angle adjusted so that the laser spot is both circular and centred on the cross hair. The distance between the two lenses should then be adjusted so that the expanded laser beam is collimated. This can be verified by moving a screen backwards and forwards along the beam axis and noting any change in spot size.

Finally, a component should be placed under the laser beam and the vision camera positioned and focused to give a clear image of the sample on the reference monitor. Ideally the sample should occupy as much of the field of view as is possible.

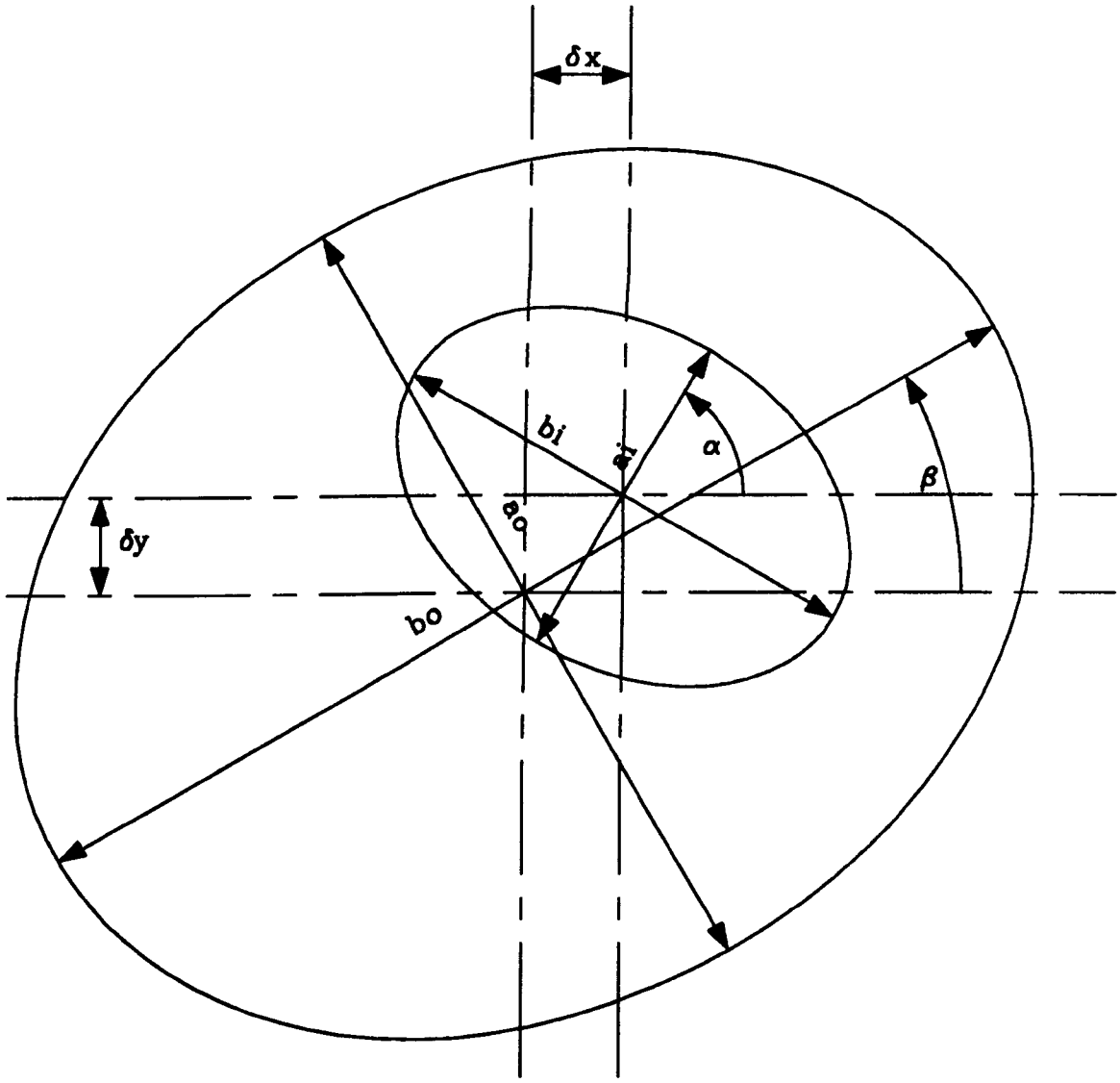
5.5.2 RUNNING THE SOFTWARE

The analysis program is entirely menu driven and is invoked by the operator entering

assess <enter>

The user is then faced with two options; 'p' to process a sample or 'x' to exit the system. During the processing of a sample no further input is required from the user. The results of the analysis are presented to the user on the VDU as execution of the program proceeds. Ellipse parameters are given first; these consist of major and minor axis length

and major axis rotation for both the inner and outer edges, offset between ellipse centres and percentage of the annular region occupied by defects. Edge profiles for the inner and outer edges are then presented in graphical form showing the deviation of the edge from the approximating ellipse, in pixels, against angle of rotation. Finally, a summary of chip and crack defects is given in tabular form. For each defect the following parameters are calculated: position of centroid, area, height, width, ratio of height to width and compactness ($perimeter^2/area$).



($\delta x, \delta y$) offset between centres

(ai,bi) inner edge eccentricity

(ao,bo) outer edge eccentricity

α inner edge rotation

β outer edge rotation

Figure 5.1: SUMMARY OF GEOMETRIC DEFECTS

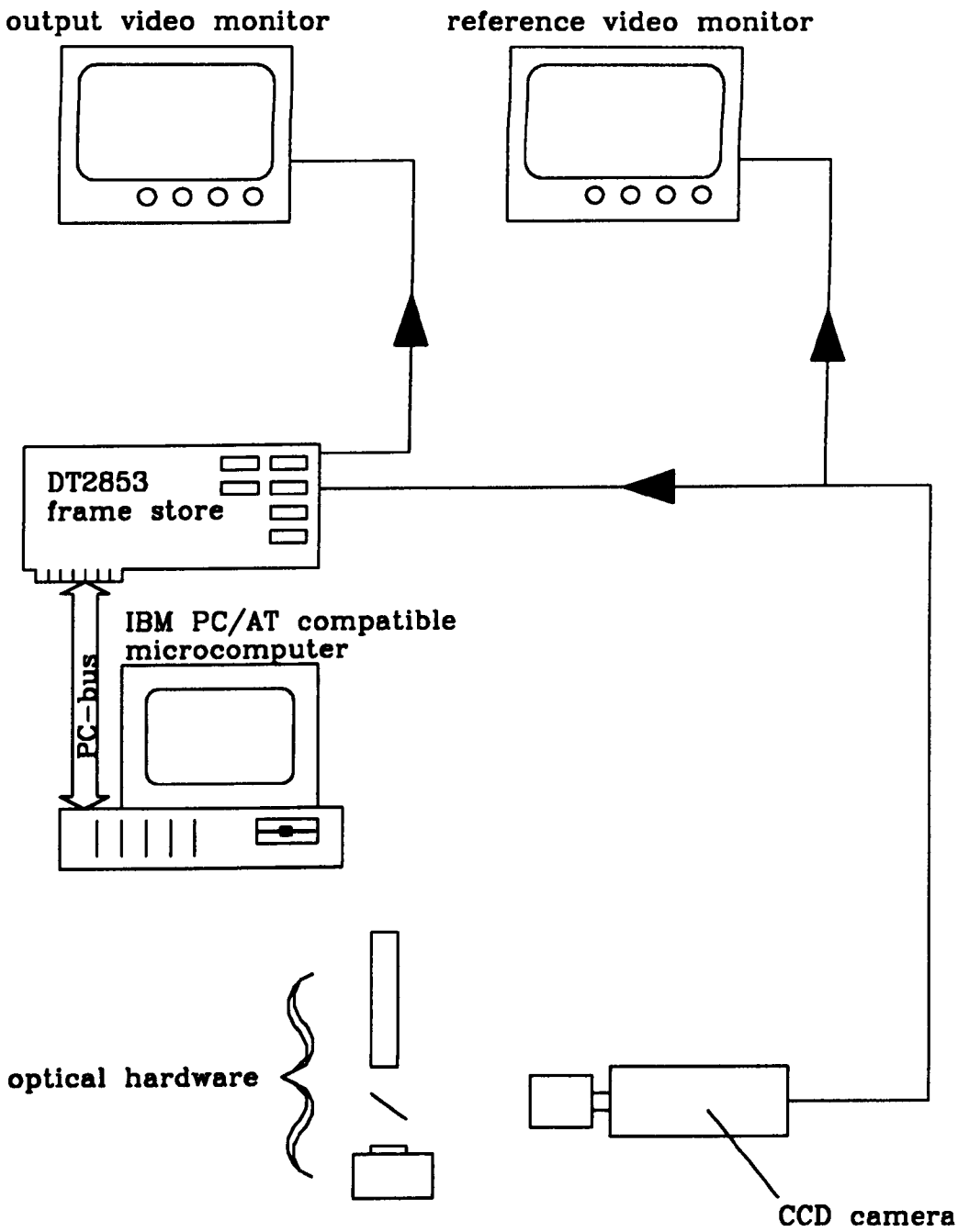


Figure 5.2: VISION SYSTEM HARDWARE

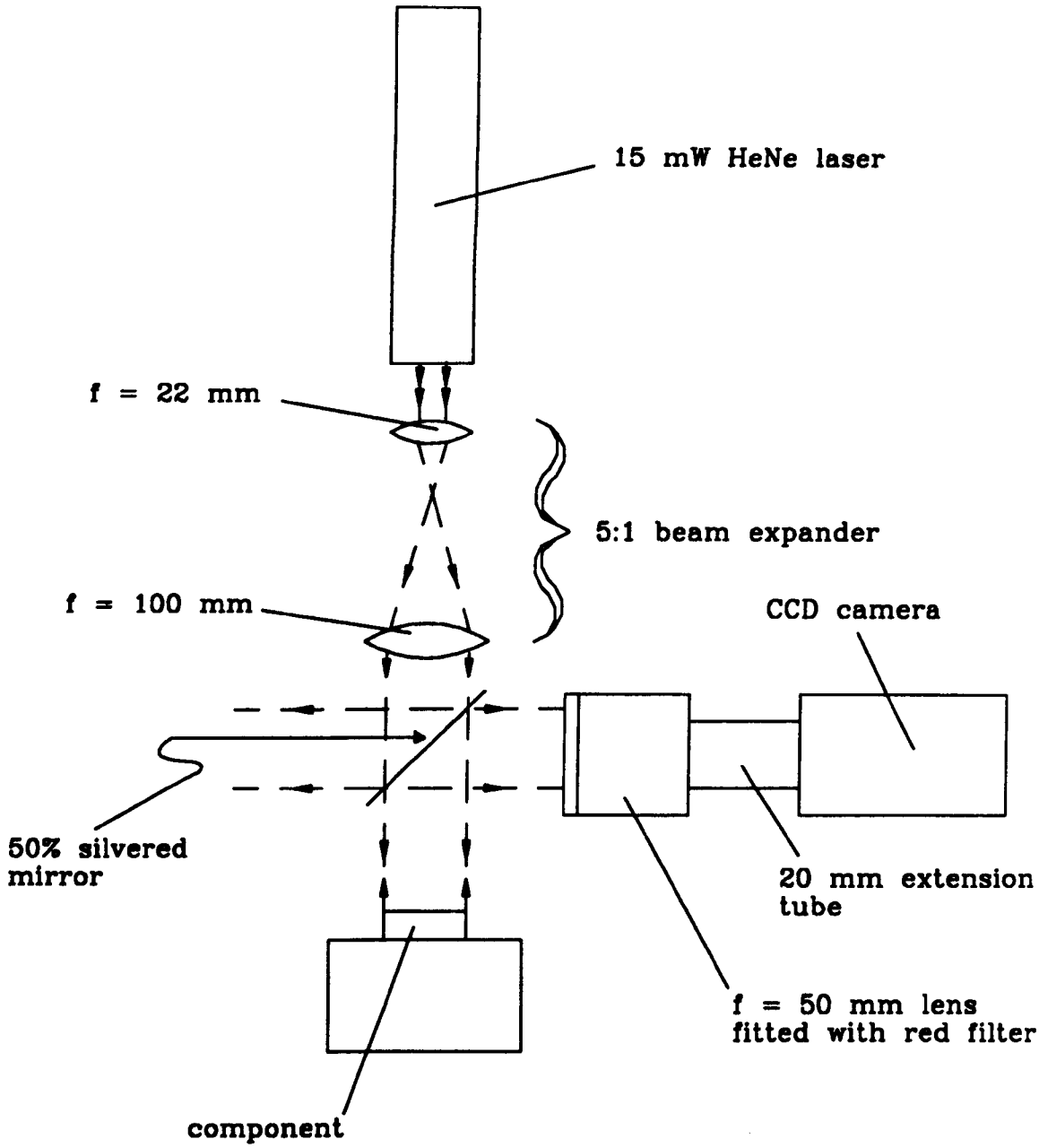


Figure 5.3: OPTICAL HARDWARE

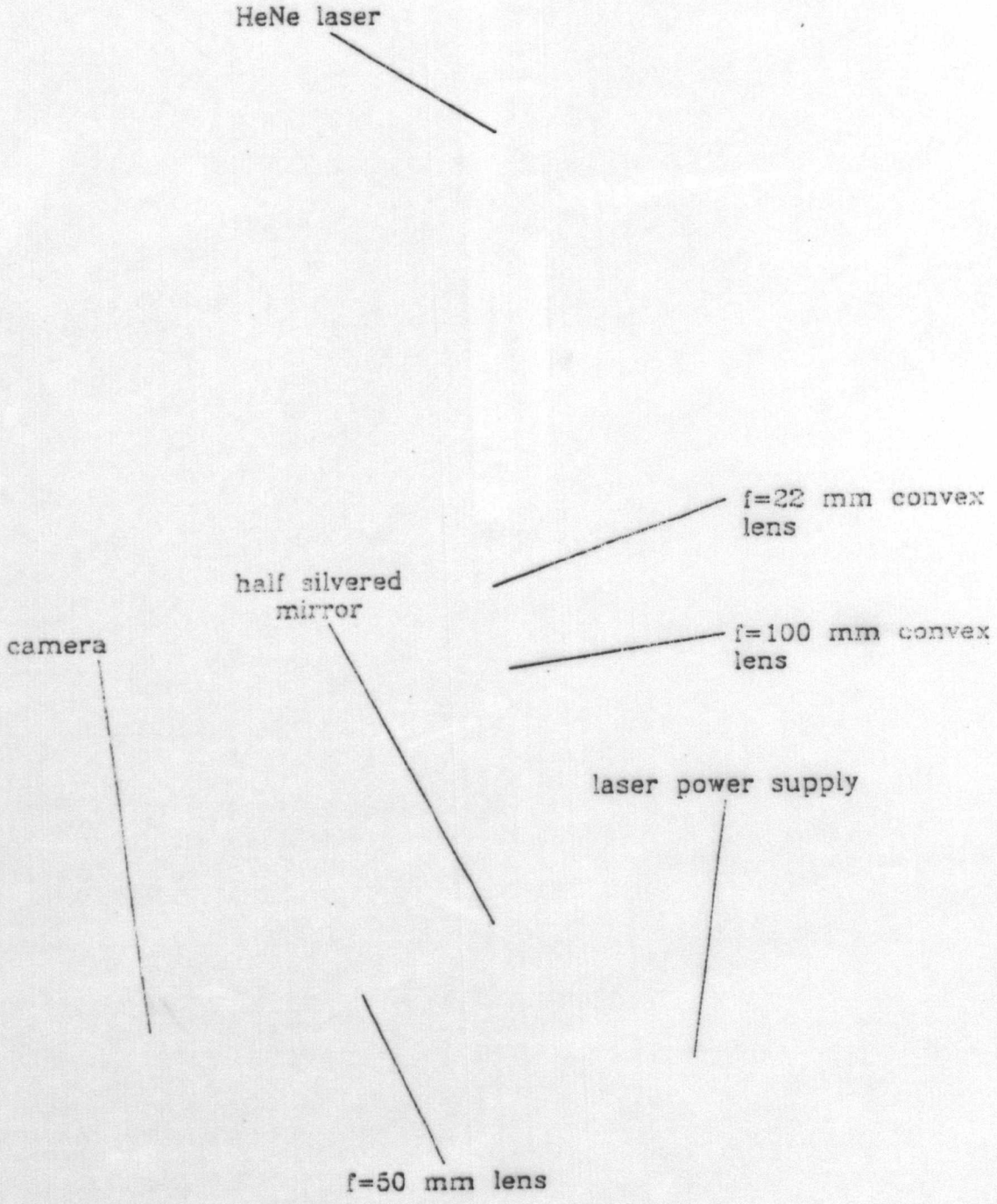


PLATE III: OPTICAL HARDWARE

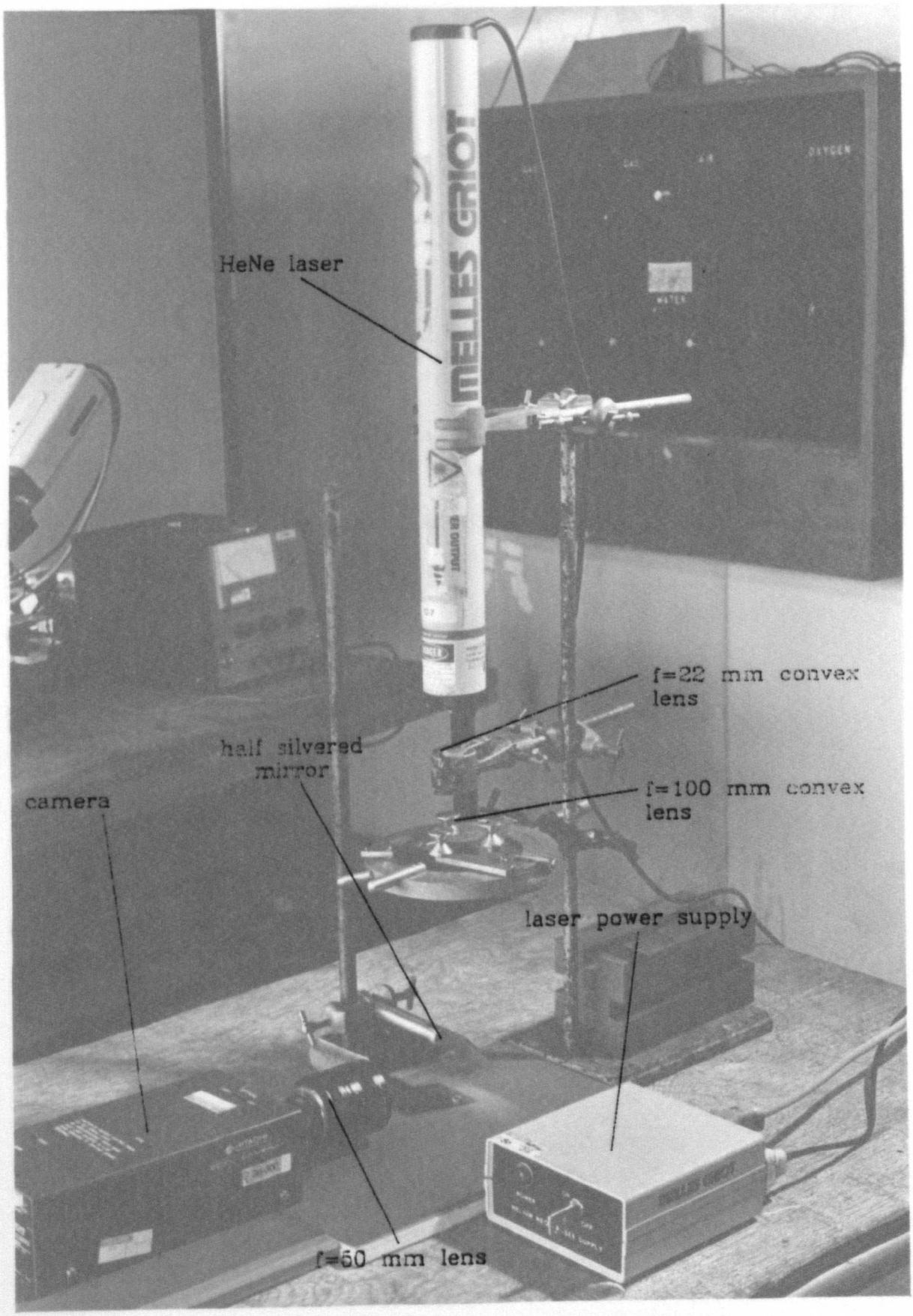
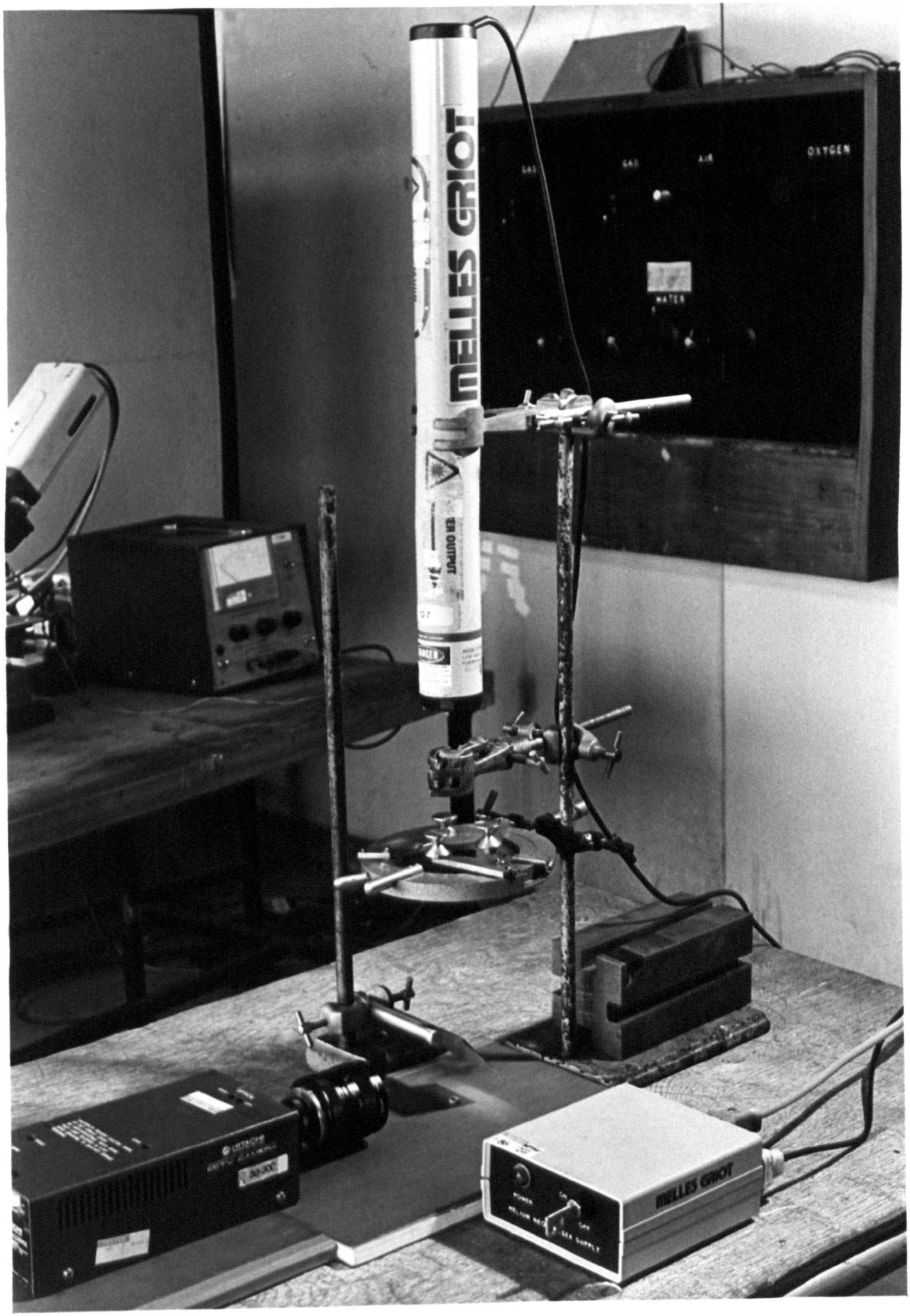


PLATE III: OPTICAL HARDWARE


MELLES GRIOT
OUTPUT
MELLES GRIOT

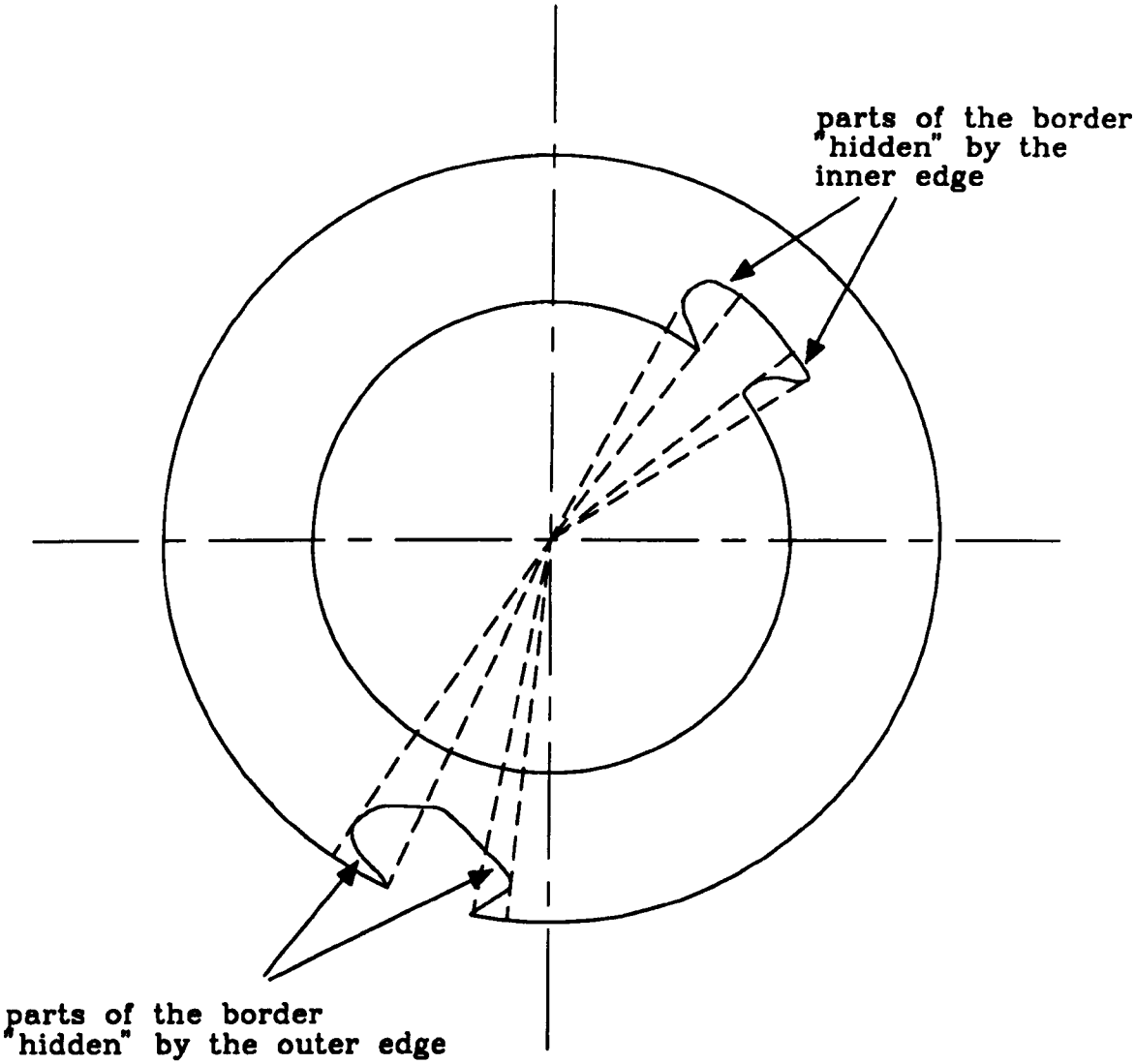
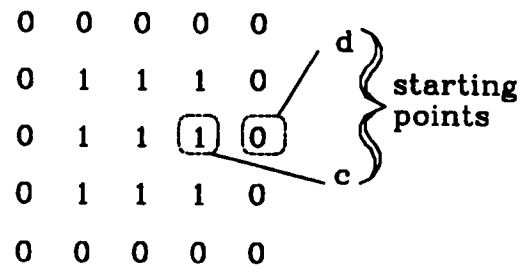
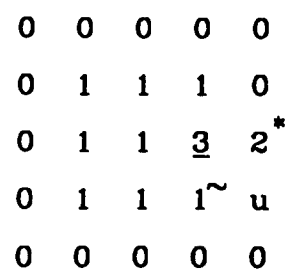


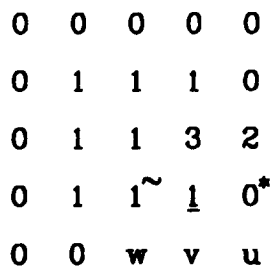
Figure 5.4: EDGE PROFILE PROBLEMS



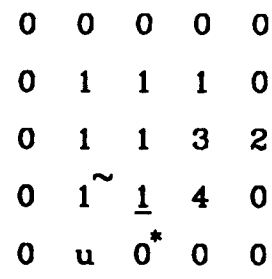
(1)



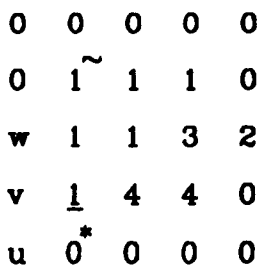
(2)



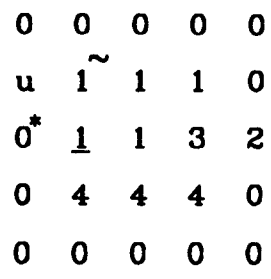
(3)



(4)



(5)



(6)

Situation at successive entries into step (2) of BF algorithm. Current c is underlined, e_1 is starred (*) and e_k is marked with a tilde (~); e_2,...., e_{k-1} are marked u, v, w...

Pixels having grey level 0 form connected region D, pixels having grey level 1 form connected region C.

Figure 5.5: APPLICATION OF THE BF ALGORITHM

(continued)

0	u	v	0	0
0^*	$\underline{1}$	$1^{\sim}$	1	0
0	4	1	3	0
0	4	4	4	0
0	0	0	0	0

(7)

0	0	0^*	u	0
0	4	$\underline{1}$	$1^{\sim}$	0
0	4	1	3	2
0	4	4	4	0
0	0	0	0	0

(8)

0	0	0	0^*	u
0	4	4	$\underline{1}$	v
0	4	1	$3^{\sim}$	2
0	4	4	4	0
0	0	0	0	0

(9)

0	0	0	0	0
0	4	4	4	0
0	4	1	$\underline{3}$	2^*
0	4	4	$4^{\sim}$	u
0	0	0	0	0

(10)

0	0	0	0	0
0	4	4	4	0
0	4	1	4	0
0	4	4	4	0
0	0	0	0	0

(11)

Situation at successive entries into step (2) of BF algorithm. Current c is underlined, e_1 is starred (*) and e_k is marked with a tilde (~); e_2,..., e_{k-1} are marked u, v, w...

Figure 5.6: APPLICATION OF THE BF ALGORITHM

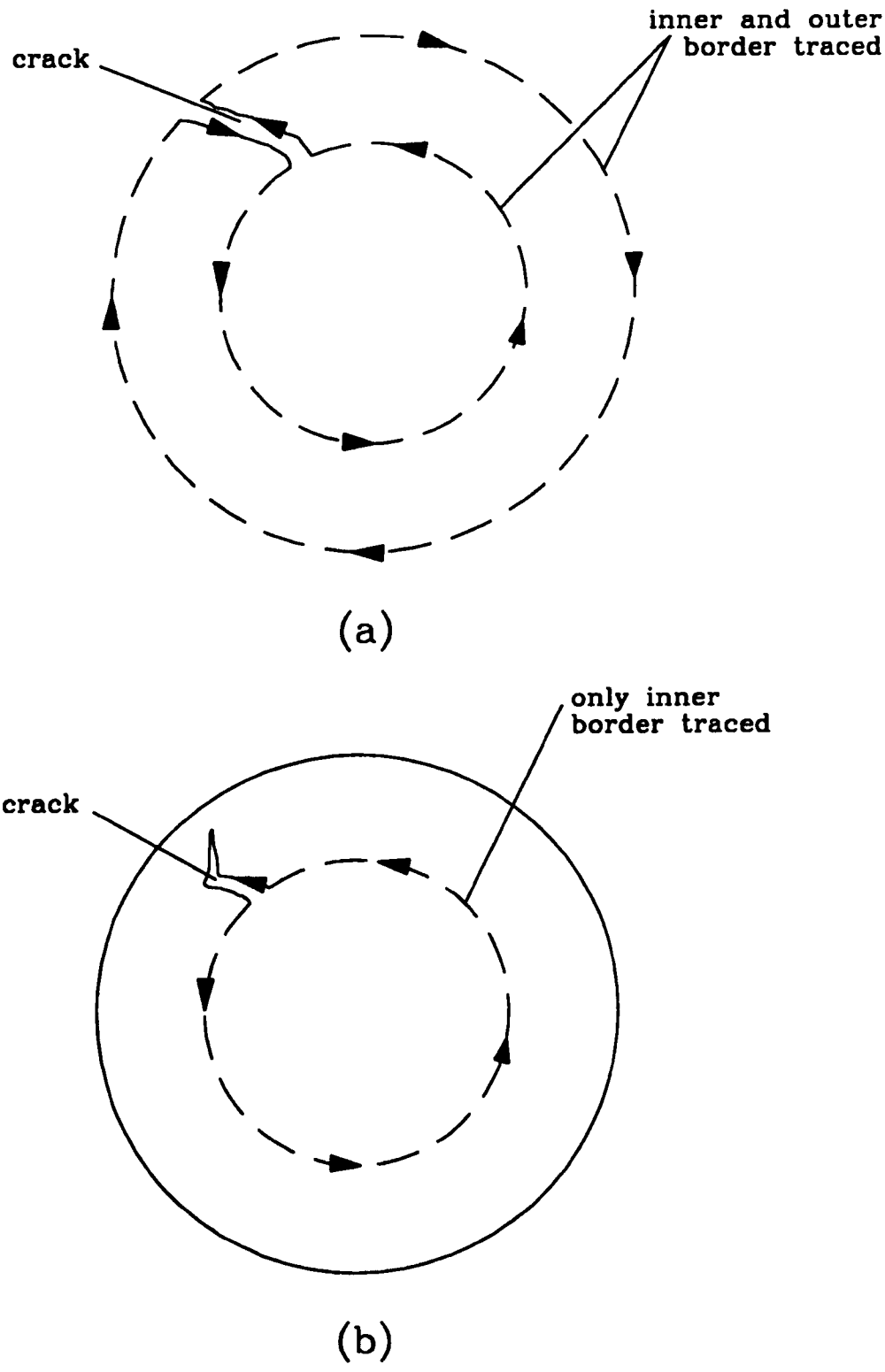
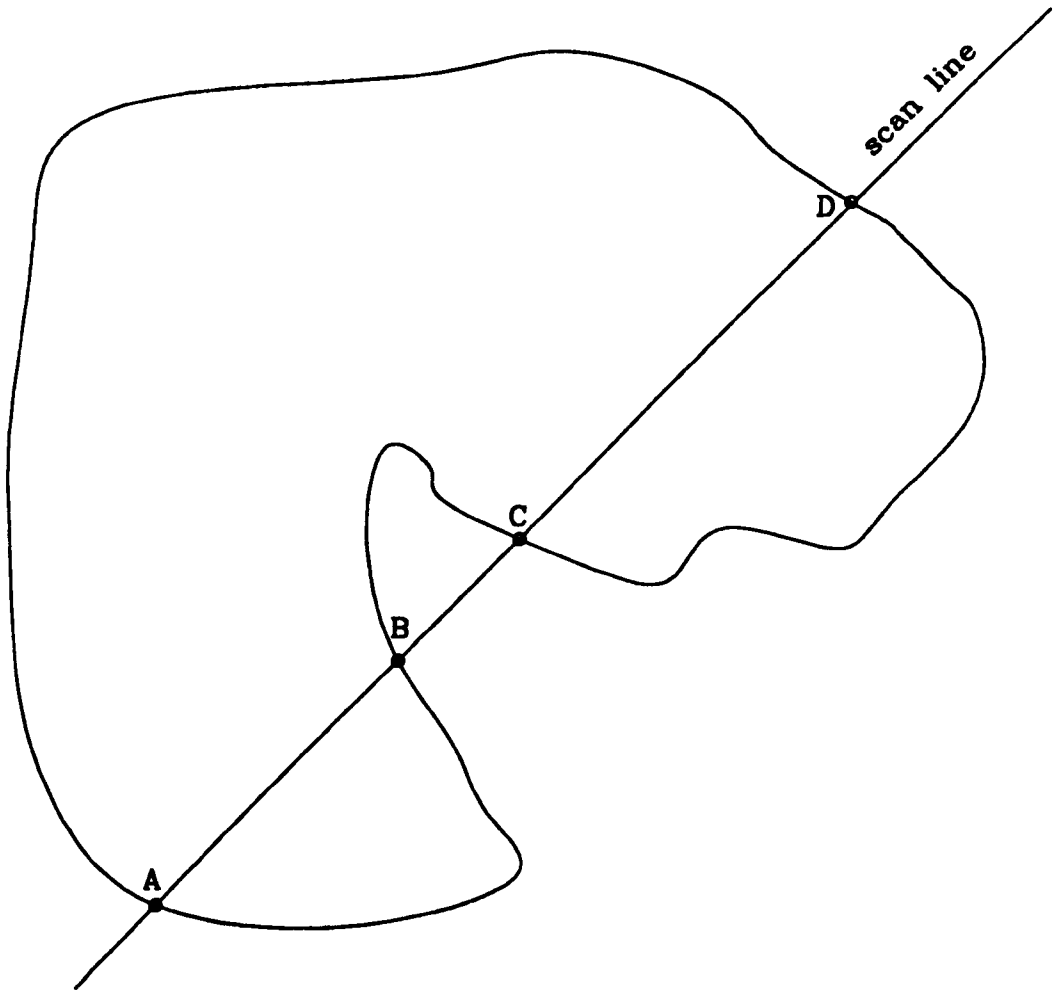


Figure 5.7: RESULTS OF BF ALGORITHM

(a) disc with crack extending from outer to inner edge, (b) disc with internal crack



**Filling begins at A and stops at B;
filling then continues at C and finishes at D.**

Figure 5.8: PARITY FILLING OF A CONTOUR

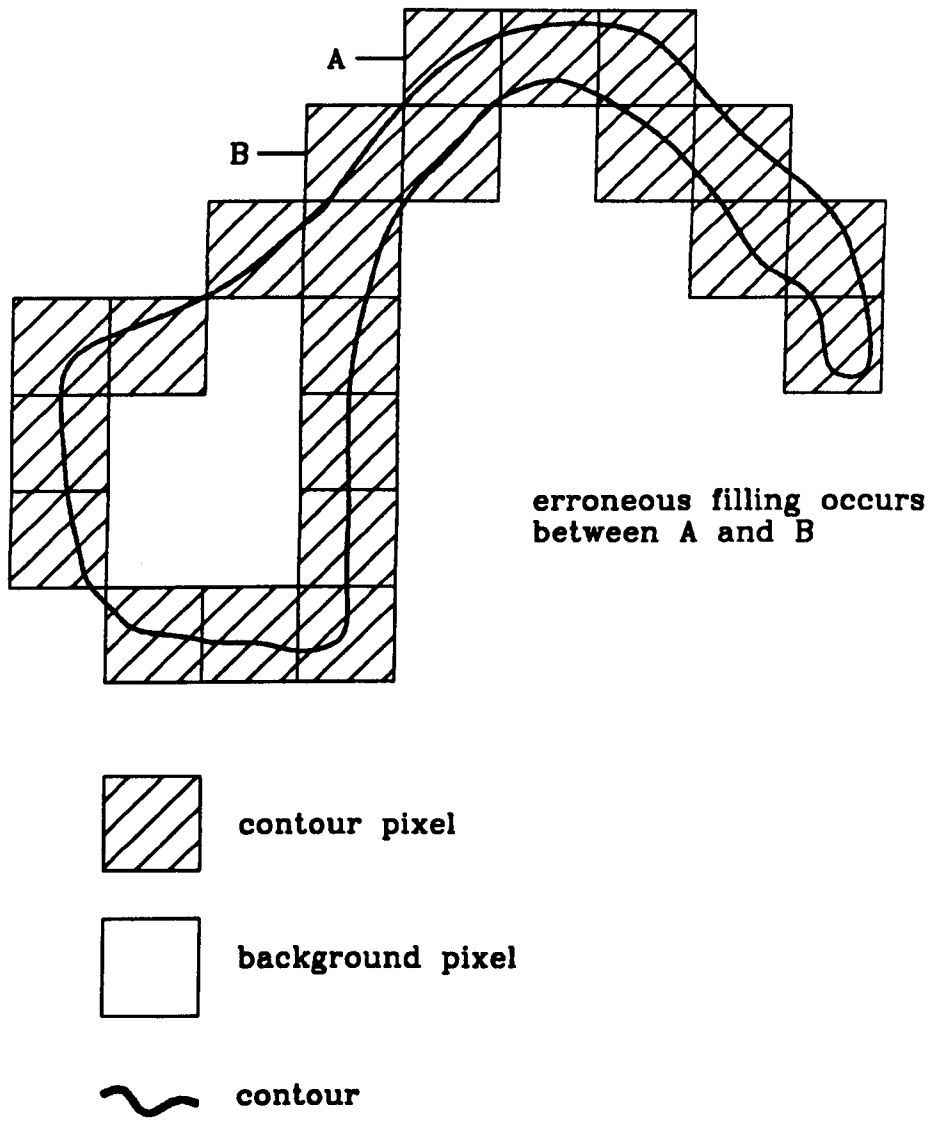
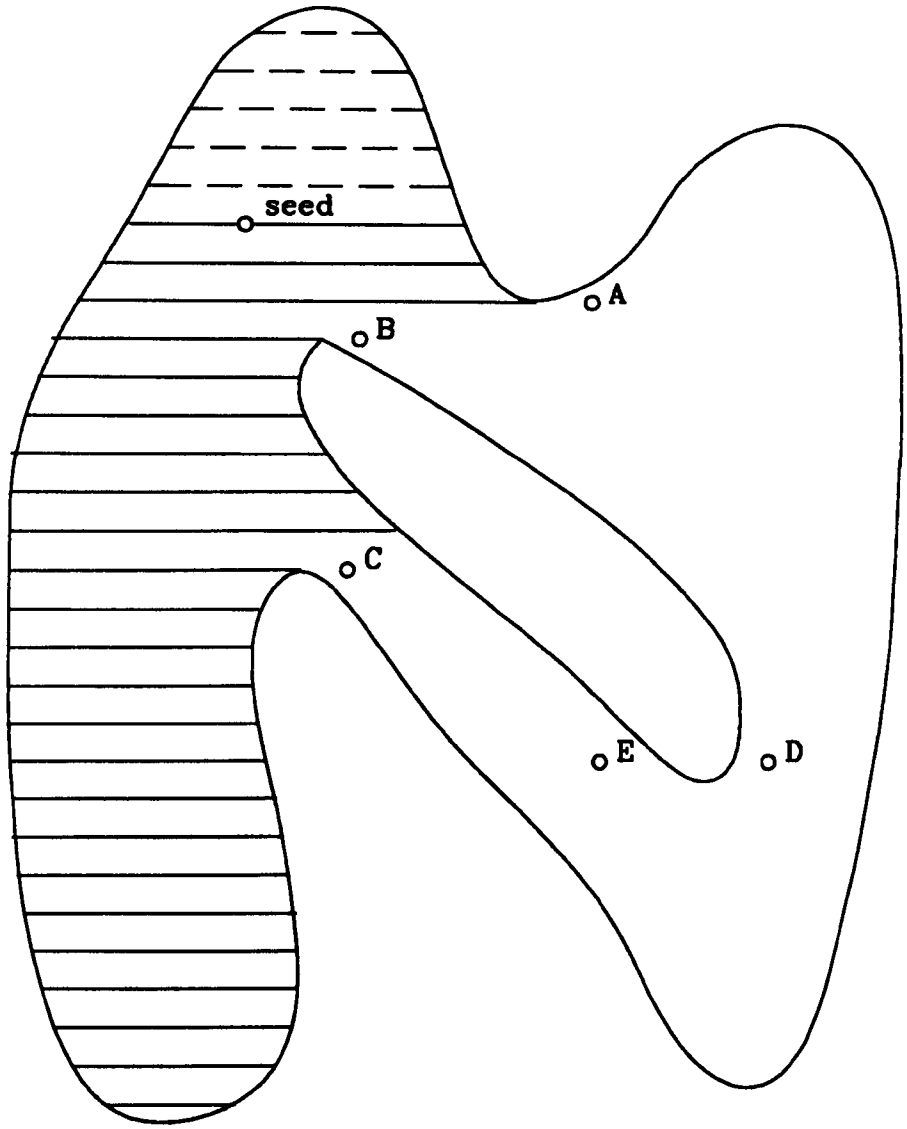
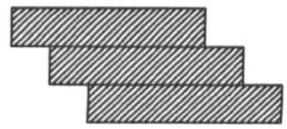


Figure 5.9: CONTOUR WHICH CANNOT BE FILLED PROPERLY BY PARITY FILLING



Filling proceeds downwards from the seed pixel (shown by ————) and then in the upward direction (shown by — — —). Pixels A,B,C are placed on the stack, in order, during the filling of the shaded part. Pixels D and E are placed on the stack, later, during filling of the unshaded part.

Figure 5.10: CONNECTIVITY FILLING FOR A NON CONVEX REGION



simple arc: above=1, below=1

local minimum: above=2, below=0

local maximum: above=0, below=2

impossible condition for a "full" region:
above=2, below=1

Figure 5.11: EXAMPLE INTERVAL OVERLAPS

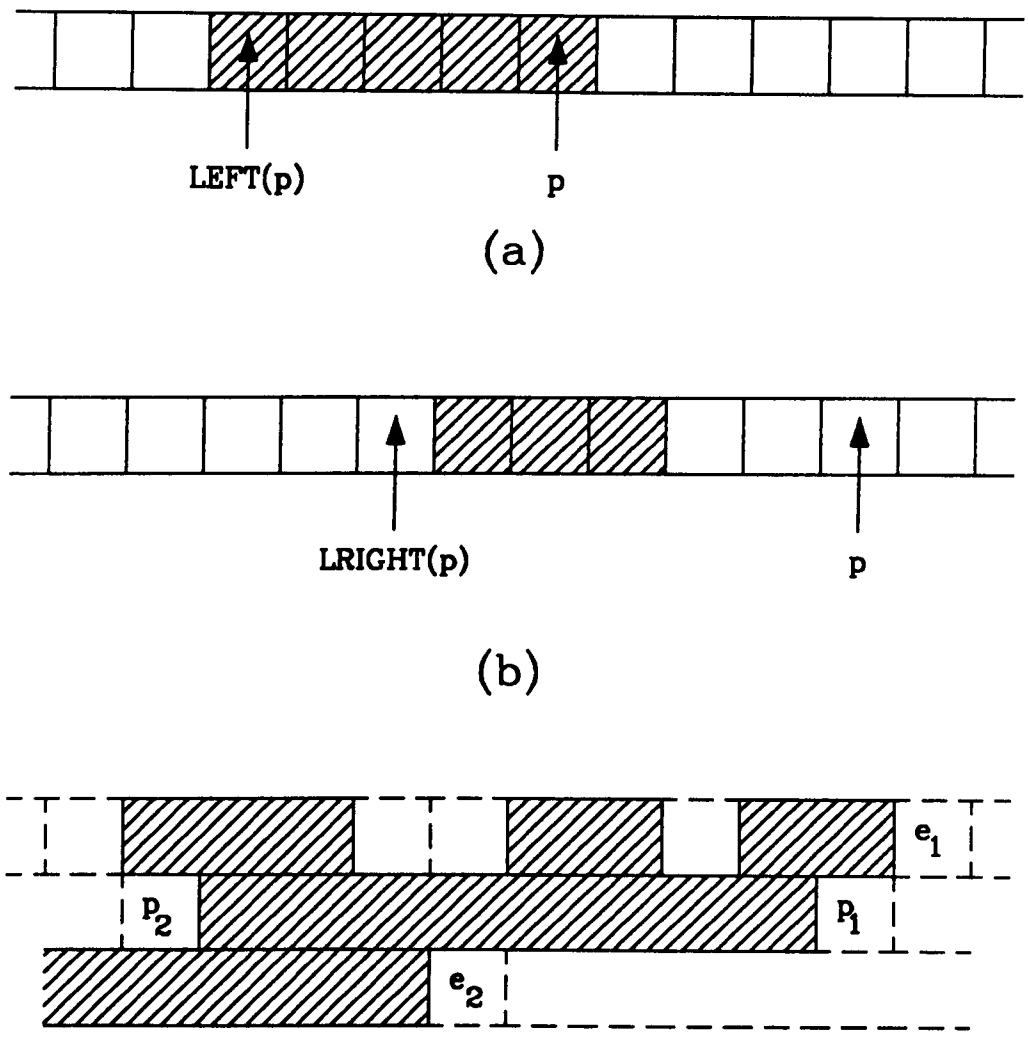


Figure 5.12 PIXEL DEFINITIONS

(a) definition of LEFT(p), definition of LRIGHT(p),
(c) definition of p1, p2, e1 and e2.

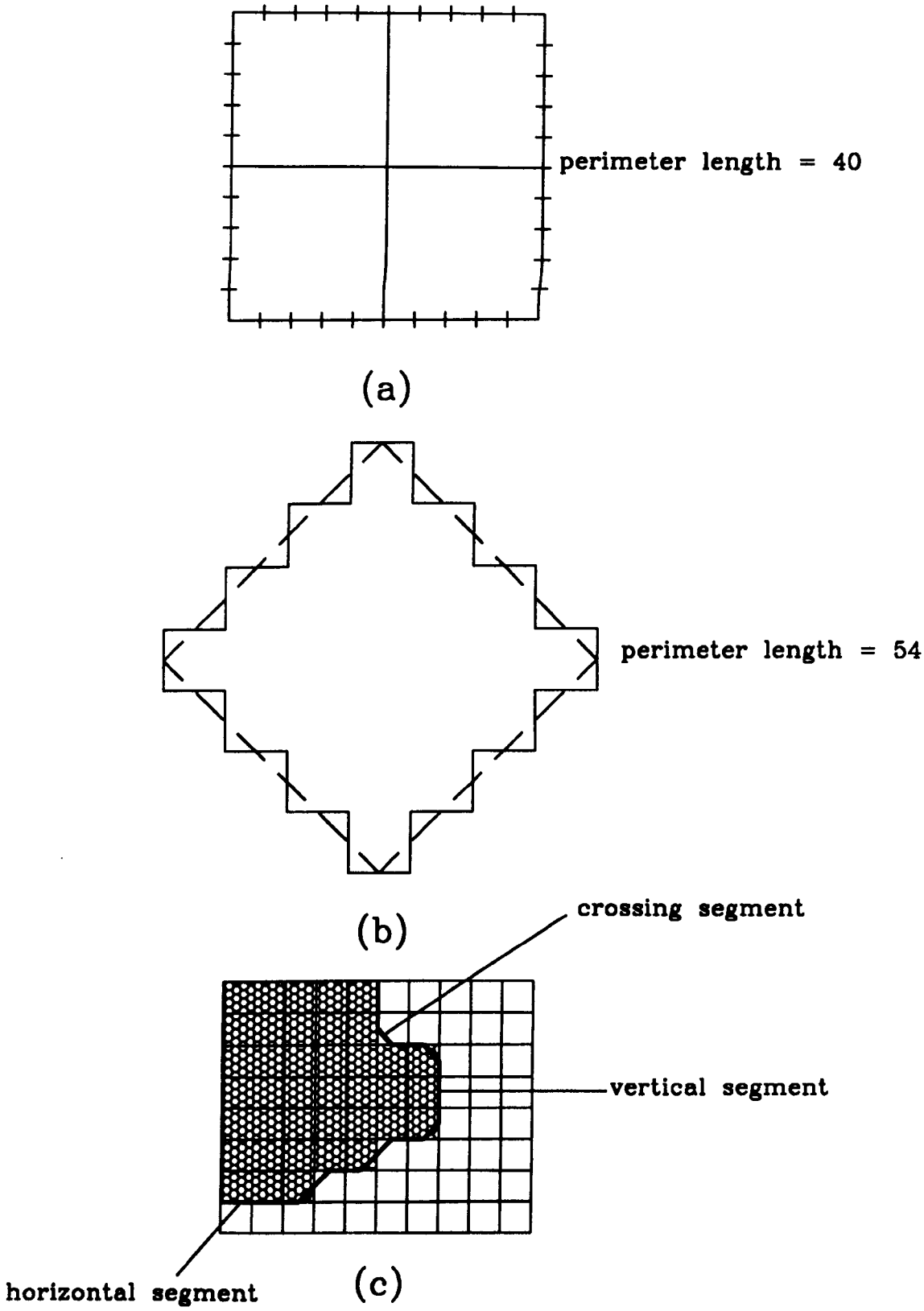
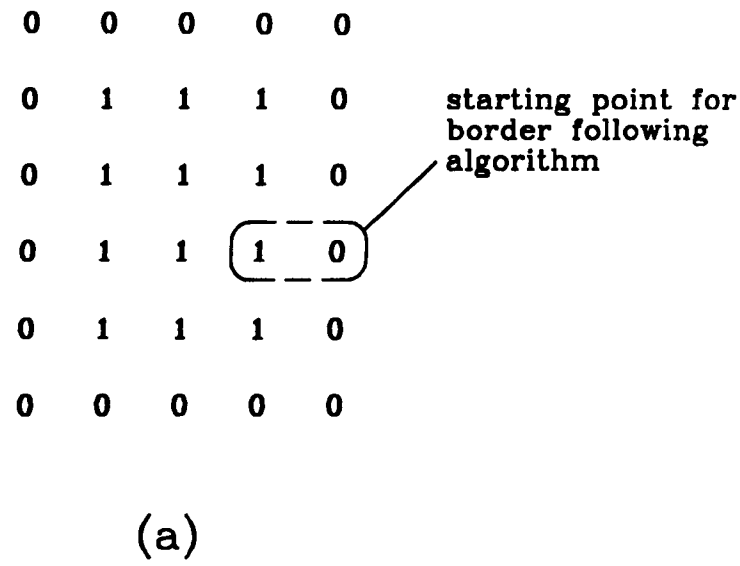
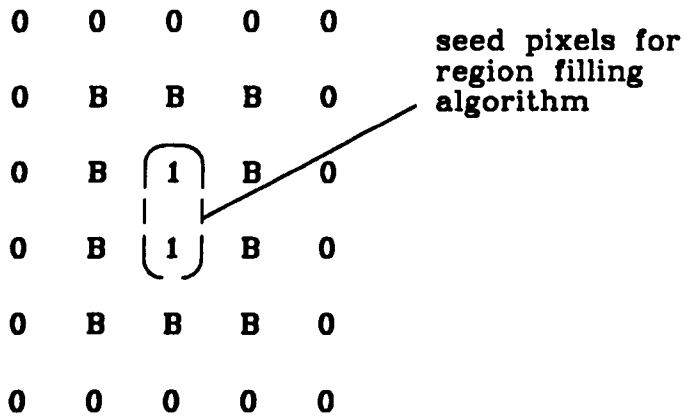


Figure 5.13: PERIMETER LENGTH CALCULATION

(a) perimeter of square, (b) perimeter of square rotated 45°, (c) reduction in errors by corner cutting.



(a)

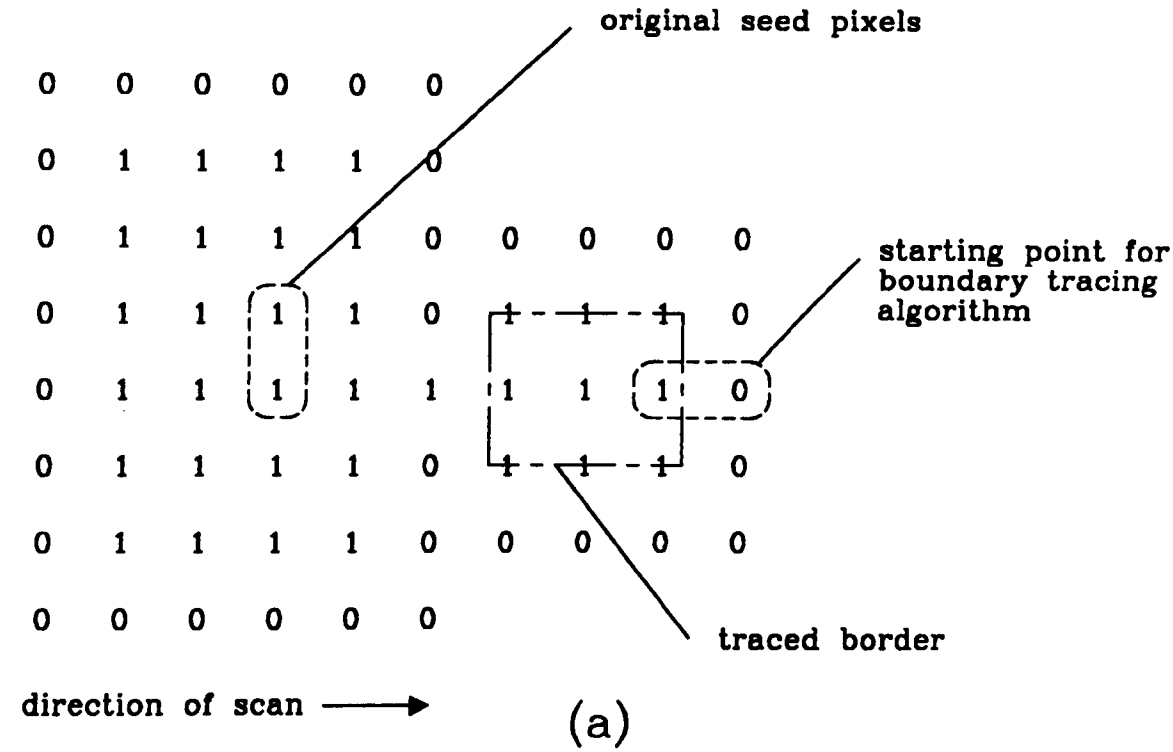


(b)

0 = background pixel
1 = object pixel
B = border pixel

Figure 5.14: MINIMUM SIZE REQUIREMENTS

(a) region before border tracing, (b) after border tracing



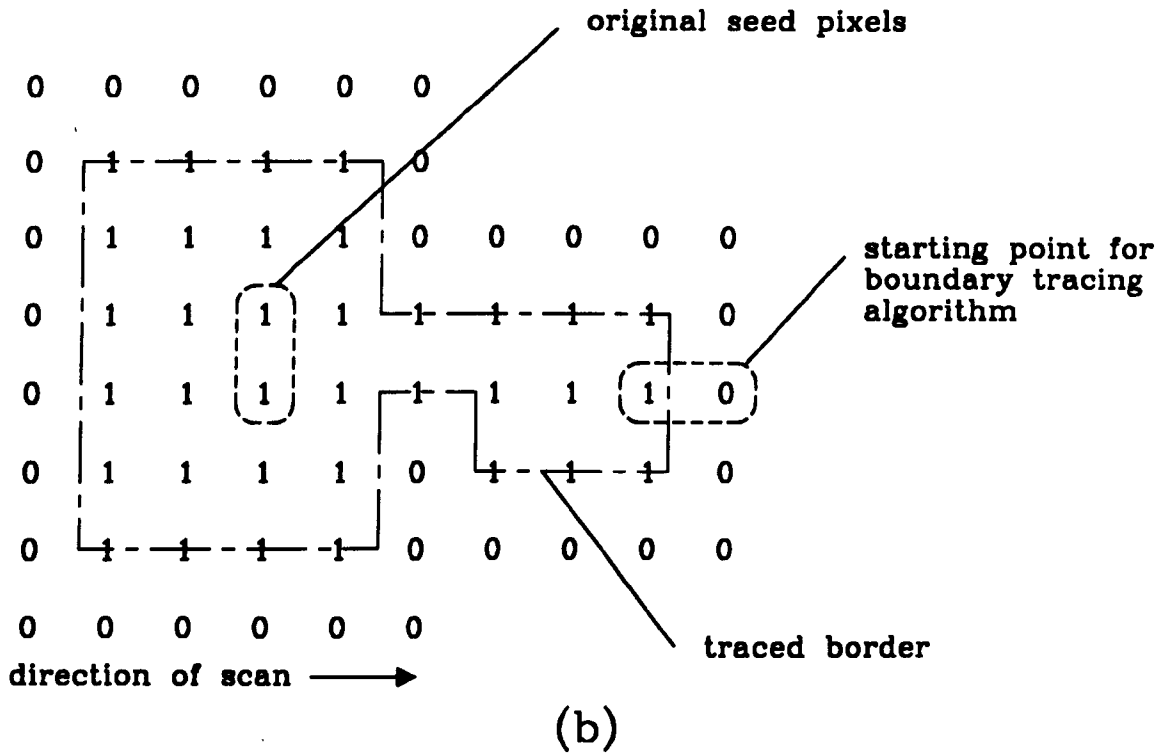


Figure 5.15: BORDER TRACING PROBLEMS

(a) regions connected by a single pixel, (b) regions connected by a pair of pixels

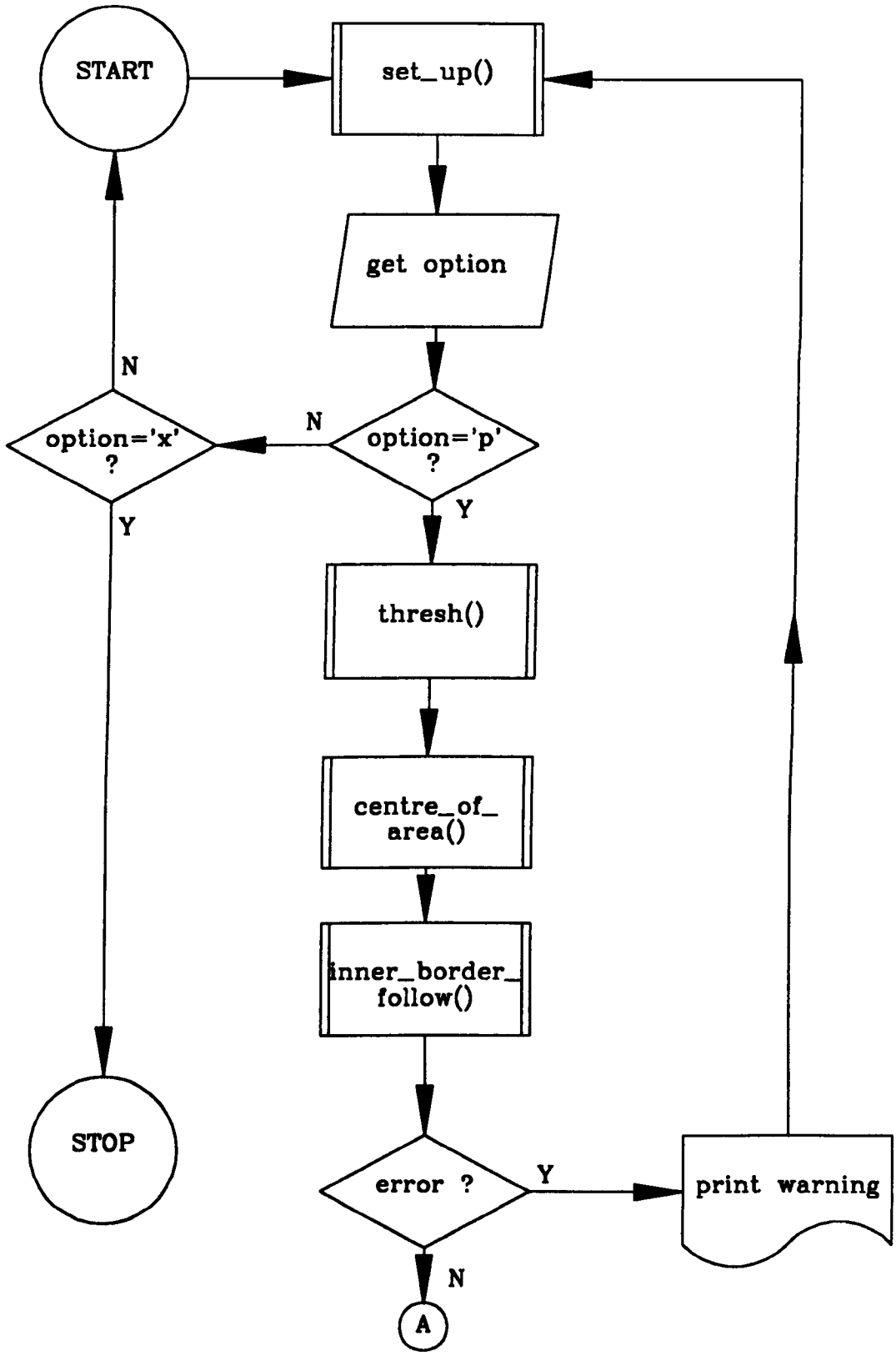


Figure 5.16: FLOWCHART FOR THE QUALITY ASSESSMENT PROGRAM / cont'd

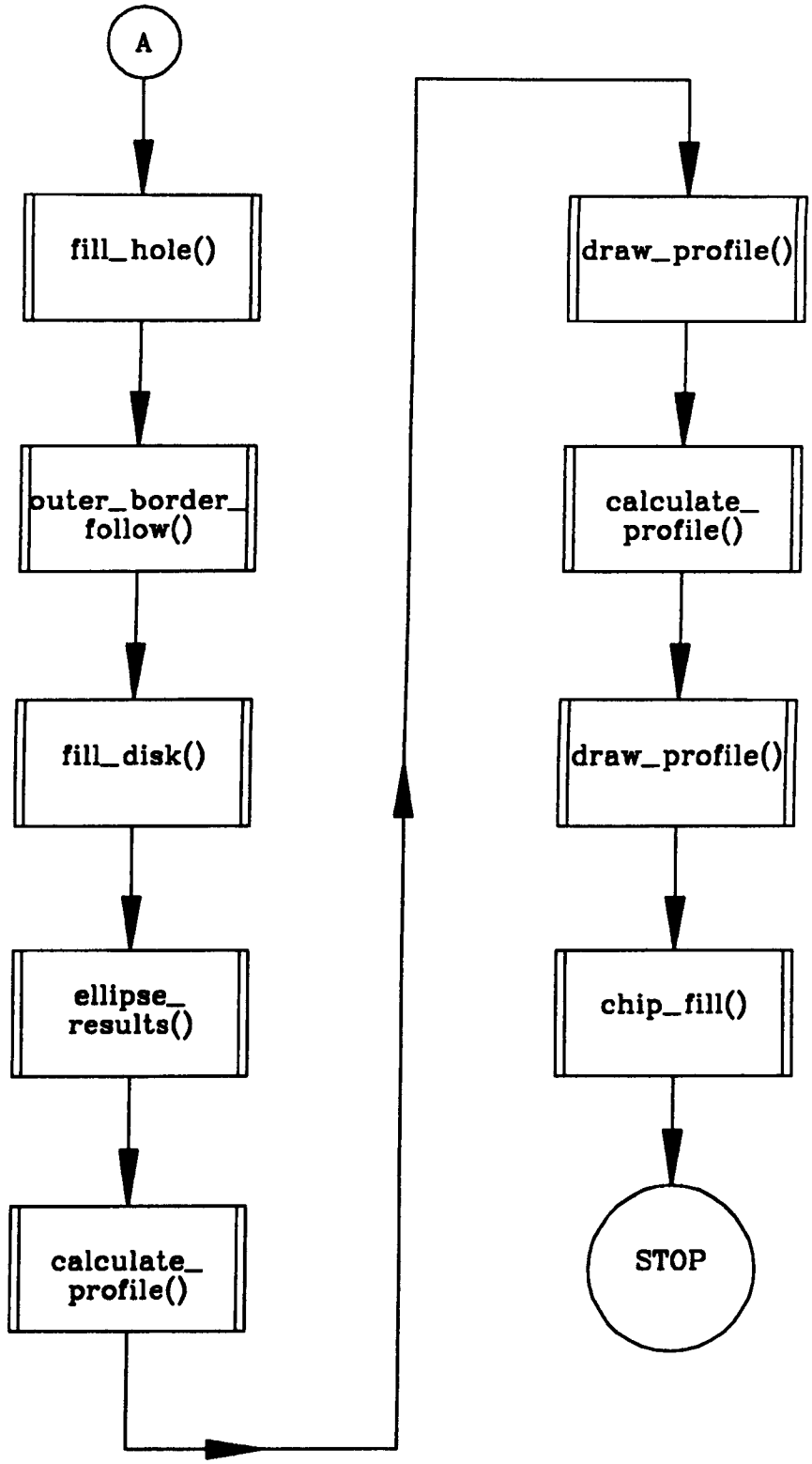


Figure 5.17: FLOWCHART FOR THE QUALITY ASSESSMENT PROGRAM / cont'd

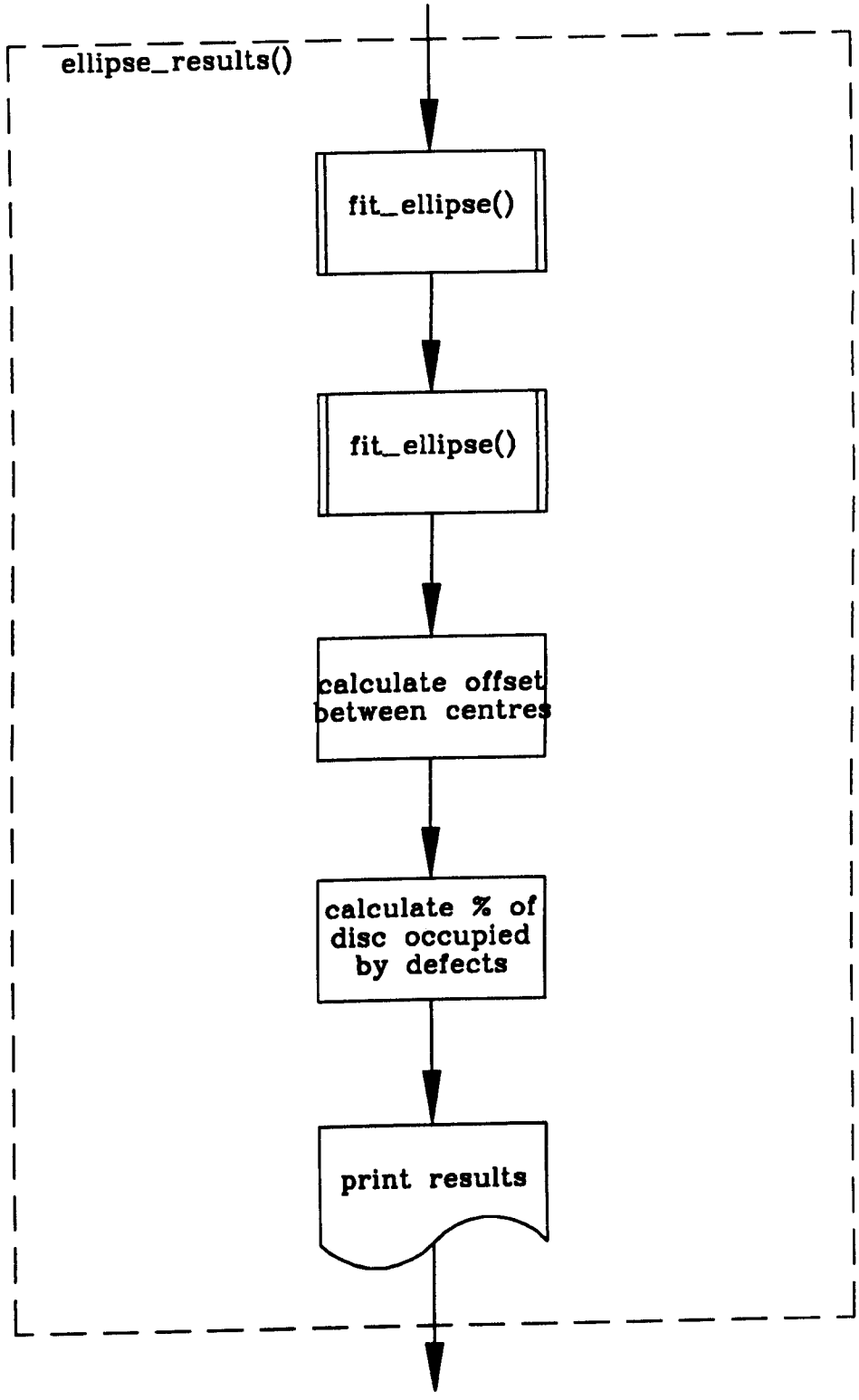


Figure 5.18: FLOWCHART FOR THE QUALITY ASSESSMENT PROGRAM / cont'd

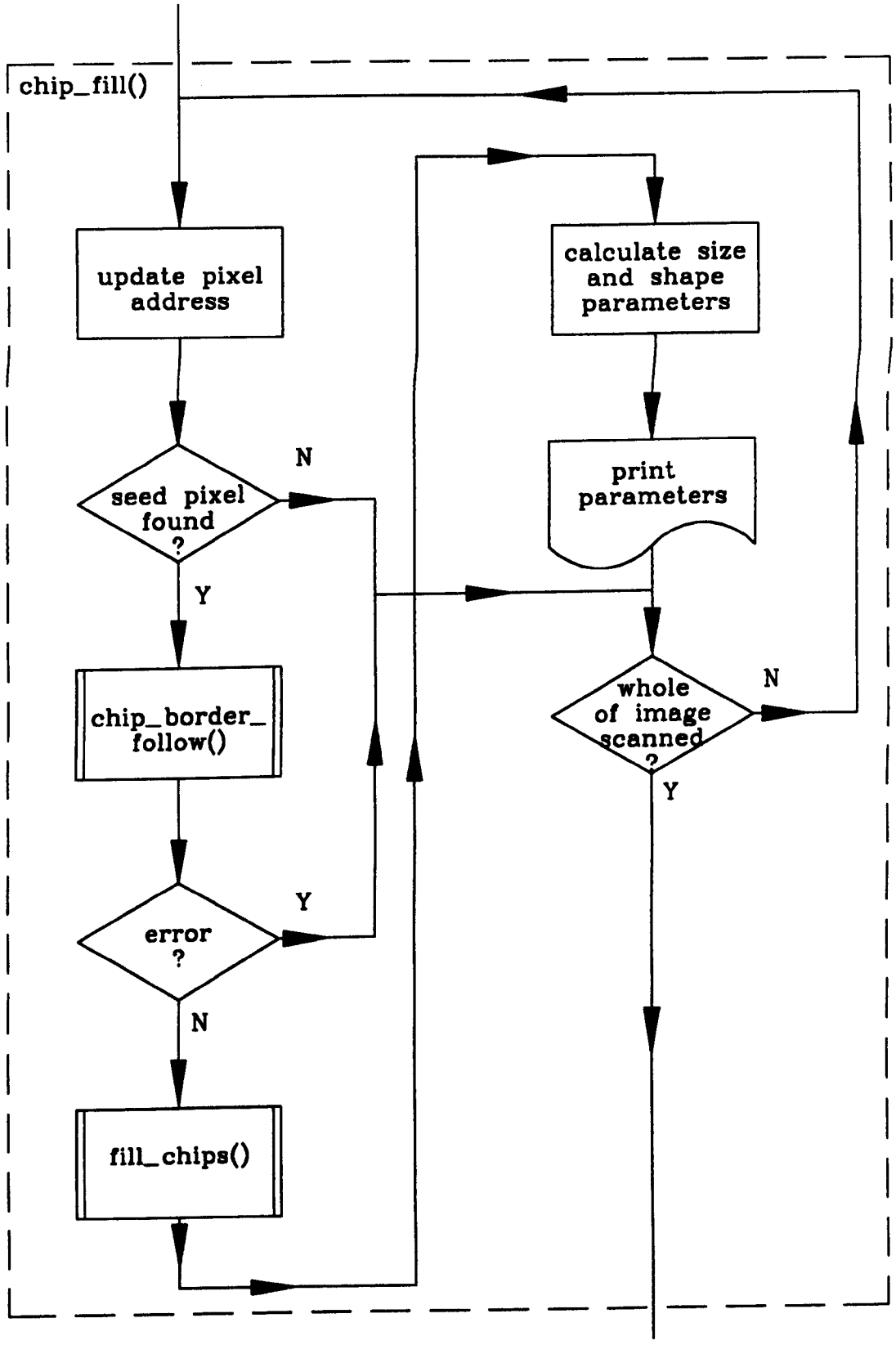


Figure 5.19: FLOWCHART FOR THE QUALITY ASSESSMENT PROGRAM

CHAPTER 6

RESULTS

6.1 INTRODUCTION

The previous chapters of this work have been concerned with the theoretical and experimental development of three vision based NDT systems. In this chapter typical results obtained from the systems are presented since space limitations prevent a comprehensive survey.

In this chapter, the visual results produced by the systems have in some cases been recorded using photography, however use has also be made of a laser printer. In the latter case a special program was written to convert frame store image data into PostScript [55] format, suitable for document production using an Apple ImageWriter [56] laser printer. The poor contrast and resolution of the laser printer output in comparison to photography necessitated remapping of the grey scale of the original frame store image data in order to produce satisfactory results.

Another problem in presenting results was that no facility existed for the production of hard copy of the graphical output produced by some of the vision systems. In this case a purpose written program was used to convert graphics data into AutoLisp [57] format,

suitable for input to the AutoCAD [57] computer-aided design package. The graphical output was then plotted using the AutoCAD package.

6.2 SHORE HARDNESS TEST

A typical indenter path record produced by the vision system for the Shore hardness test is shown in Plate IV. Here time is increasing from the right of the image to the left. The distance from the top of the photograph to the dotted line on the left represents 25 cm. Prior to the indenter entering the guide tube, a horizontal line is produced by reflection of the laser light from the base of the test rig. The line momentarily disappears when the indenter enters the guide tube and the parabolic motion (displacement with respect to time) of the indenter becomes apparent. Finally a thick horizontal line is produced indicating where the indenter has come to rest.

In order to calibrate the system, tests were carried out on five standard hardness blocks. The hardness of the first two blocks was measured using the a standard Vickers test machine, since a commercial Shore Scleroscope was not available, and gave values of 219 HV30 (= 31 Shore) and 340 HV30 (= 46 Shore). The hardness of the other blocks was measured using a standard Rockwell test machine giving values of 25 Rockwell C (= 38 Shore), 42 Rockwell C (= 55 Shore) and 61 Rockwell C (= 85 Shore). Several measurements were taken at different parts of each block (to prevent local work hardening) using the vision system and the results are presented in Tables 6.1 and 6.2. The values in the Number of Samples column indicate the number of measurements which fall within a stated range (given in the Error Range column) of the mean hardness of the batch of measurements.

	31 Shore		38 Shore		46 Shore	
Error Range (%)	**Number of Samples**	**% of Batch**	**Number of Samples**	**% of Batch**	**Number of Samples**	**% of Batch**
±2	1	27	4	13	1	11
±4	4	33	8	27	4	50
±6	6	40	14	47	5	55
±8	11	53	23	77	7	77
±10	18	60	25	83	9	100
±15	27	93	26	87	-	-
±20	30	100	30	100	-	-
Mean Shore Number	47		55		62	
Standard Deviation (σ)	6		7		14	
Confidence Interval	47 ± 2		55 ± 3		62 ± 11	

Table 6.1: Shore Hardness Test Results

	55 Shore		85 Shore	
Error Range (%)	**Number of Samples**	**% of Batch**	**Number of Samples**	**% of Batch**
±2	4	13	2	13
±4	7	23	4	27
±6	11	36	6	40
±8	16	53	10	67
±10	20	67	12	80
±15	29	97	15	100
±20	30	100	-	-
Mean Shore Number	72		78	
Standard Deviation (σ)	14		6	
Confidence Interval	72 ± 5		78 ± 3	

Table 6.2: Shore Hardness Test Results

A plot of mean Shore number measured by the system against Shore number measured by the Rockwell and Vickers tests, for the five hardness standards, is shown in Fig. 6.1. Here the error bars indicate a range of $\pm 1\ \sigma$ The entries in the confidence interval row indicate the expected variation in the population mean for a 95% certainty, as determined by the z test.

It can be stated that if 30 measurements are taken using the system then the measured hardness will be within ± 5 Shore of the true value, with a 95% certainty. This range corresponds to a variation of $\pm$ 8 mm in the height calculation and is significantly greater than the errors due to limited spatial resolution ($\pm$ 0.5 mm) and sampling frequency ($\pm$ 0.5 mm), this may be due to blooming variations caused by the video camera a.g.c.

It can be seen that points having the four lowest hardness values fall roughly on straight line indicating a linear relationship between measured hardness and actual hardness. It can also be seen that the measured hardness numbers are always greater than the actual hardness numbers; this is to be expected since the test rig used is only an approximation to a Shore Scleroscope. In particular, the sloping branch, through which the indenter is introduced into the guide tube, increases the effective drop height by around 20 mm. This produces a corresponding increase in the indenter impact velocity and therefore the rebound height so that the Shore number is also increased. In the light of these points the Shore hardness equation:-

$$\text{Shore number} = \frac{\text{rebound height in mm}}{250} \times 140 \qquad [6.1]$$

may be modified to suit the vision system to:

$$\text{Shore number} = \left[\frac{1}{1.12} \times \frac{\text{rebound height in mm}}{250} \times 140 \right] - 10.7 \qquad [6.2]$$

where the scaling factor (= 1.12) and offset (= 10.7) are determined from the calibration graph shown in Fig. 6.2.

Clearly, the result obtained from the 85 Shore block is erroneous; however, as indicated in Chapter 2, at high hardness values sample deformation in the Shore test is predominantly elastic whilst deformation in the Rockwell and Vickers tests is still predominantly plastic. This means that there is a departure from linearity in the Shore test for Shore number greater than about 70. A large spread in the hardness values for any given block is also apparent; this may attributed to variations in angle of impact, windage and friction etc. Both of these observations are in keeping with calibration data originally given by Shore [16].

6.3 VICKERS HARDNESS TEST

A typical result of applying the Vickers hardness test to a material is shown in Plate V. The large dark area in the centre of the image is the indentation and the background is the surface of the sample, which in this case is slightly blemished. The grey level histogram for the image (Fig. 6.2) indicates strong contrast between the object and background allowing the image to be effectively segmented, using a fixed threshold, by the analysis program. The result of this operation, together with subsequent processing steps, is shown in Plate VI. Here, the two white cross hairs mark the position of the indentation centroid based, in first case, on the entire image, and, in the second case, on just the border pixels of the indentation. The bright pixels on the periphery of the

indentation are the border pixels, as found by the perimeter tracing algorithm, whilst the four straight lines inside the indentation form the best-fit quadrilateral approximation to the border.

The signature for the indentation perimeter is shown in Fig. 6.3. Clearly, the four peaks in the graph are not of equal magnitude, neither are they equally spaced, indicating a slight error in the centroid calculation (almost certainly introduced by the presence of spuriously thresholded blemishes on the surface of the sample); however, this seems to have had only a slight effect on the registration accuracy of the best-fit quadrilateral as seen in Plate VI. The results of the analysis program, as presented to the user, are shown in Fig. 6.4. It was noticed that in several samples the diagonal lengths of an indentation varied widely (see Fig. 6.6 for example) due to anisotropy in the sample crystal structure. In these cases, it is clearly important that the lengths of both diagonals be calculated and presented to the user, rather than simply calculating the indentation area by counting pixels. This accounts for the need for the perimeter calculation algorithm given in Chapter 4.

A design objective of the system was that the software should be capable of analysing poor quality images of the type shown in Plate VII. The sample shown here contains a large surface blemish which effectively reduces the contrast between the object and the background in the captured image. Using the grey level histogram (Fig. 6.5), it is difficult to delineate between object and background pixels and in this case the program cannot segment the indentation from the blemish region, as shown in Plate VIII. Under these circumstances, the analysis program recognises that the thresholding operation may give unsatisfactory results and isolates the indentation using region labelling (section 4.4.5). The outcome of the process is shown in Plate IX.

As indicated in section 4.3.4, the perimeter tracing algorithm for the Vickers test program was designed to be insensitive to defects adjacent to the indentation. An example of such a defect is shown in Fig. 6.6 where it can be seen that the best-fit quadrilateral (shown by the four dotted straight lines) is still a good approximation to the indentation border.

The image analysis program can also trap another error condition; the computed centroid lying outside of the indentation. An example of this is shown in Fig. 6.7 where the position of centroid (marked by the cross-hairs) has been biased towards the spuriously thresholded region at the bottom of the image (as seen holding the page vertically). In this instance the program isolates the indentation using region labelling and recalculates the centroid position to give a more accurate result. The outcome of this is shown in Fig. 6.8 where the centroid position is marked by the cross on the indentation.

As indicated in Chapter 4, the system was calibrated by moving a sample under the microscope, using a micrometer drive, and measuring the displacement, in pixels, of an indentation corner in relation to the physical displacement read from the micrometer barrel. The results of the calibration test are shown in Fig. 6.9. The scale factors for the system were found to be 454 pixels/mm in the horizontal direction and 673 pixels/mm in the vertical direction. The corresponding pixel resolutions are 2.2 μm and 1.5 μm which are in accordance with the BSI specified resolution of 2 μm. No significant geometric distortion is shown by the graphs. The maximum deviation of any of the points from the fitted line is approximately 5 pixels corresponding to a distance of 10 μm. This is in keeping with the estimated uncertainty in locating the indentation edge by eye ($\pm$ 5 μ m).

Tests were carried out several standard hardness blocks in order to gauge the accuracy of the system. The hardness of two of these blocks was stated by the manufacturers as:

Block A	Block B
214-223 HV5	336-345 HV5
215-220 HV30	337-344 HV30
215-220 HV50	339-346 HV50

Each block already contained several indentations made at different loads (5 Kg, 30 Kg and 50 Kg) and for each indentation the hardness was first measured using a standard Vickers machine and then using the vision system. In each case the highest available microscope magnification (x10) was used in the vision system and the magnification used on the Vickers machine was also kept constant. In the case of Block B, the indentations made using the 5 Kg load were too small to be analysed by the vision system. The results of the tests are shown in tables 6.3 and 6.4. The entries in the confidence interval row indicate the expected variation in the population mean for a 95% certainty, as determined by the z test.

	214-223 HV5		215-220 HV30		215-220 HV50	
	Vickers Machine	Vision System	Vickers Machine	Vision System	Vickers Machine	Vision System
Mean Vickers Hardness	223	240	223	228	227	229
Standard Deviation (σ)	5	9	12	13	9	8
Number of Measurements	6		28		9	
Confidence Interval	231-249		223-233		223-235	

Table 6.3: Vickers Hardness Test Results for Block A

	337-344 HV5		339-346 HV30	
	Vickers Machine	Vision System	Vickers Machine	Vision System
Mean Vickers Hardness	338	351	338	345
Standard Deviation (σ)	14	18	25	27
Confidence Interval	343-359		328-362	
Number of Measurements	21		12	

Table 6.4: Vickers Hardness Test Results for Block B

It can be seen that some of the hardness values determined from the Vickers machine actually fall outside the manufacturers stated range. This may have been due to defects in the machine when applying the test loads. The deviation from the mean of the

manufacturers stated hardness is at most 7%, for the 214-223 HV30 indentations, but in all other cases is no greater than 5%. In light of this, it is more instructive to assess the accuracy of the vision system by comparing the hardness values found by the vision system to those found from the Vickers machine rather than the manufacturers stated values. On this basis the accuracy of the vision system in all cases is better than 4%. It can also be seen that the hardness values as determined by the vision system are all greater than those determined by the Vickers machine. This means that the vision system has a tendency to under estimate the size of indentations. The likely cause of this is the erroneous thresholding of pixels lying on the border of the indentation as background pixels rather than object pixels. This problem is discussed further in Chapter 7.

It can be seen that the precision of the vision system, as judged by the standard deviation values, is roughly the same as that of the Vickers machine. In the latter case precision is determined by the ability of the user to position the microscope vernier at the exact corners of the indentation. The effects of 'sinking in' and 'piling up' (see section 2.2.3) as well as finite depth of field in the microscope mean that the edges of the indentation do not appear as sharp black to white transitions but contain grey areas. The user therefore has to make an arbitrary decision on the position of the indentation corners. The likely positioning error was estimated at $\pm 5\mu m$ leading to an error of $\pm 10\mu m$ in the diagonal length measurement. The effects of this random error on the hardness measurement are shown in table 6.5. Clearly the smaller the indentation the greater the error in the hardness measurement will be. These observations are partly bourne out by comparison of table 6.5 with the confidence intervals shown in tables 6.3 and 6.4. However, there still seems to be a small, systematic over estimation of the hardness by the vision system. This is probably due to a slight error in the camera calibration.

To some extent, the same points also apply to the vision system. Any blurring of the indentation edges will reduce the image contrast and make the thresholding process less reliable. Since the effectiveness of the thresholding algorithm was determined subjectively the vision system precision cannot be greater than the precision of the manual system. This is verified by the results shown in table 6.3 and 6.4.

Vickers Hardness	Mean Diagonal Length (μm)	Hardness Range due to Measurement Errors
220 HV5	206	199-241 HV5
340 HV30	404	324-358 HV30
220 HV30	506	208-226 HV30
340 HV50	525	325-351 HV50
220 HV50	643	210-224 HV50

Table 6.5: Effect ofUncertainty in Diagonal Length on Hardness Measurements

Two additional tests were made to ensure that the vision system software was insensitive to changes in the indentation position and orientation. In the former case the position of an indentation (made using a 30 Kg load) was altered, using two micrometer drives, placed normal to each other, and the hardness result produced by the system recorded for each new position. The results are presented in table 6.6.

	Vertical Displacement (mm)			
Horizontal Displacement (mm)	**0.000**	**0.005**	**0.010**	**0.015**
0.000	266	270	270	264
0.005	268	274	272	274
0.010	268	272	269	264
0.015	264	262	267	265
0.020	269	268	271	268
0.025	262	263	259	264

Table 6.6: Variation In Hardness Values With Position

The mean hardness was 268 HV30 and the standard deviation 4 so that almost 70% (i.e. the fraction $erf(\frac{1}{\sqrt{2}})$) of the results were within ± 1.5% of the mean. Furthermore, none of the results deviate by more that ± 3% of the mean. The same number of measurements were made on the Vickers machine giving a mean hardness of 260 HV30 with a standard deviation of 5. The spread of results in table 6.6 is within the precision of the system, as discussed earlier, therefore there is no discernible relationship between measured hardness and indentation position.

In the orientation test, the centre of the an indentation (made using a 20 Kg load) was placed at the centre of the vision camera field of view and the camera rotated on the microscope eye piece. The hardness result, produced by the system, was then recorded for each rotation angle. The results are presented in table 6.7.

rotation (degrees)	0	45	90	135	180	225	270	315
hardness	206	205	207	209	211	210	205	205

Table 6.7: Variation In Hardness Values With Rotation

The mean hardness was 207 HV20 and the standard deviation 2 so that almost 70% of the results were within ± 1.1% of the mean value. Furthermore, none of the results deviate by more than ± 2% of the mean value. The same number of measurements were made on the Vickers machine giving a mean hardness of 203 HV20 with a standard deviation of 3. The spread of results in table 6.7 is within the precision of the system, as discussed earlier, therefore there is no discernible relationship between measured hardness and indentation orientation.

6.4 QUALITY ASSESSMENT SYSTEM

A typical set of samples to be analysed by the quality assessment system is shown in Plate X; each disc component being approximately 8 mm in diameter. The grey scale image of one such component is shown in Fig. 6.10; in this case the disc contains a crack extending from the inner to the outer edge. The same image as processed by the vision system is shown in Fig. 6.11 where the thresholding and inner border tracing operations have been applied. Here the traced border is shown by the bright pixels. Clearly, the border tracing algorithm has traced around both the inner and outer edge of the disc indicating a fully penetrant crack. This condition was detected by the analysis program.

An example of a component containing a large chip in the outer edge is shown in Fig. 6.12 while Fig. 6.13 shows the same image after processing by the vision system. The processed image illustrates the results of thresholding followed by tracing of the inner and outer edges of the disc and filling of the hole and annular regions. The dotted white lines form the fitted ellipses whereas the white pixel clusters correspond to chip defects as identified by the analysis program. Fig. 6.14 shows the same image with the grey level of pixels corresponding to the hole region remapped to improve visibility. An approximate scale has also been added and in this case the x axis is vertical as seen on the page.

The results of the ellipse fitting operations are given in Fig. 6.15. The inner edge of the disc is calculated as circular while the outer edge has been found to be exhibit slight eccentricity; furthermore, there is only a small offset between the edge centres. The inner and outer edge profiles for the disc are shown in Fig. 6.16 and Fig. 6.17 where the angle of rotation is taken in the clockwise sense starting at the -x axis. Parts of the border

falling outside of the fitted ellipse have negative deviations in the profile and parts of the border falling inside the fitted ellipse positive deviations. The chip in the outer edge can clearly be identified by the peak at 220 - 240 degrees while small chips in the inner edge are seen as peaks around 260 degrees in the inner edge profile.

The analysis of chip and crack defects is shown in Fig. 6.18. For each defect region the area is given along with the perimeter length, compactness (P*P/A), location of the centroid, maximum bounding radius (max r), minimum bounding radius (min r) and ratio of maximum to minimum radius (max/min). In this example the chip defects are not significant.

Another example of a component containing defects is shown in Fig. 6.19. The processed image (Fig. 6.20) reveals that the disc has a chip on the outer edge as well as a large surface chip, located at (154,174), and two possible surface cracks, located at (289,396) and (318,405). The ellipse results (Fig. 6.21) indicate slight eccentricity in both the inner and outer edges as well as a small hole offset. The inner edge profile (Fig. 6.22) indicates few defects in the inner edge of the disc whereas the outer edge chip can clearly be identified in the outer edge profile (Fig. 6.23) by the peak at 290-300 degrees. From the list of chip parameters (Fig. 6.24 and 6.25) a distinction can be drawn between the defect located at (154,174) and the defect located at (289,396). In the former case the low values of compactness (= 15) and ratio of maximum to minimum radius (= 2) indicate a chip feature while in the latter case the high values of compactness (= 109) and maximum to minimum radius (≐ 18) indicates a crack-like feature.

All of the samples discussed so far contain defects which signal immediate failure of the component, however fig. 6.26 and 6.27 illustrate a component of acceptable quality. The results (Fig. 6.28) show small eccentricity in the disc edges along with a small hole offset.

The profiles (Fig. 6.29 and 6.30) indicate no appreciable chips in the inner edge but small chips in the outer edge. Chip and crack defects on the surface of the component are also shown to be negligible (Fig. 6.31 and 6.32).

At present the effectiveness of the system can only be measured in relation to the information originally given by the component manufacturers; this consisted of a qualitative description of the defects within samples and a simple accept/reject decision for each. The results here show sufficiently large differences in defect parameters between acceptable and unacceptable components that an accept/reject decision can be made. In any practical system however, qualitative limits will need to be imposed on defect parameters. Furthermore, the system may require more "intelligence" in deciding which combinations of defects are acceptable and which are not. These points are discussed further in Chapter 7.

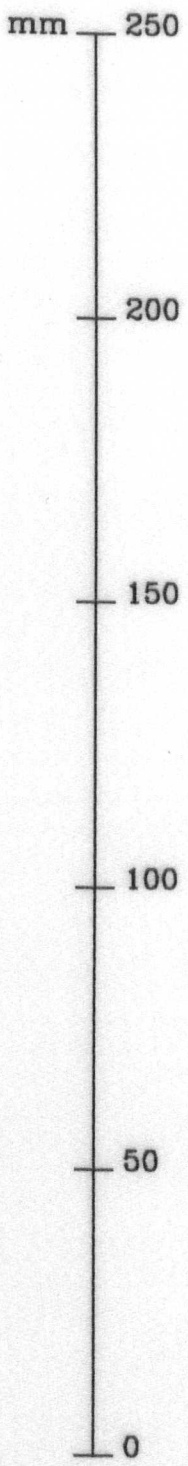


PLATE IV: TYPICAL PATH RECORD

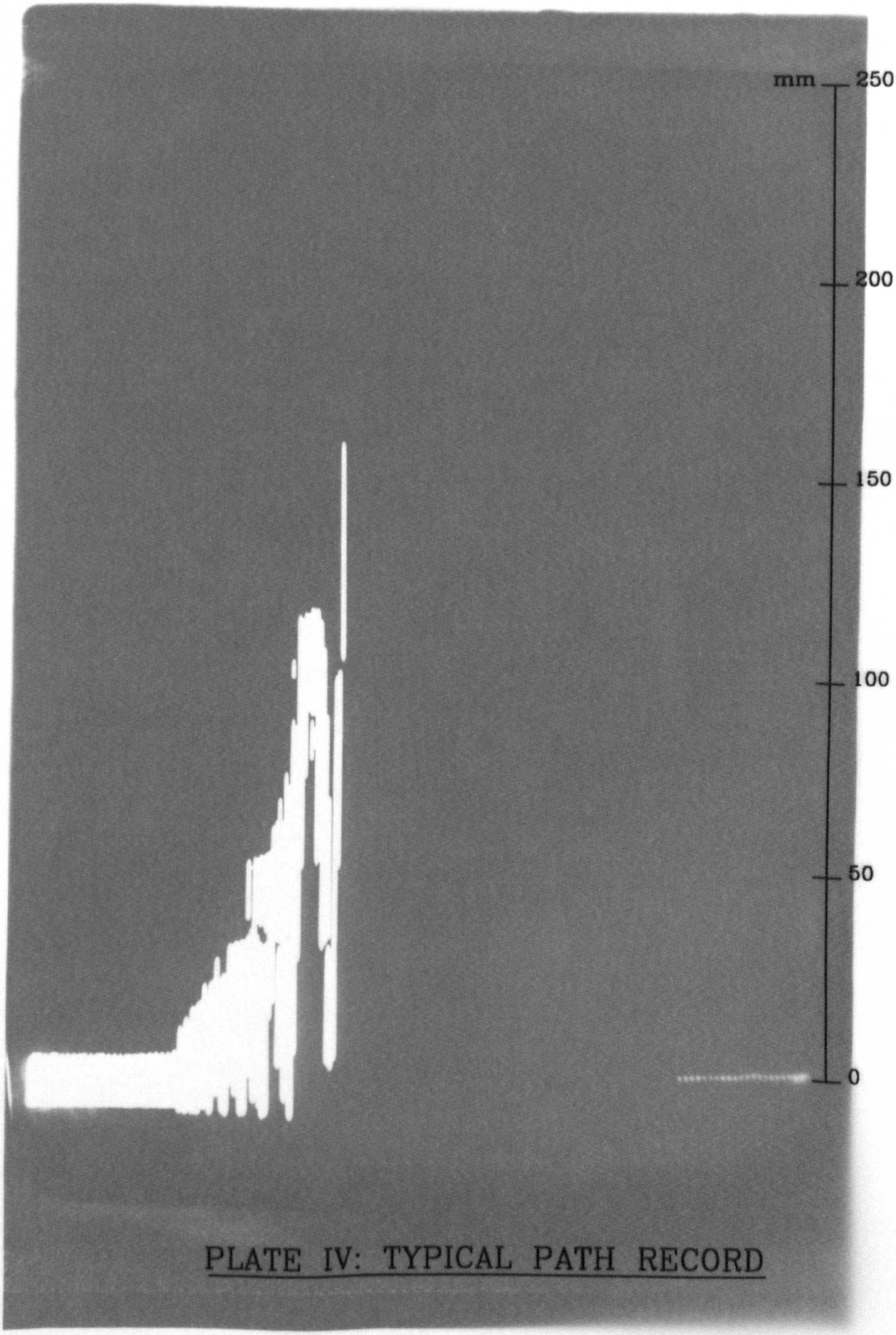
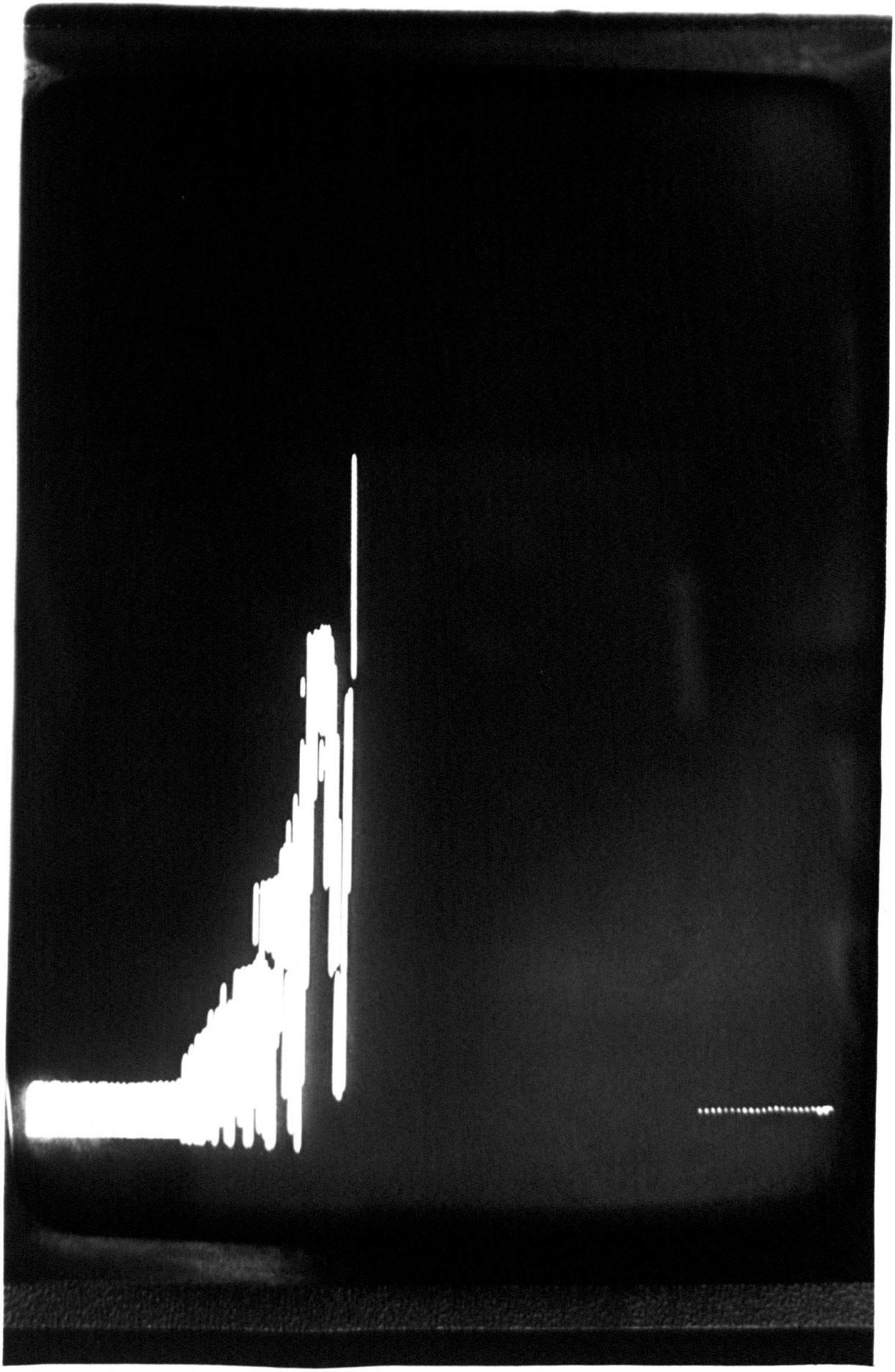
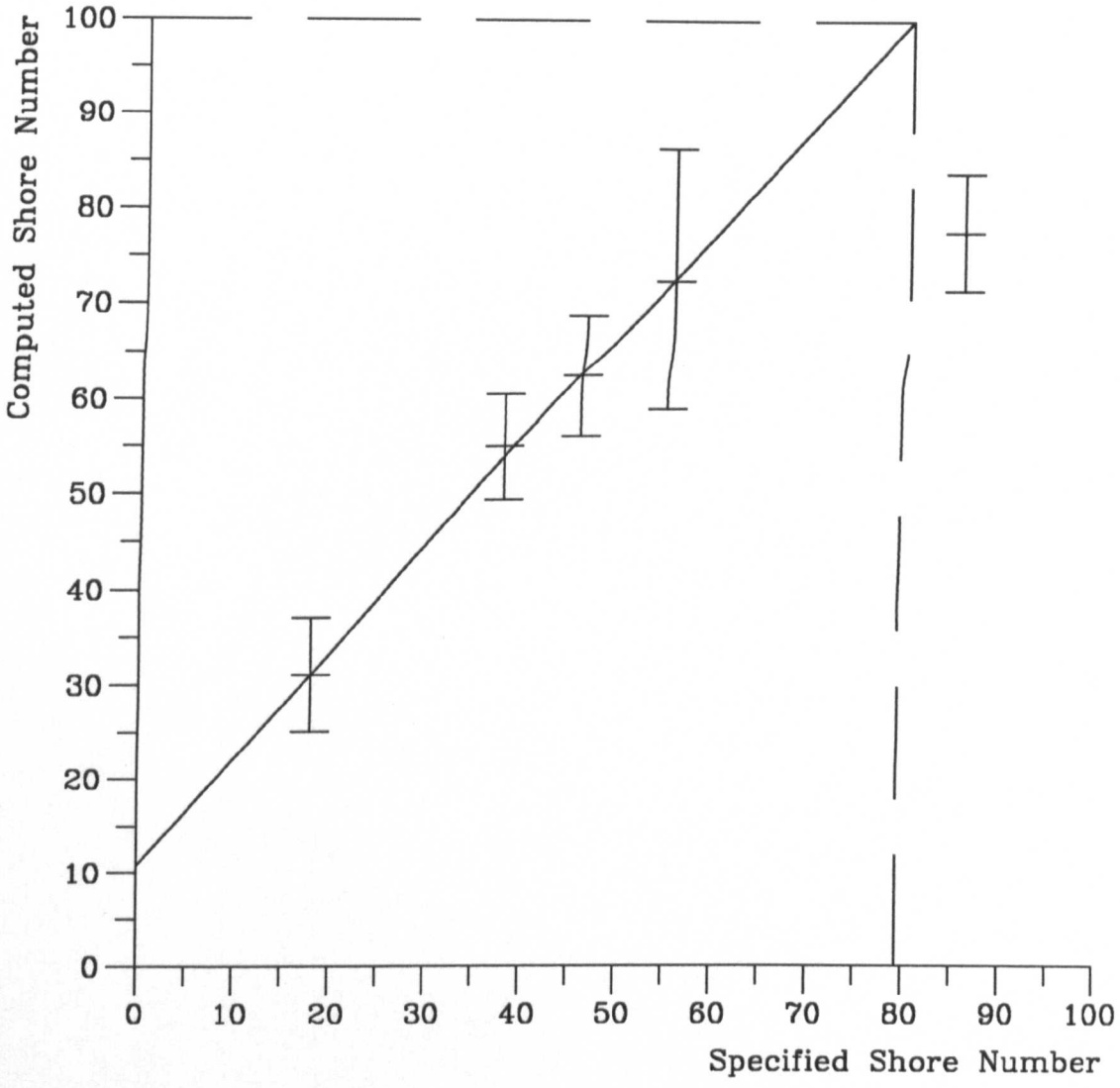


PLATE IV: TYPICAL PATH RECORD





$$\text{slope} = \frac{100.0 - 10.7}{79.4} = 1.12$$

$$\text{intercept} = 10.7$$

Figure 6.1: SHORE TEST RESULTS

PLATE V: TYPICAL VICKERS TEST INDENTATION

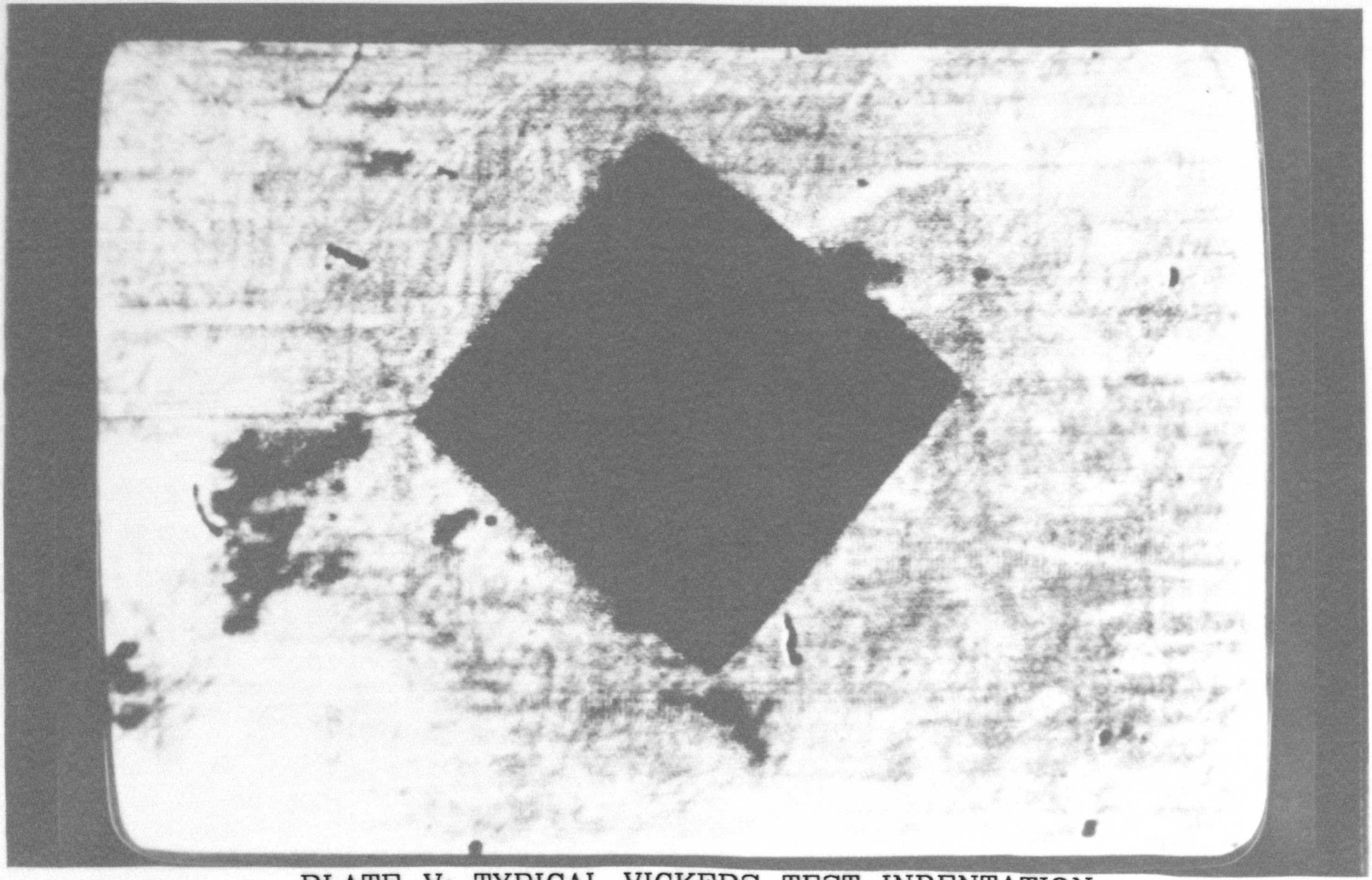
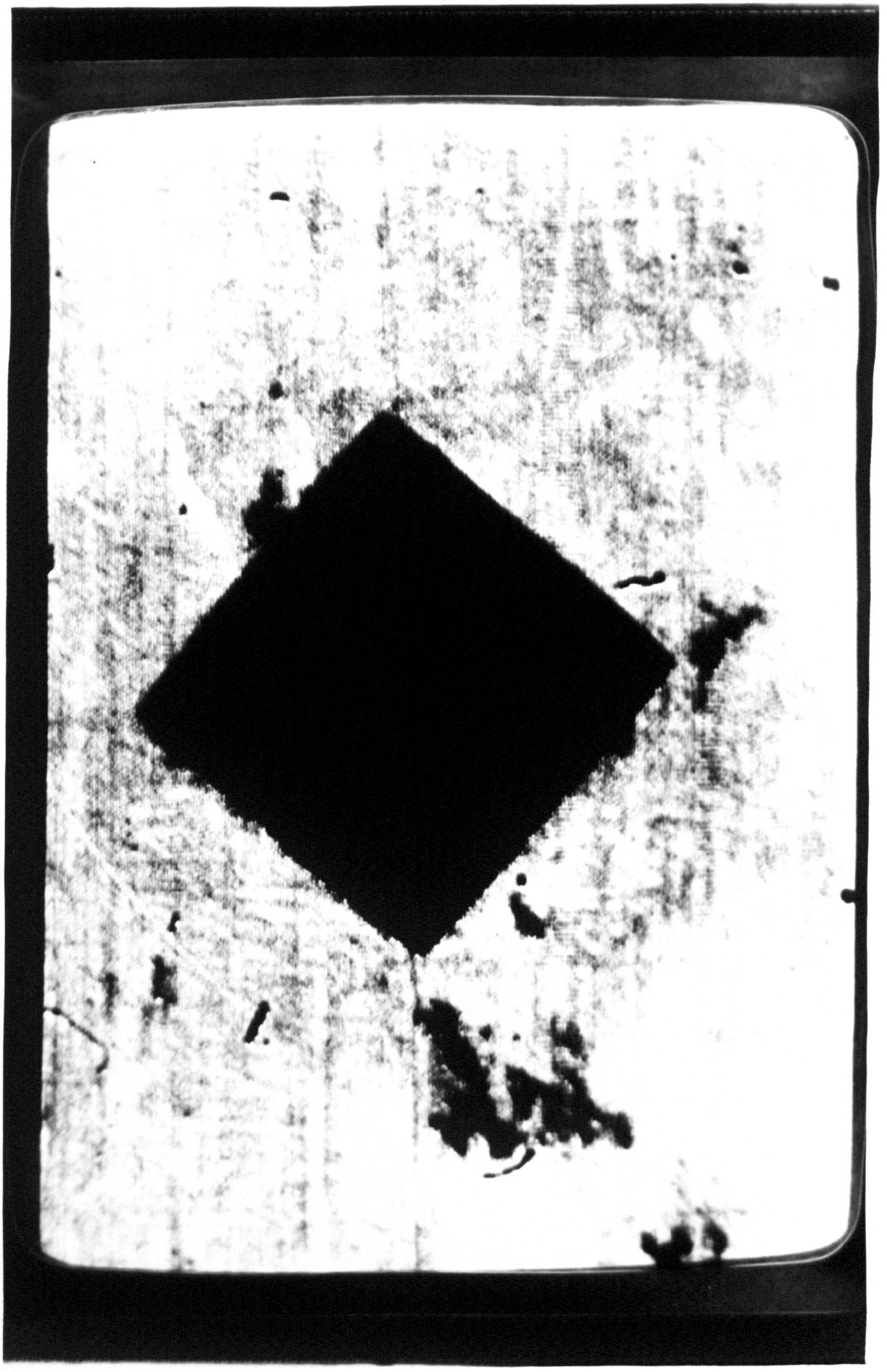


PLATE V: TYPICAL VICKERS TEST INDENTATION



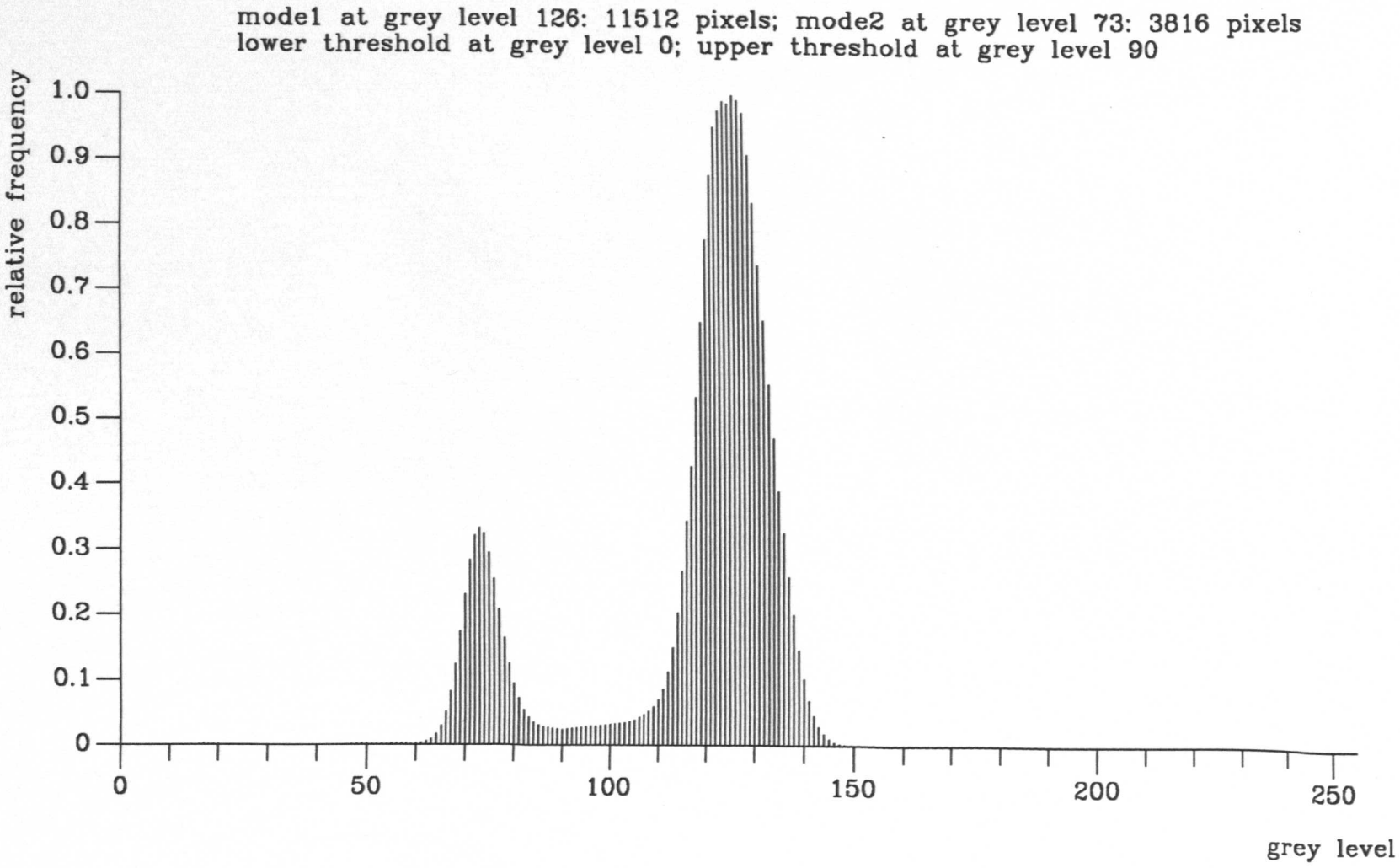


Figure 6.2: GREY LEVEL HISTOGRAM FOR IMAGE SHOWN IN PLATE V

PLATE VI: PROCESSED VERSION OF IMAGE SHOWN IN PLATE V

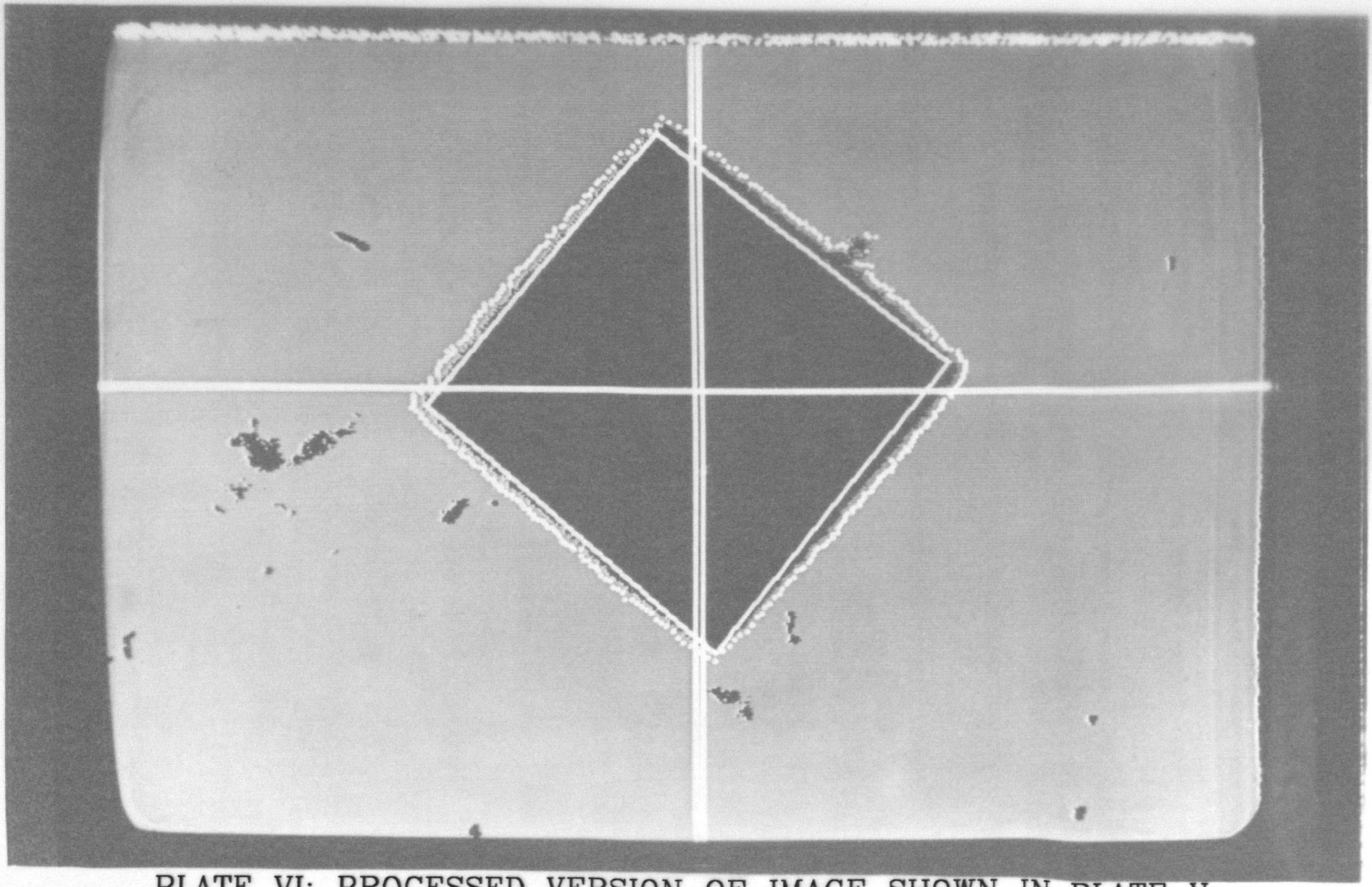
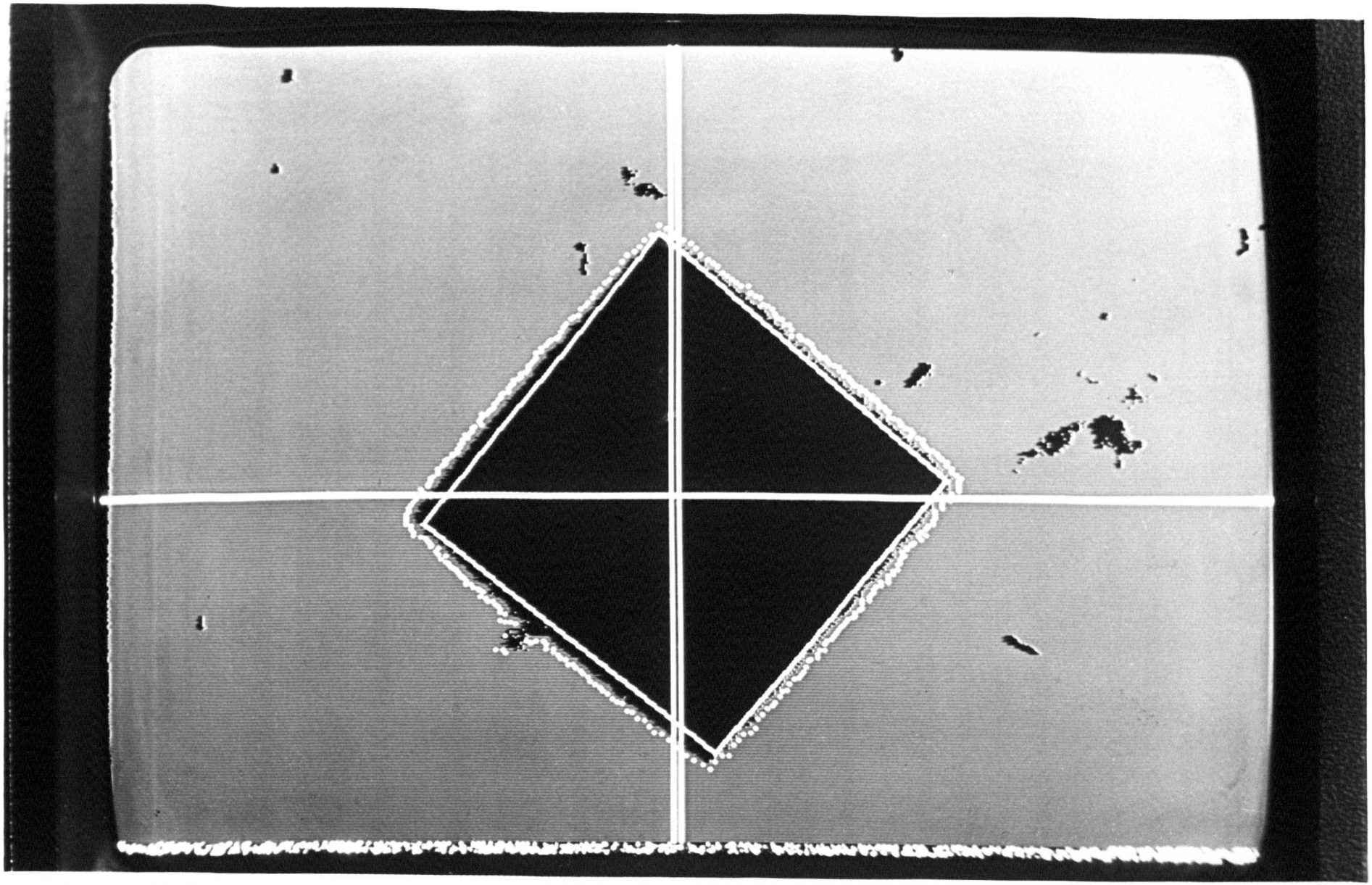


PLATE VI: PROCESSED VERSION OF IMAGE SHOWN IN PLATE V



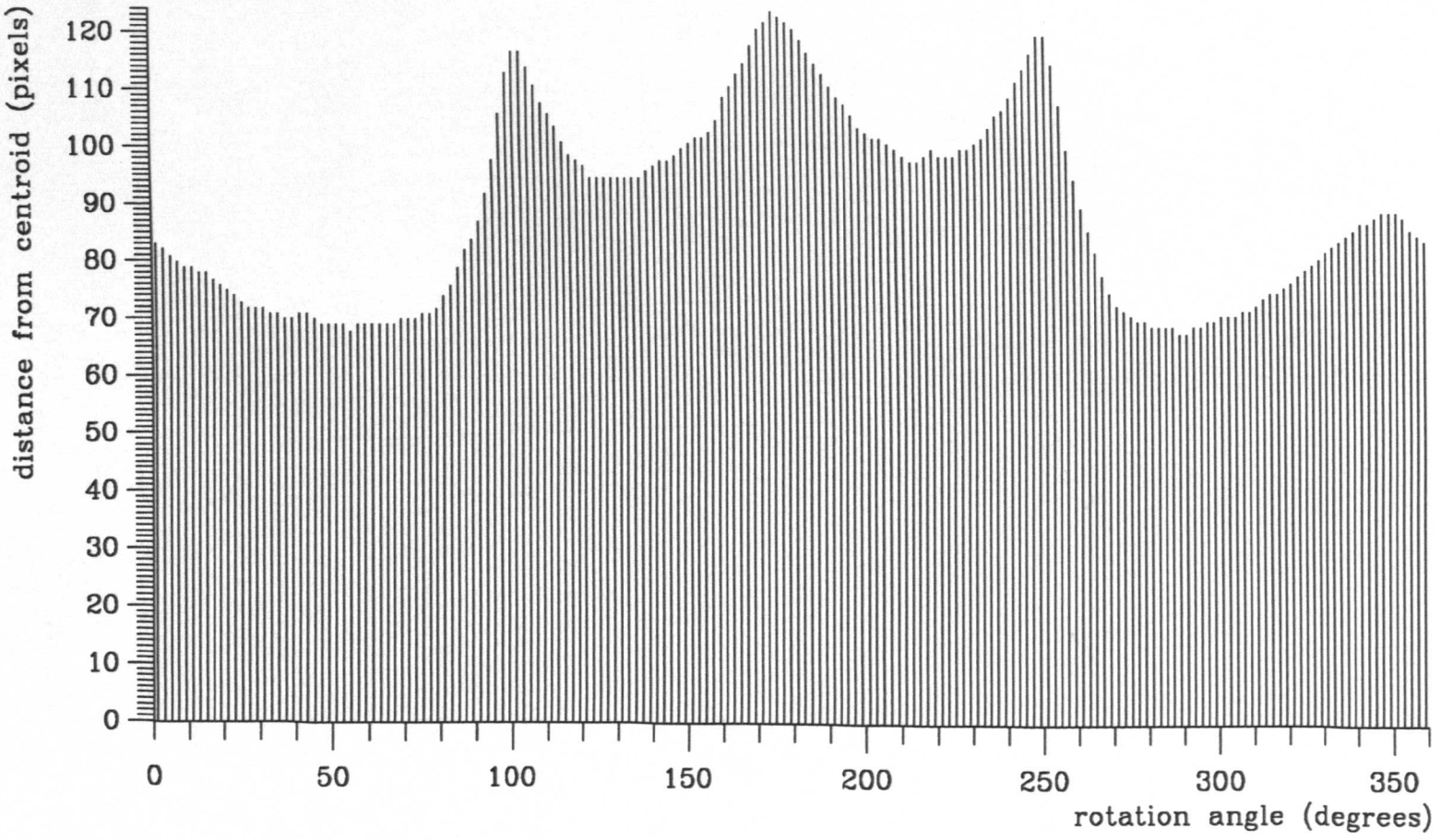


Figure 6.3: SIGNATURE FOR THE INDENTATION SHOWN IN PLATE V

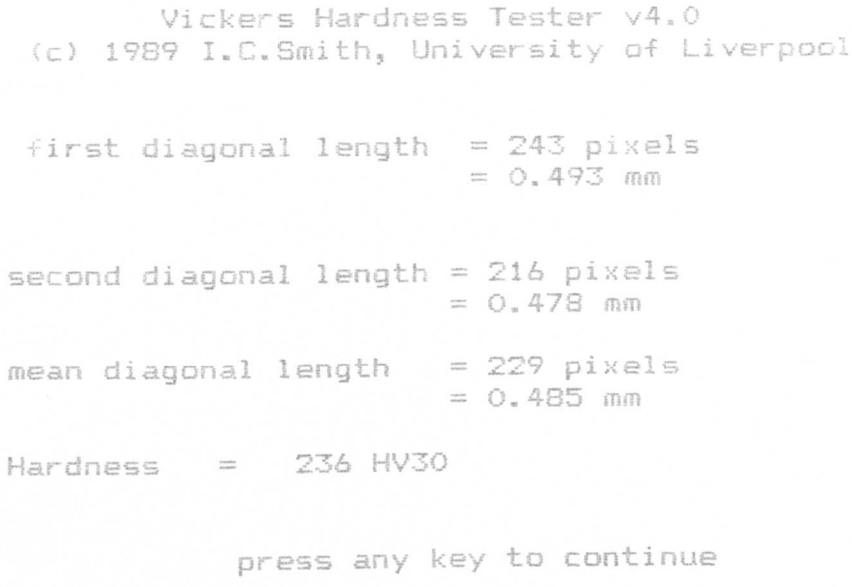


Figure 6.4: TEST RESULTS FOR INDENTATION SHOWN IN PLATE V

PLATE VII: VICKERS SAMPLE HAVING POOR CONTRAST

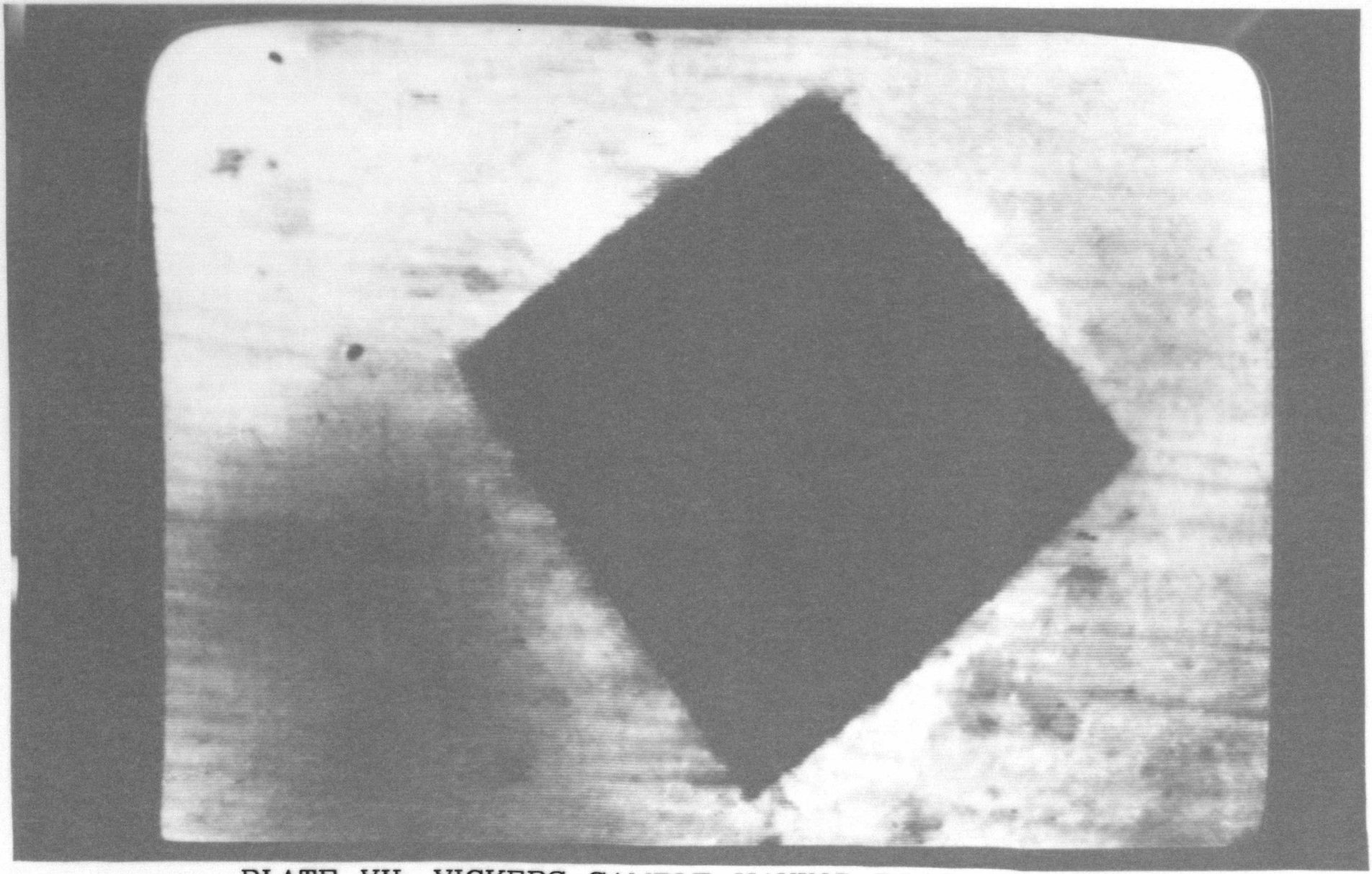
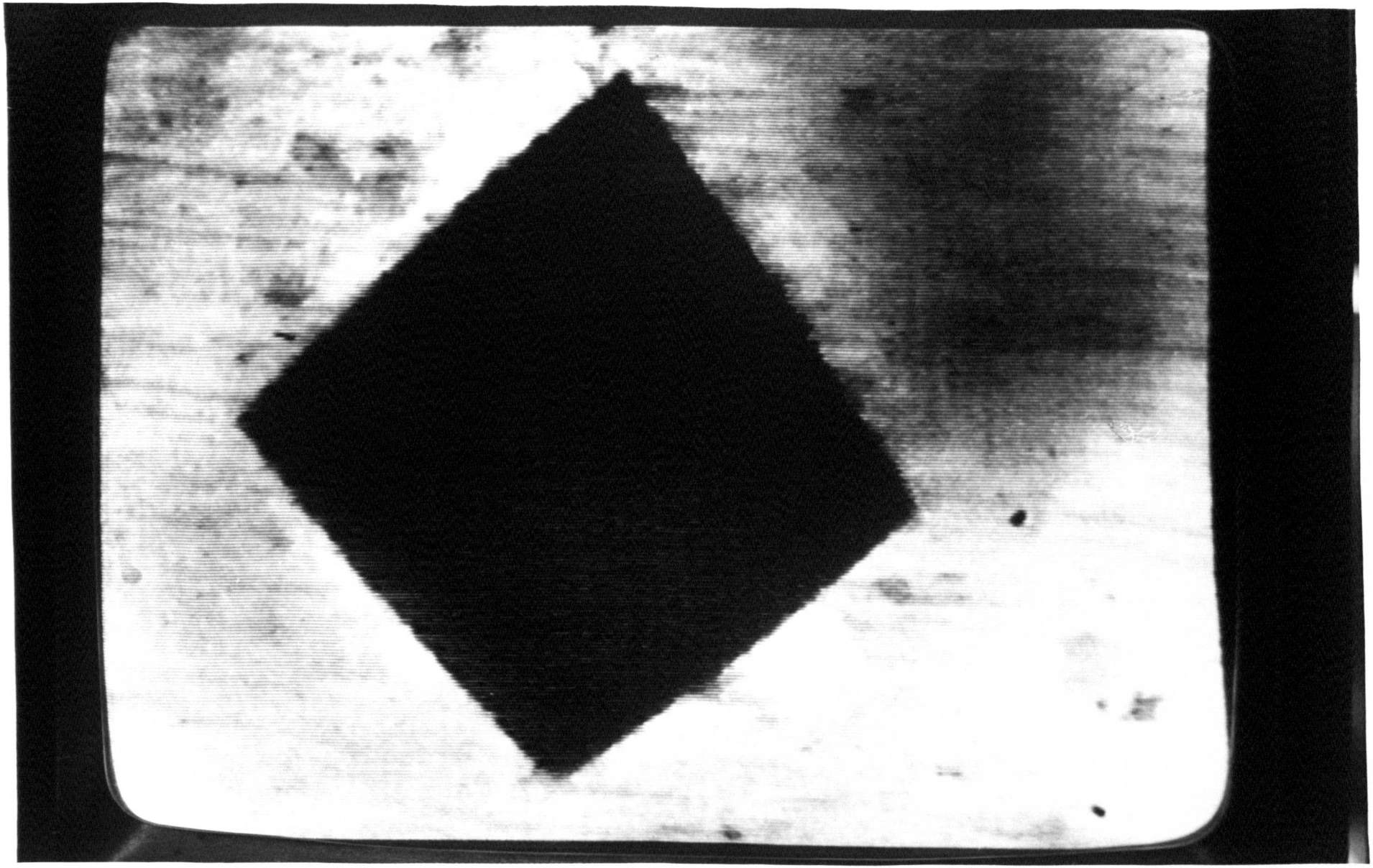


PLATE VII: VICKERS SAMPLE HAVING POOR CONTRAST



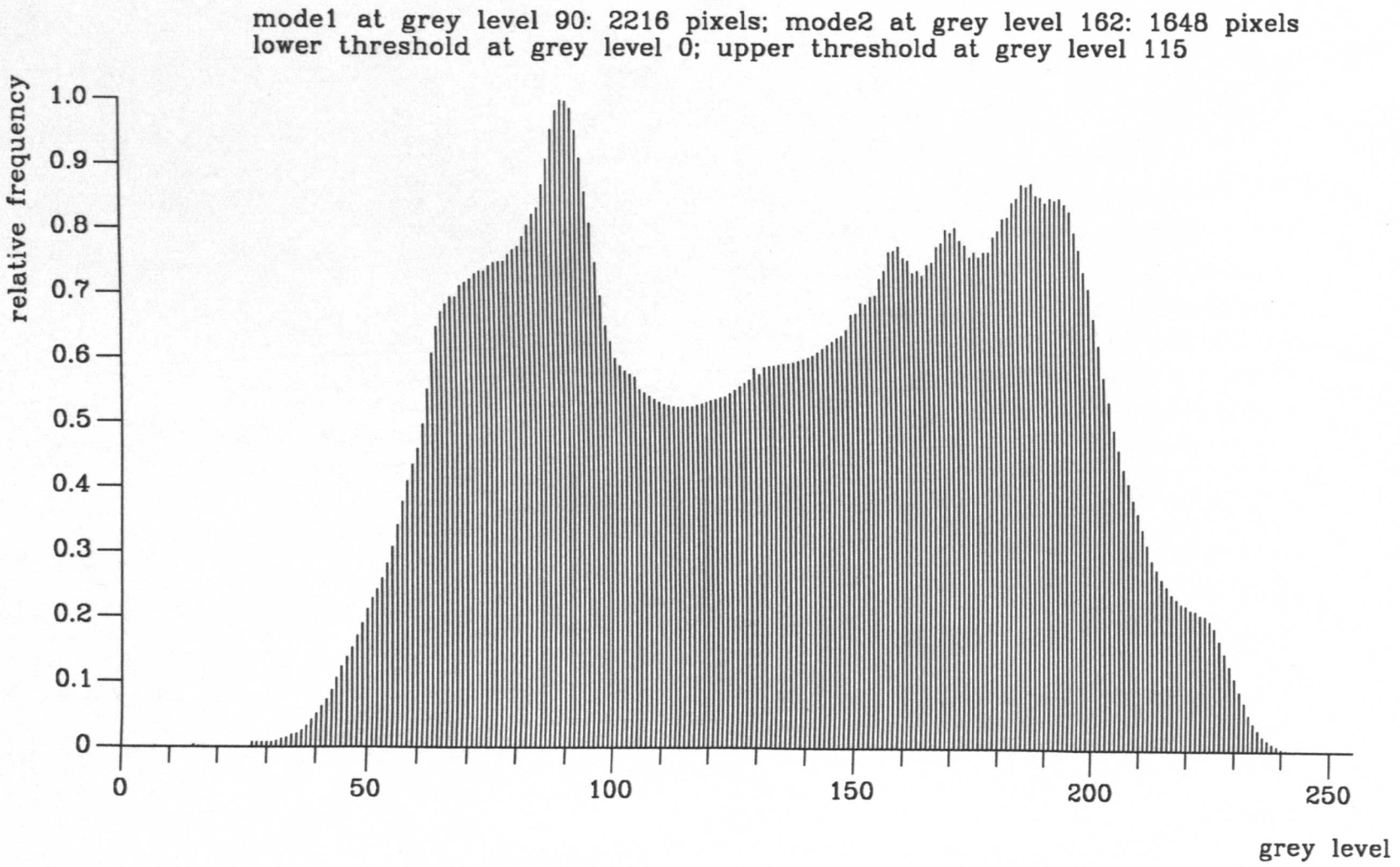


Figure 6.5: GREY LEVEL HISTOGRAM FOR IMAGE SHOWN IN PLATE VII

PLATE VIII: RESULT OF THRESHOLDING IMAGE SHOWN IN PLATE VII

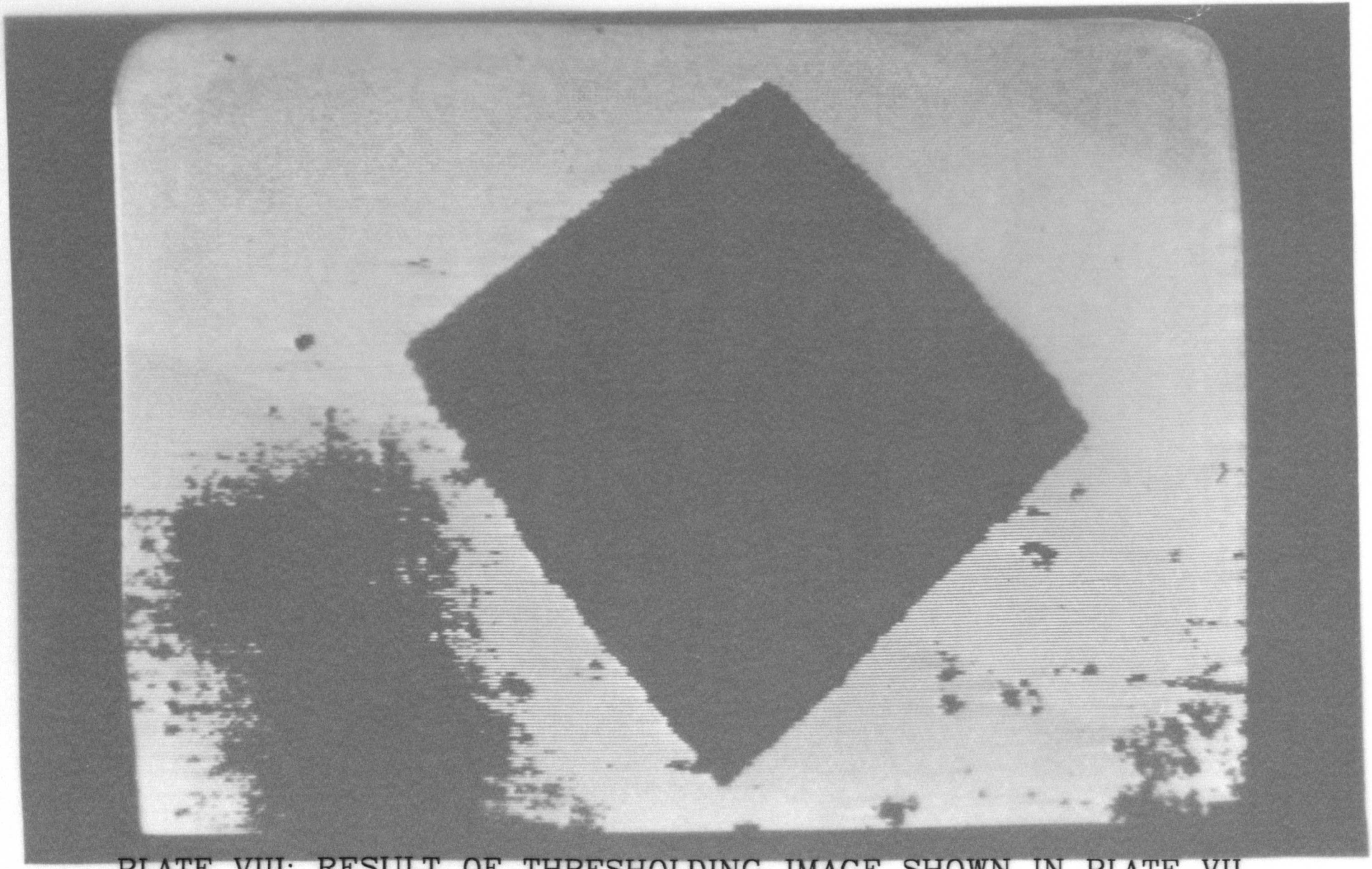


PLATE VIII: RESULT OF THRESHOLDING IMAGE SHOWN IN PLATE VII

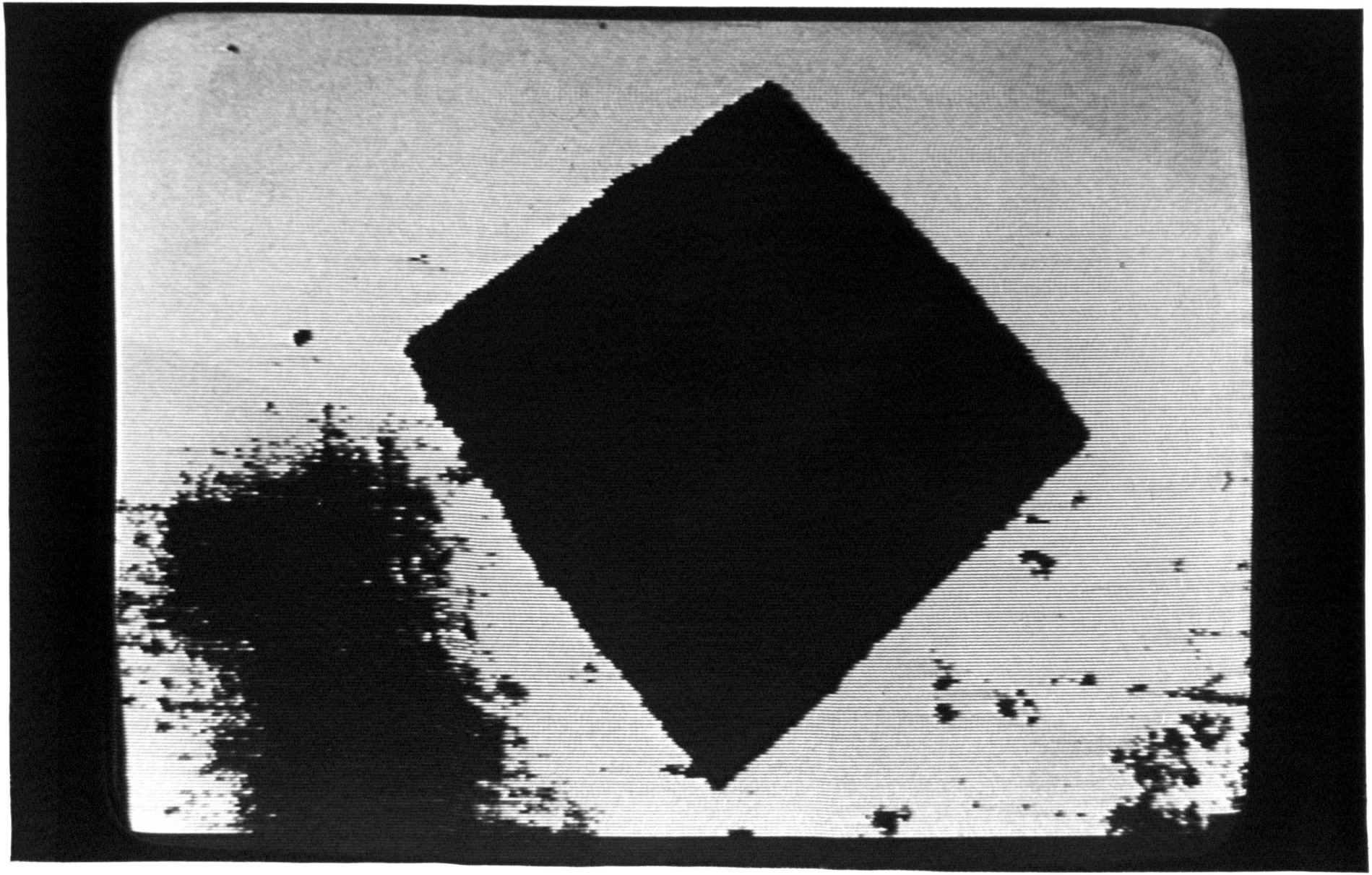


PLATE IX: RESULT OF ISOLATING LARGEST REGION FROM IMAGE SHOWN IN PLATE VIII

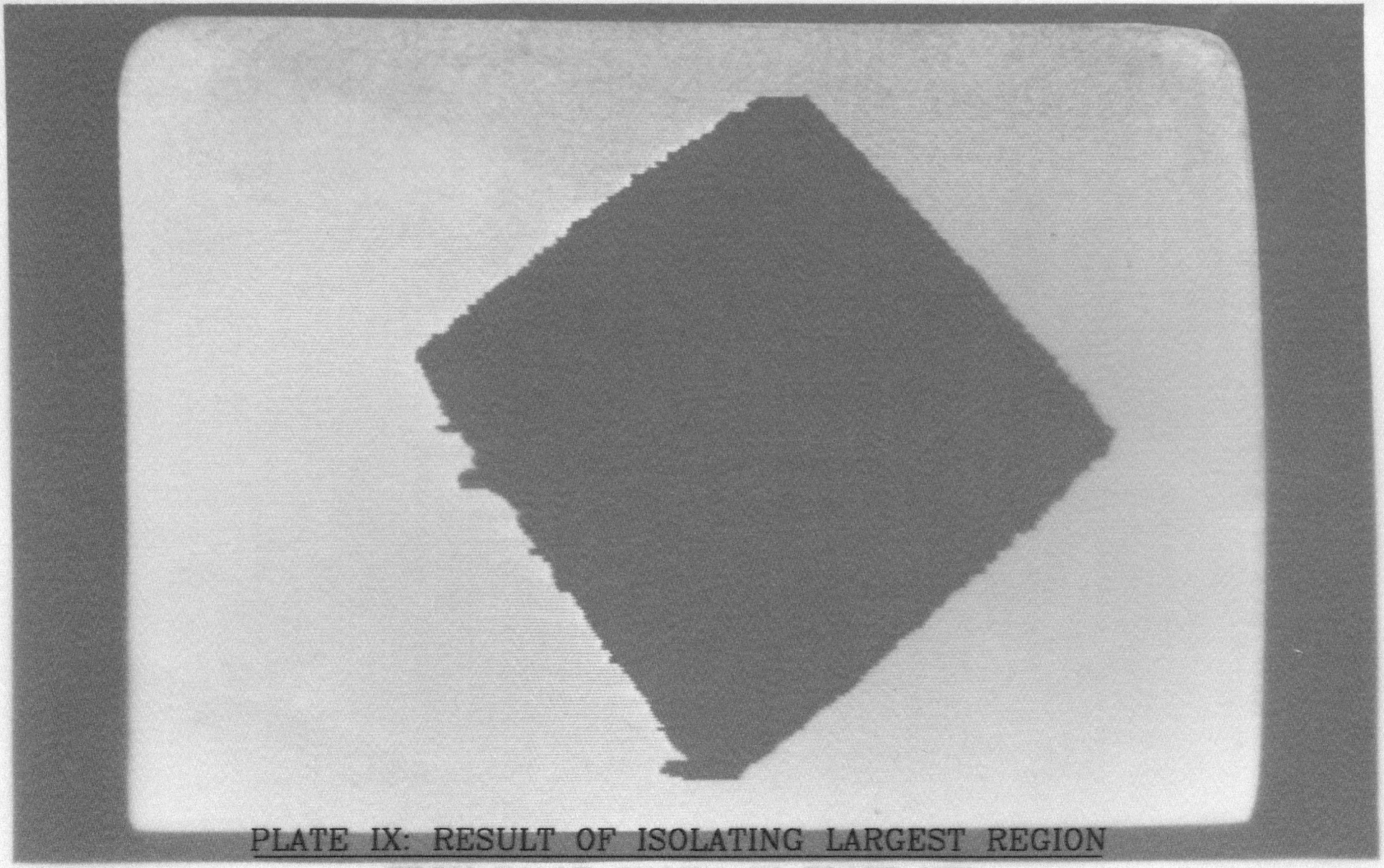
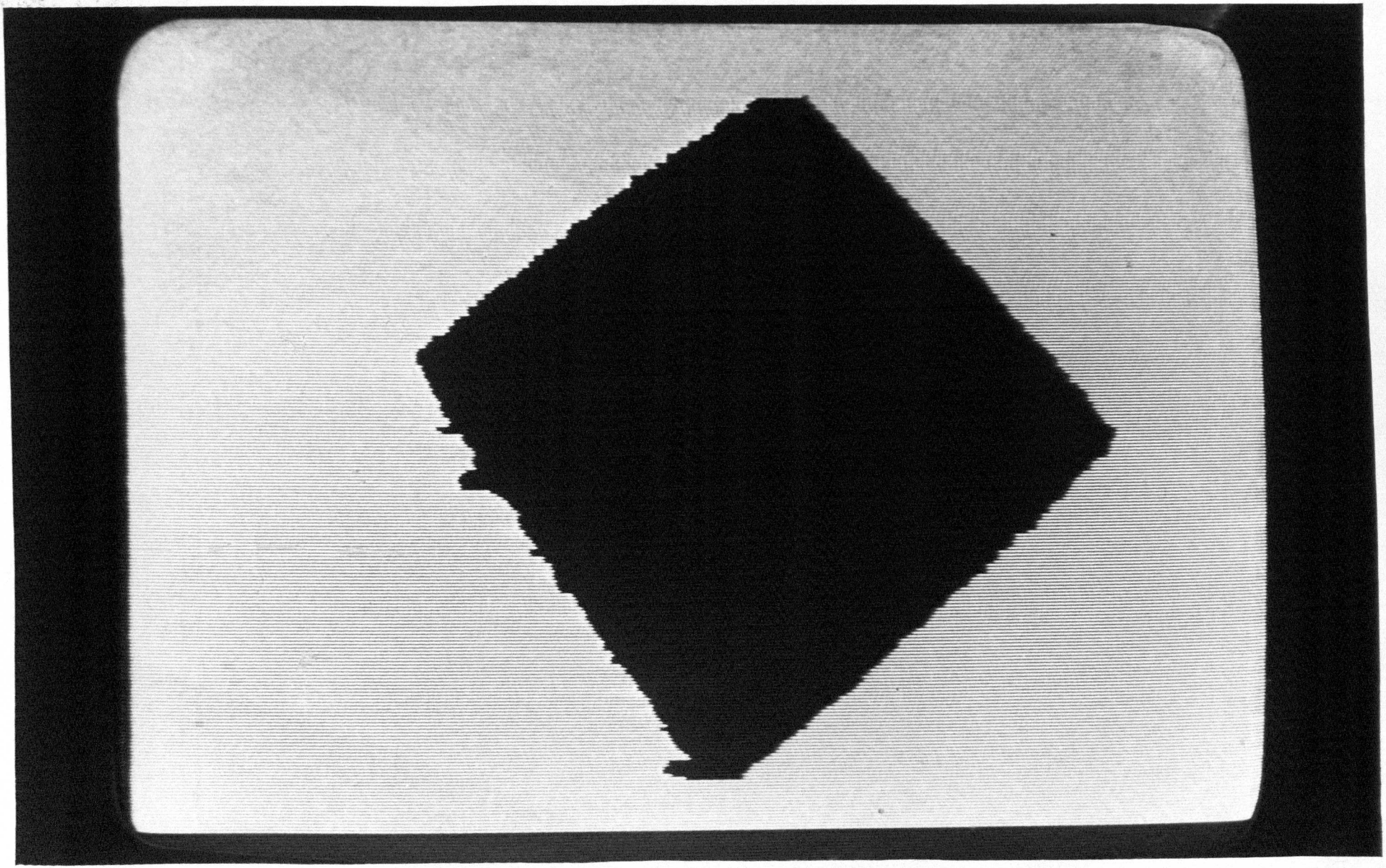


PLATE IX: RESULT OF ISOLATING LARGEST REGION
FROM IMAGE SHOWN IN PLATE VIII



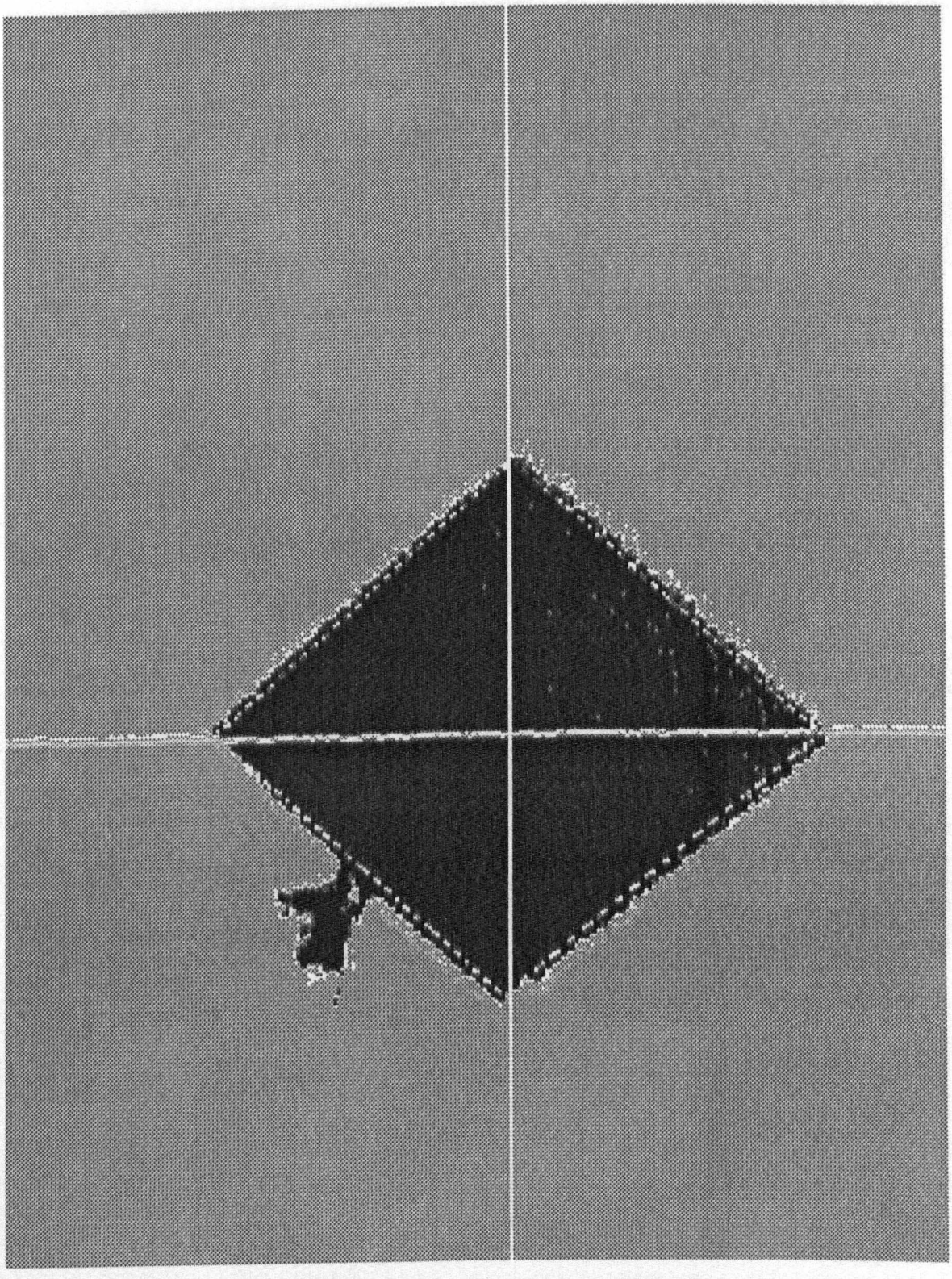


Figure 6.6: VICKERS SAMPLE CONTAINING DEFECT ADJACENT TO INDENTATION

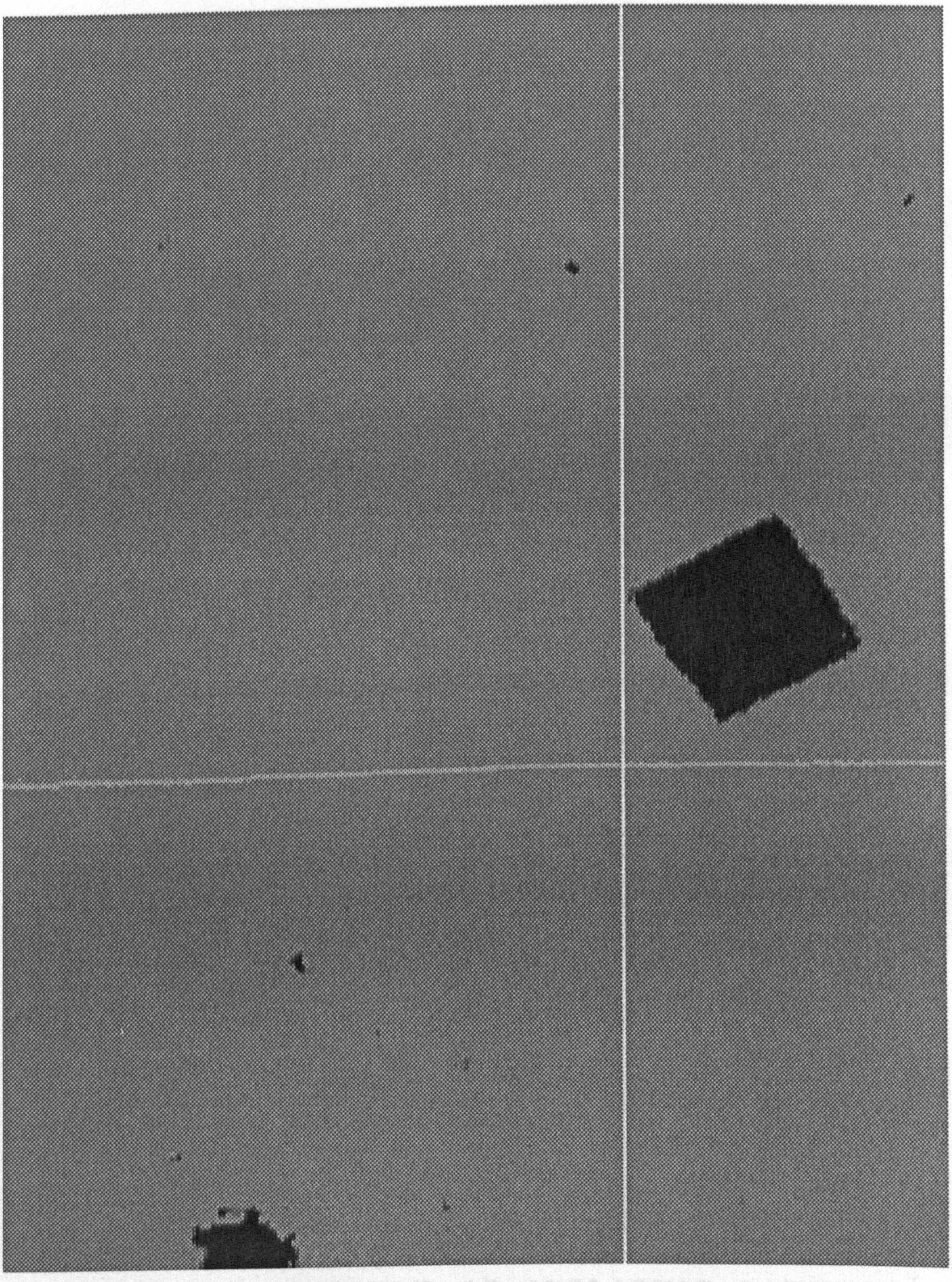


Figure 6.7: IMAGE IN WHICH CENTROID FALLS OUTSIDE OF INDENTATION

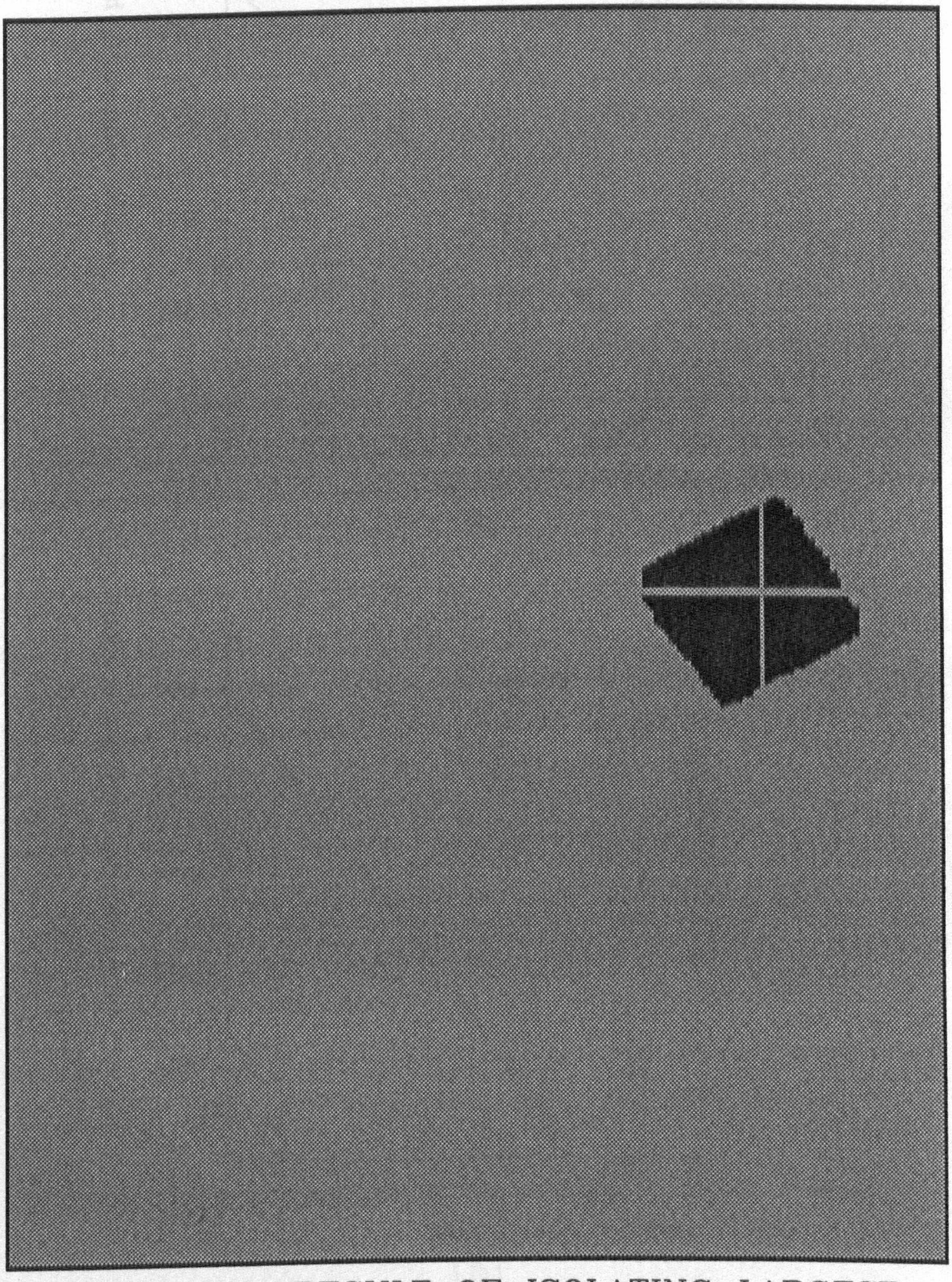
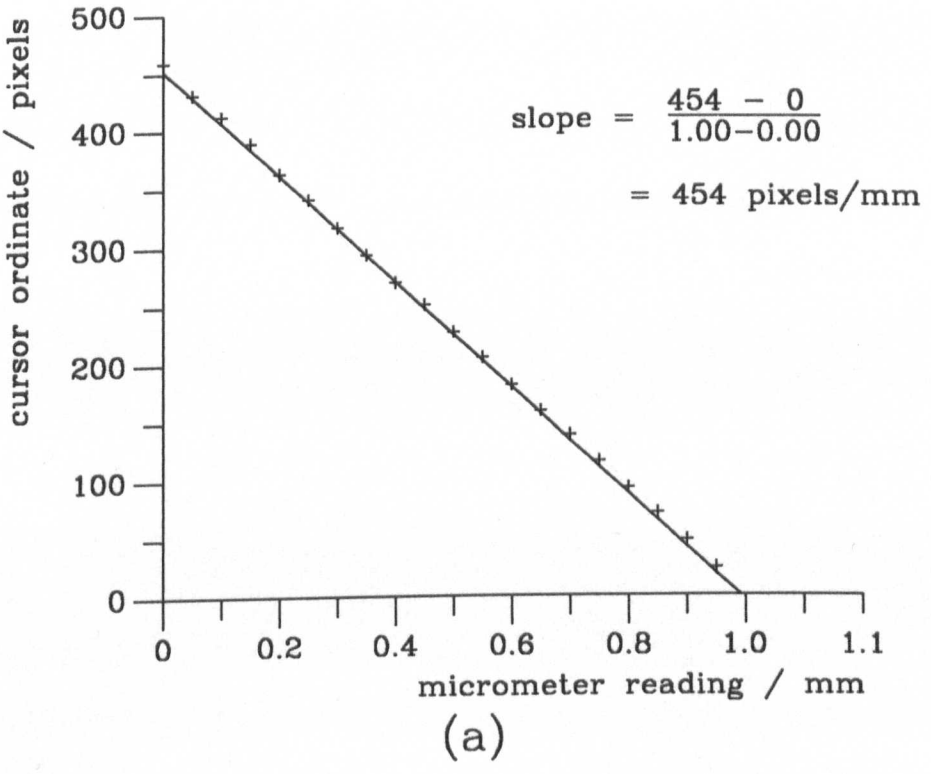


Figure 6.8: RESULT OF ISOLATING LARGEST REGION FROM IMAGE SHOWN IN FIG. 6.6



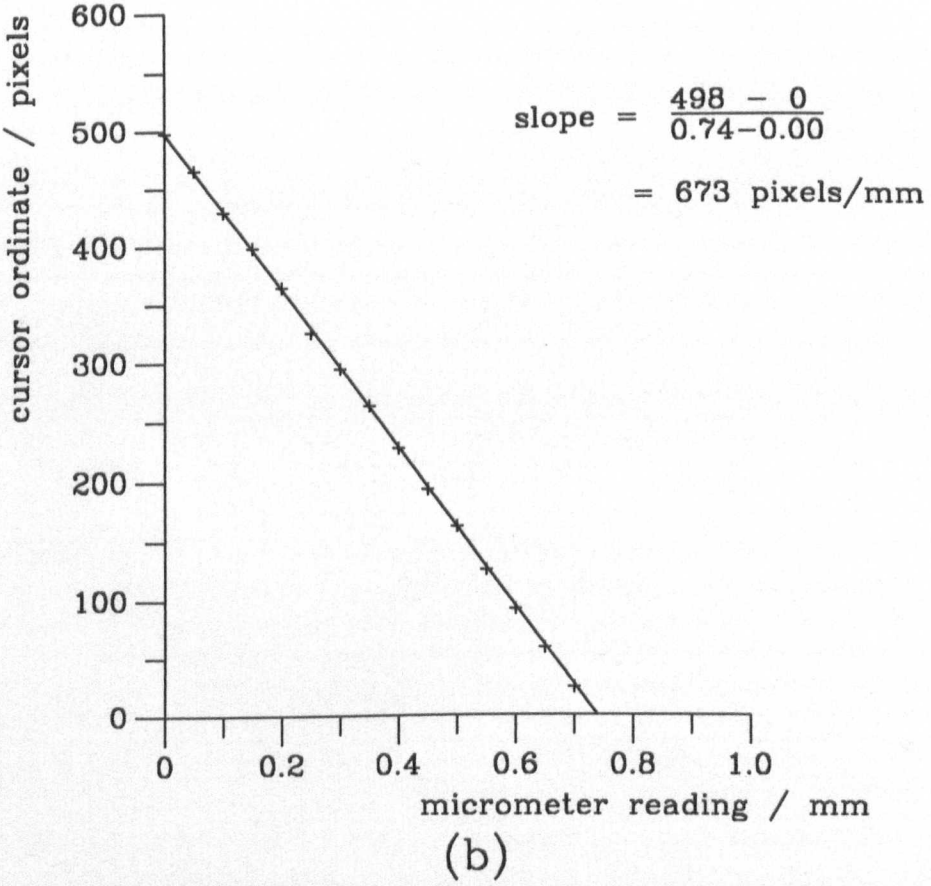


Figure 6.9: CALIBRATION GRAPHS FOR VICKERS TEST

(a) horizontal direction, (b) vertical direction

PLATE X: TYPICAL FERRITE SAMPLES

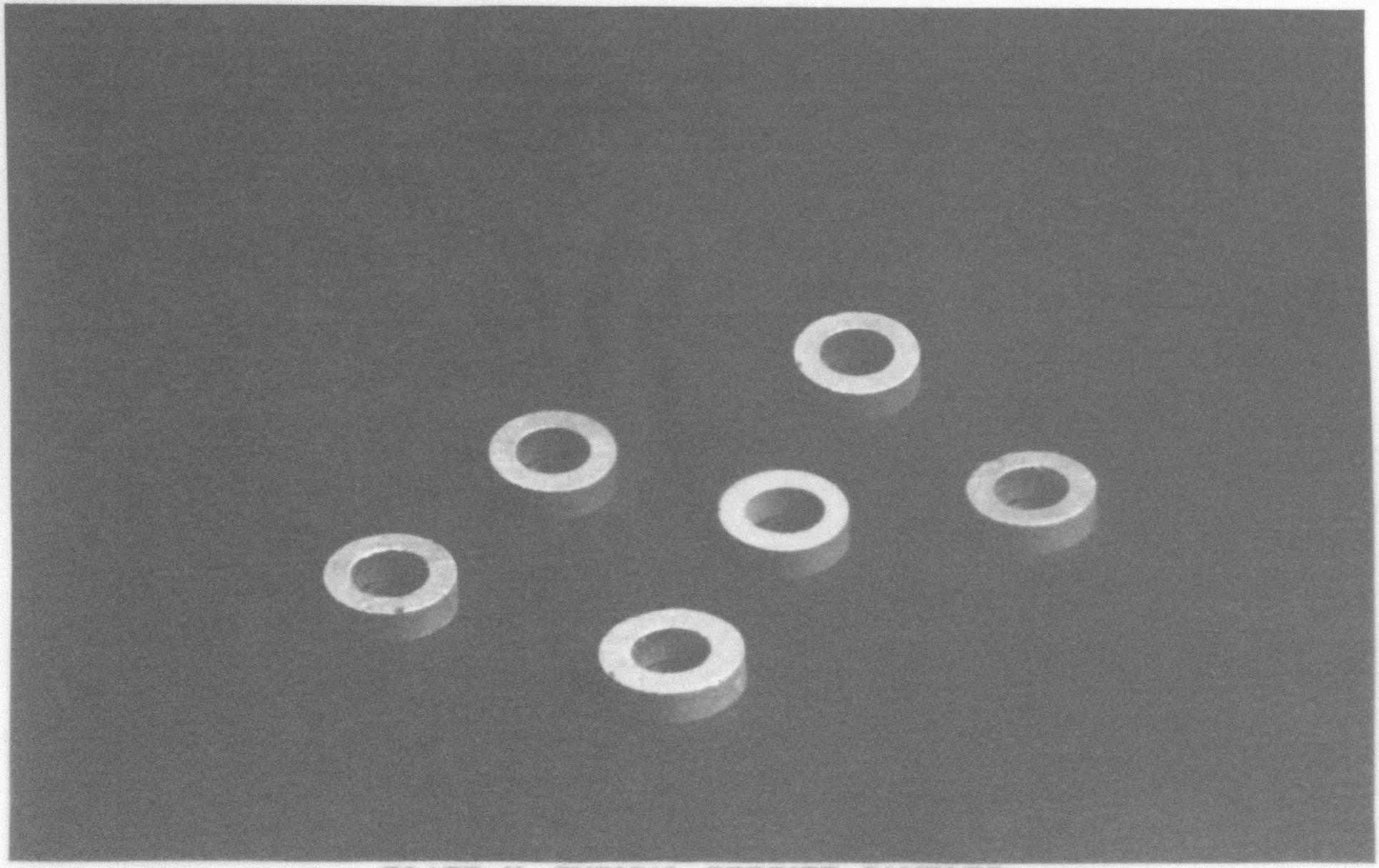
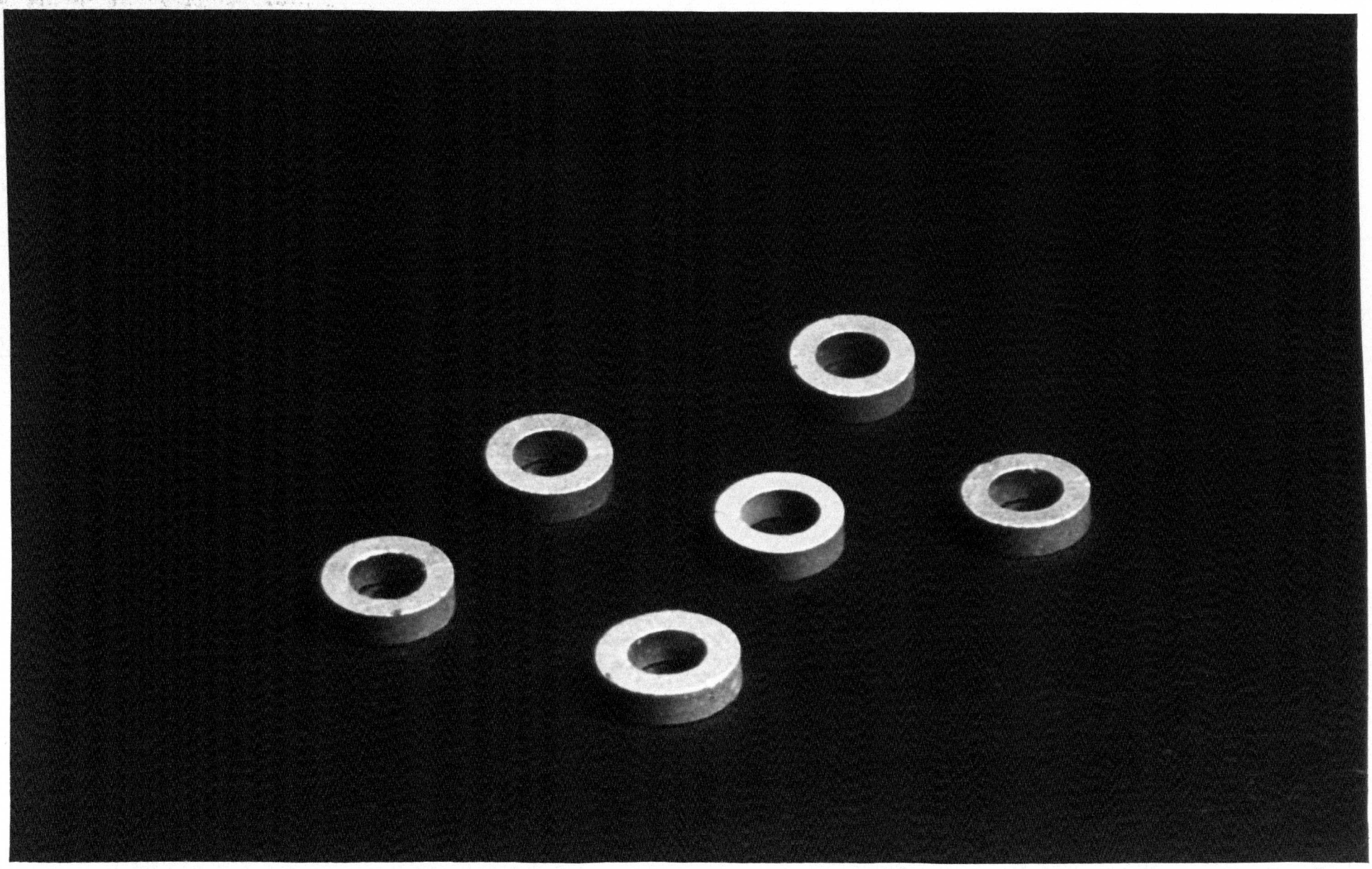


PLATE X: TYPICAL FERRITE SAMPLES



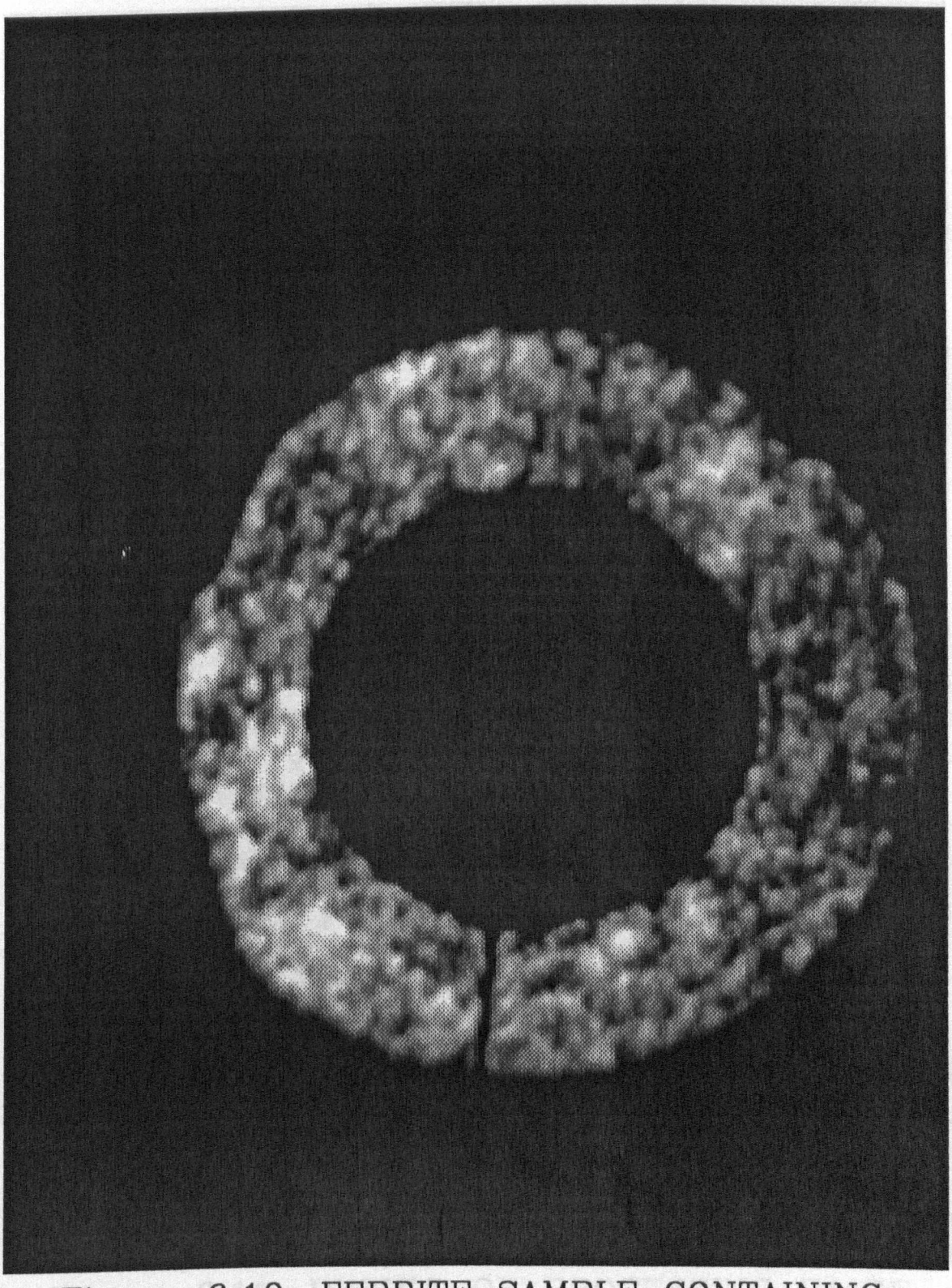


Figure 6.10: FERRITE SAMPLE CONTAINING FULLY PENETRANT CRACK

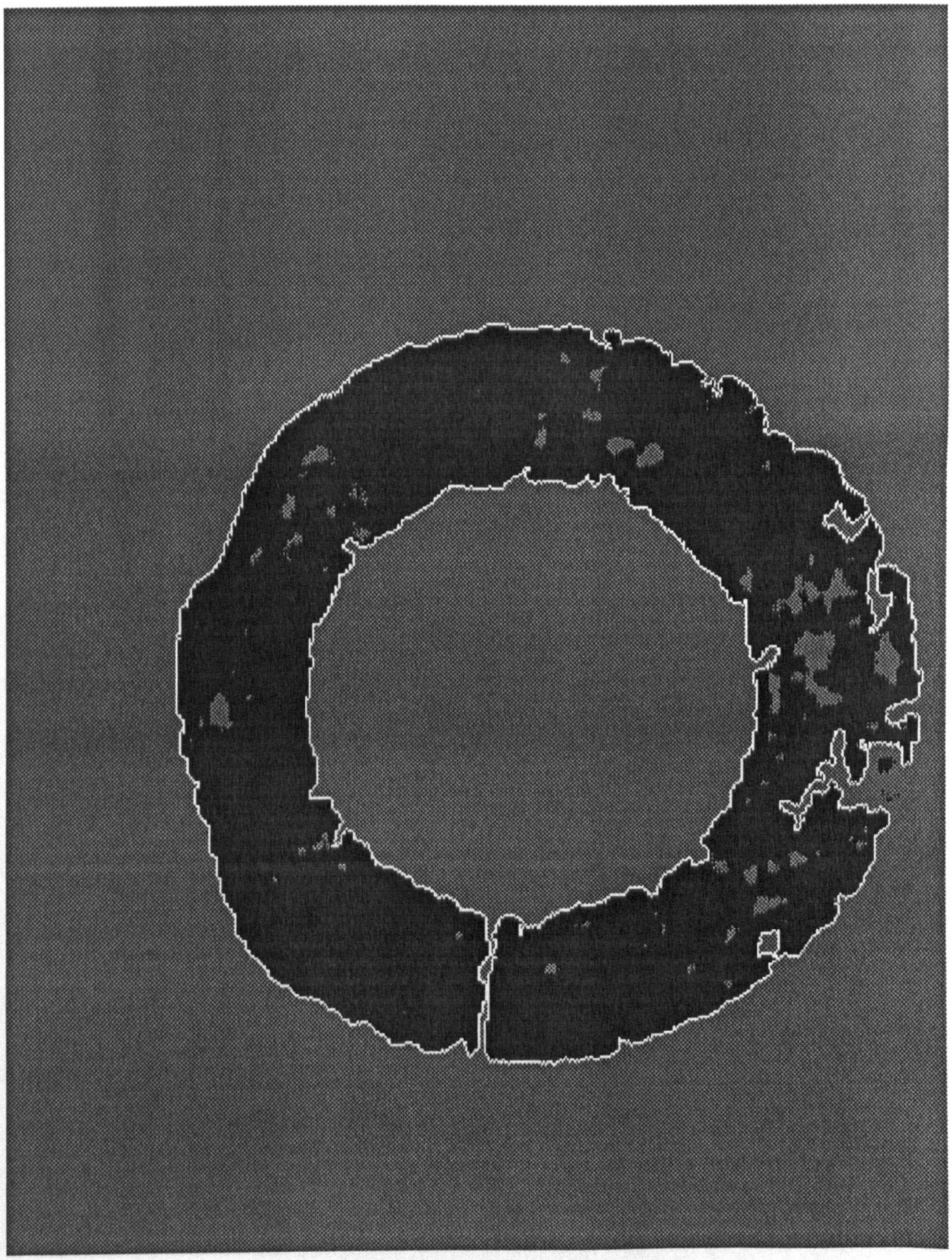


Figure 6.11: RESULT OF PROCESSING IMAGE SHOWN IN FIG. 6.10

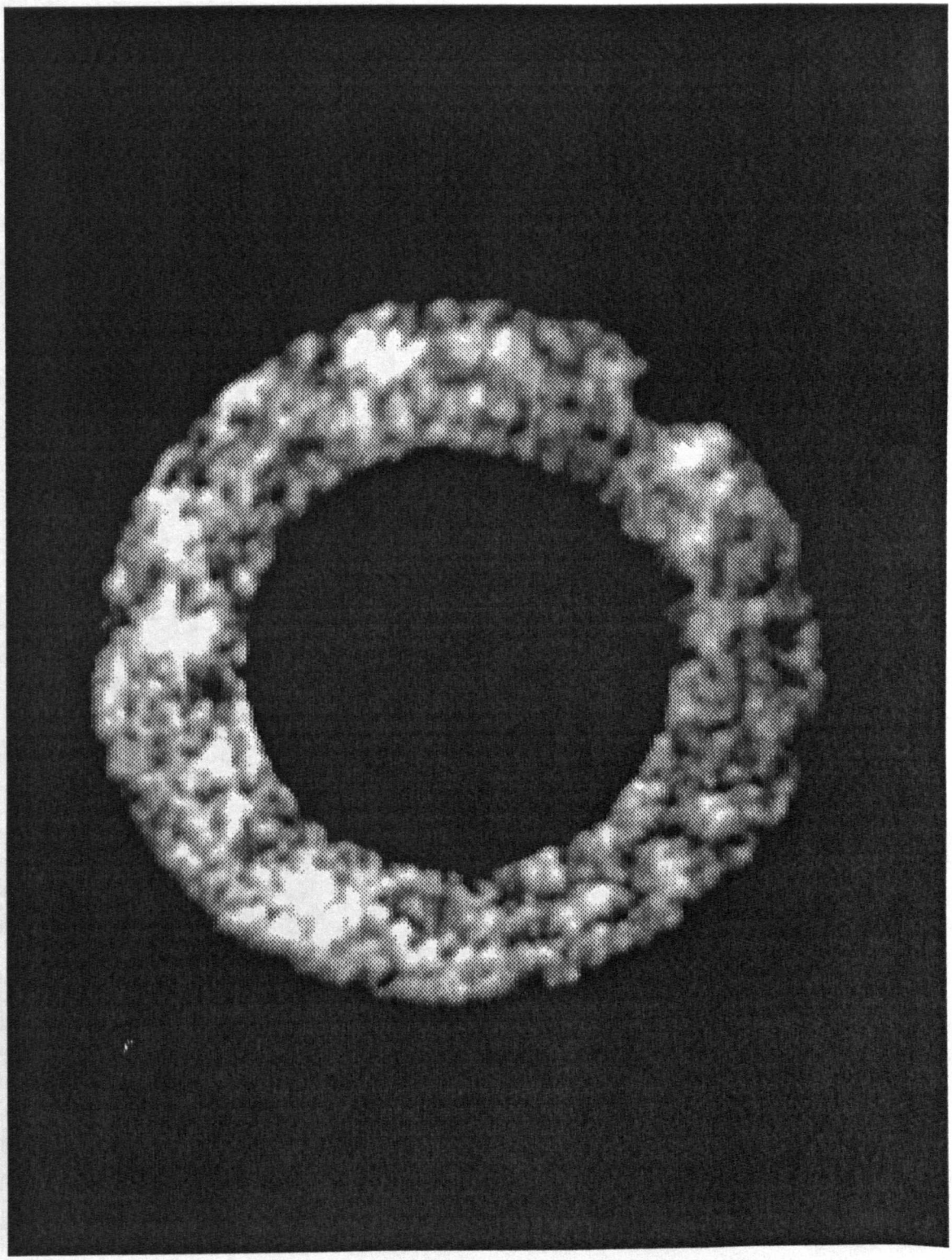


Figure 6.12: FERRITE SAMPLE CONTAINING EDGE CHIP

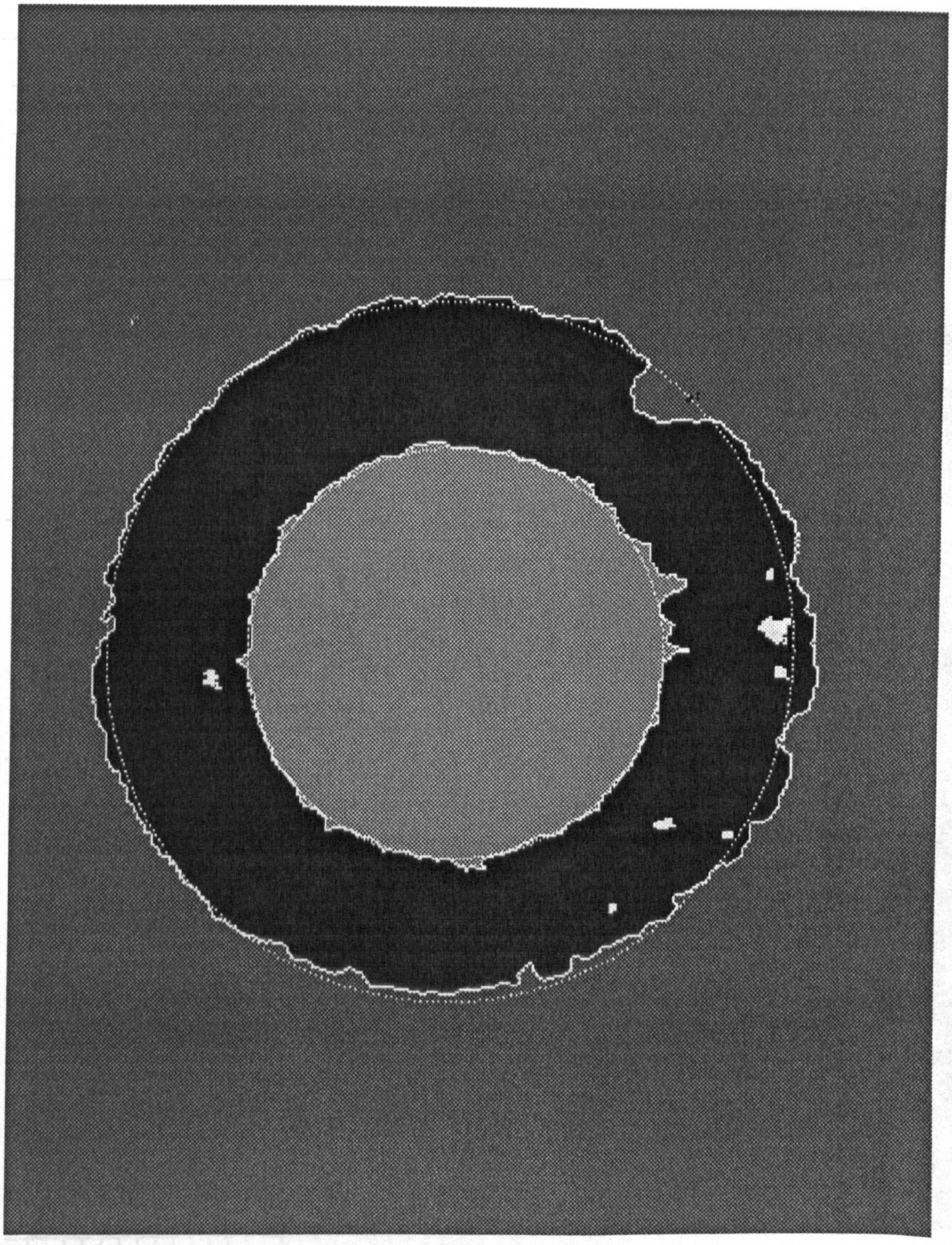


Figure 6.13: RESULT OF PROCESSING IMAGE SHOWN IN FIG. 6.12

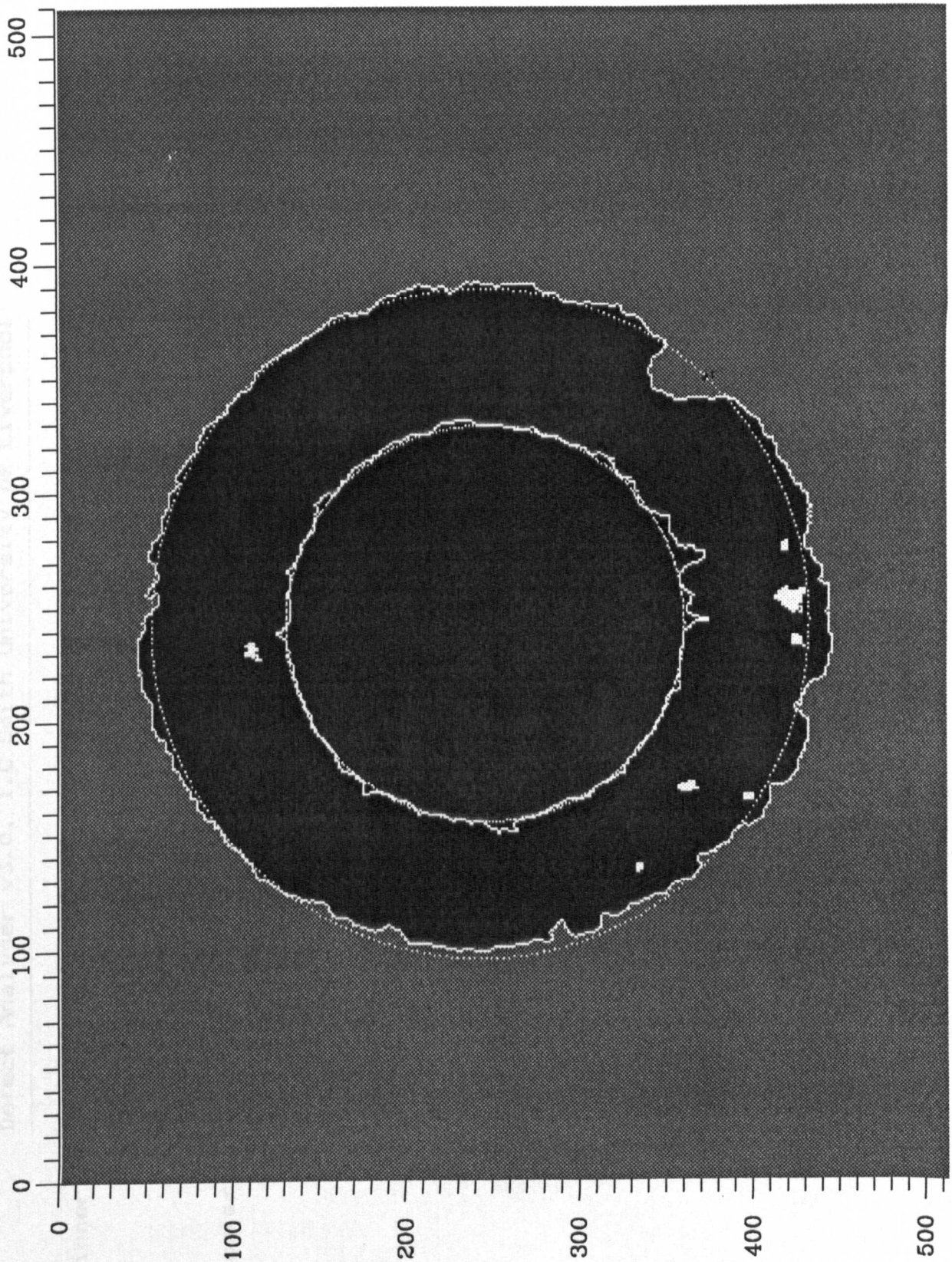


Figure 6.14: IMAGE SHOWN IN FIG. 6.13 AFTER GREY SCALE REMAPPING

```
        Defect Analyser v3.0, I.C Smith University of Liverpool
        -------------------------------------------------------

inner edge parameters:-
        centre at 242,245
        major axis length = 115
        minor axis length = 115
        major axis rotation = -28 degrees

outer edge parameters:-
        centre at 243,242
        major axis length = 196
        minor axis length = 190
        major axis rotation = 3 degrees


        offset between centres = 3
        total chip area = 460 = 0.8 % of disc area


      all dimensions in pixels normalised w.r.t aspect ratio
                    press any key to continue
```

Figure 6.15: ELLIPSE RESULTS FOR IMAGE SHOWN IN FIG. 6.13

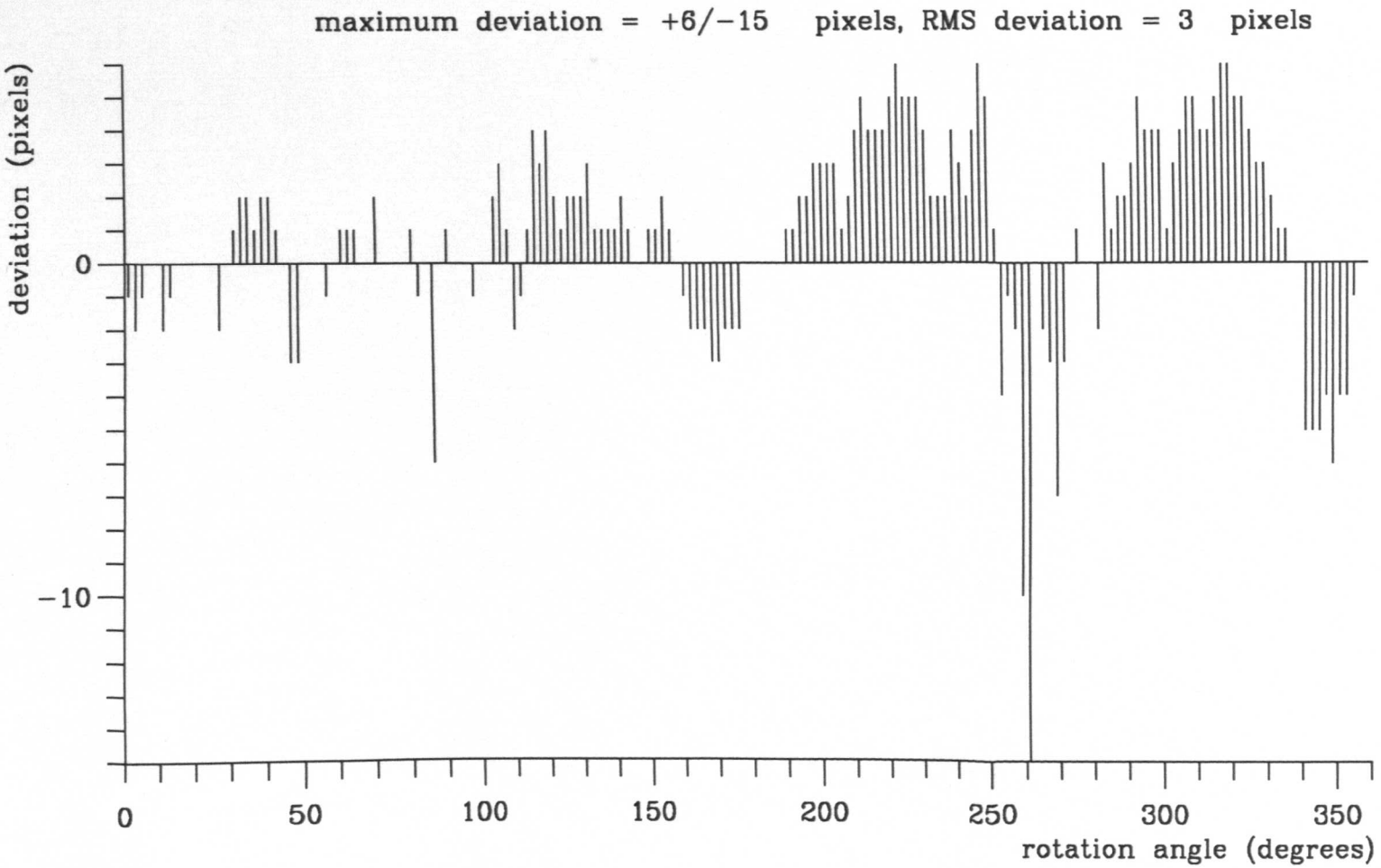


Figure 6.16: INNER EDGE PROFILE FOR IMAGE SHOWN IN FIG. 6.13

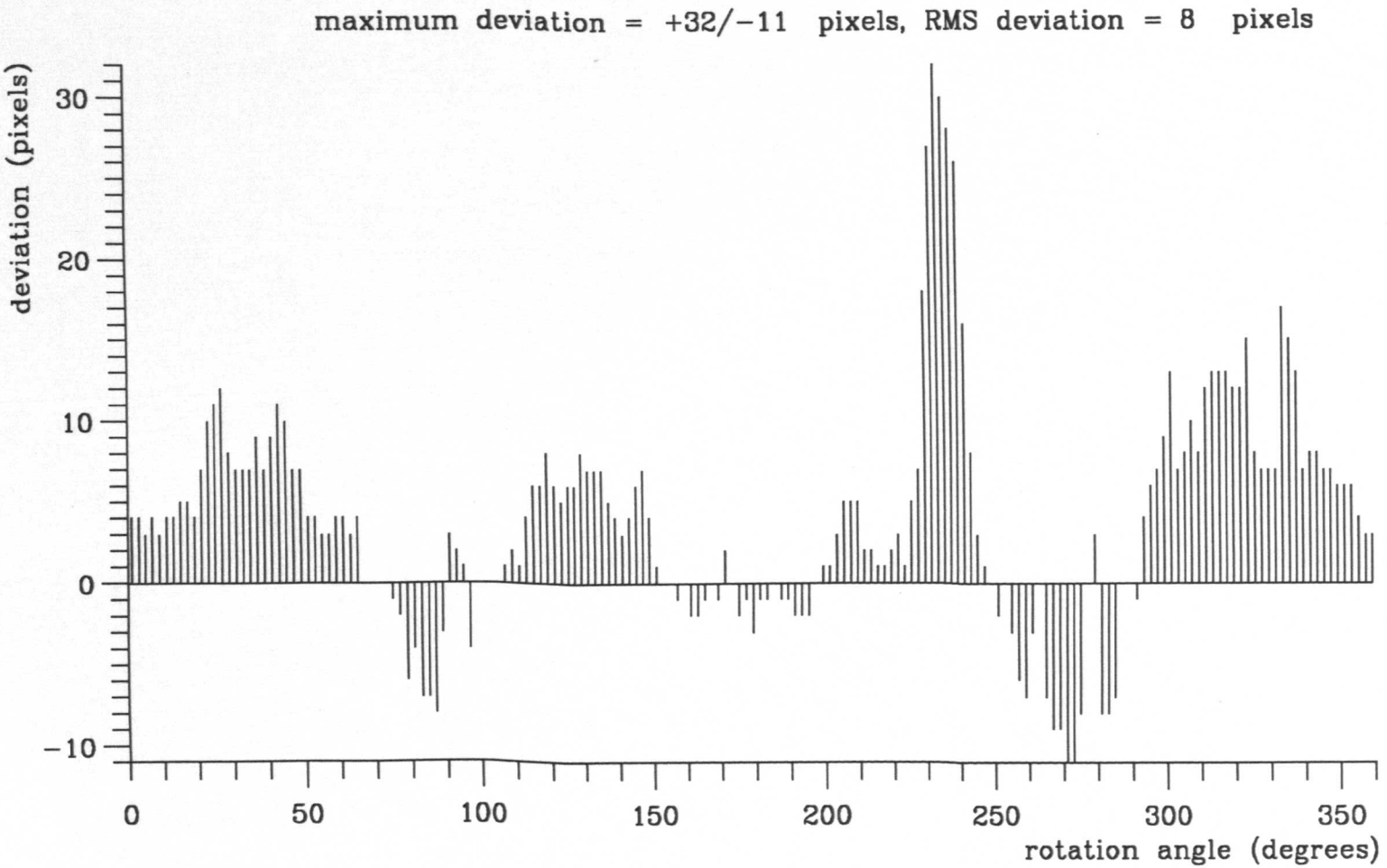


Figure 6.17: OUTER EDGE PROFILE FOR IMAGE SHOWN IN FIG. 6.13

Defect Analyser v3.0, I.C Smith University of Liverpool

chip parameters:-

area	perimeter	P*P/A	position	max r	min r	max / min
102	35	12	231,109	8	1	8
27	10	3	134,133	3	1	3
78	26	8	169,361	6	1	6
29	12	4	165,396	3	1	3
242	56	12	251,420	11	4	2
29	12	4	274,417	4	1	4
58	22	8	233,424	5	1	5

all dimensions in pixels normalised w.r.t aspect ratio
press any key to continue

Figure 6.18: DEFECT RESULTS FOR IMAGE SHOWN IN FIG. 6.13

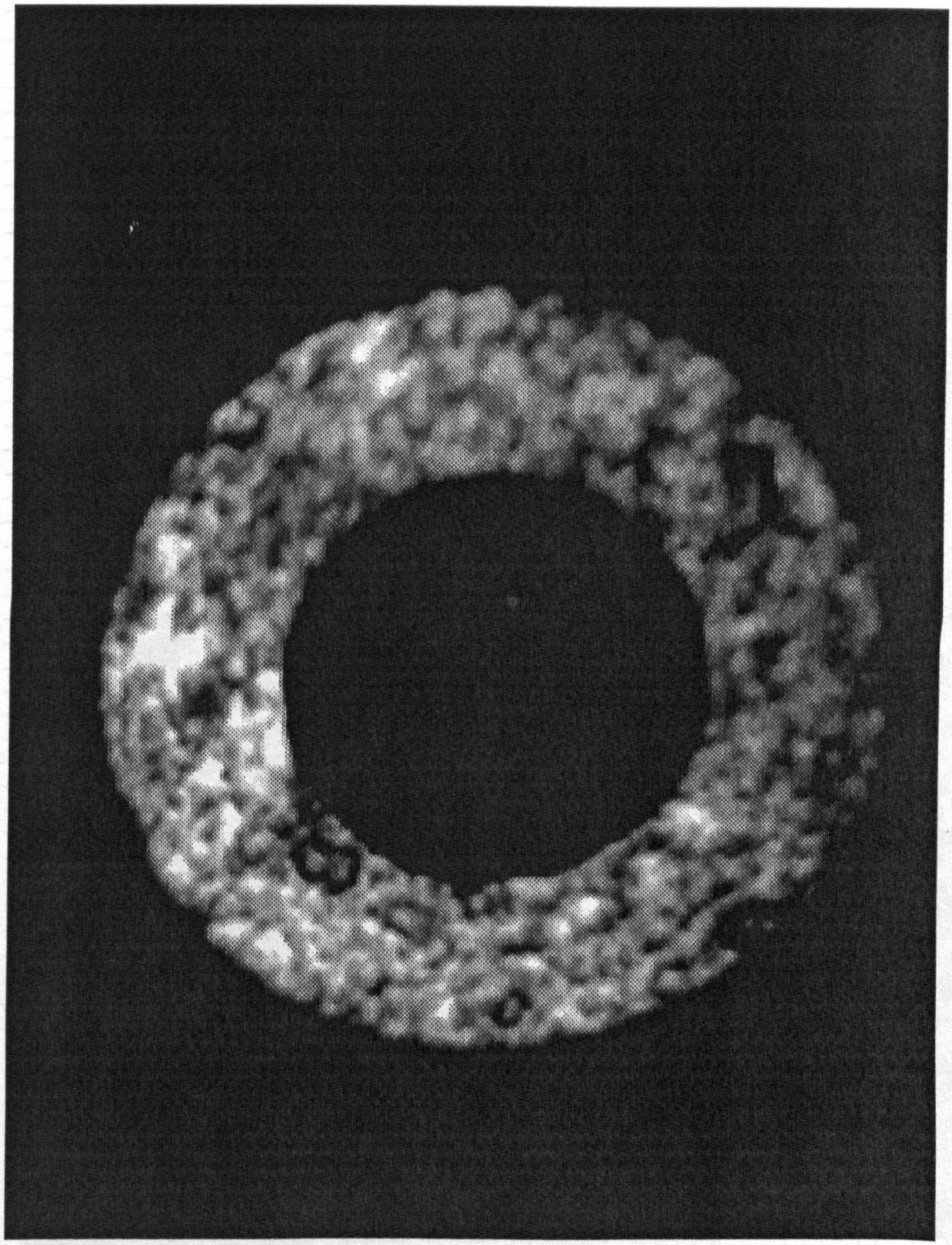


Figure 6.19: FERRITE SAMPLE CONTAINING CHIPS AND CRACKS

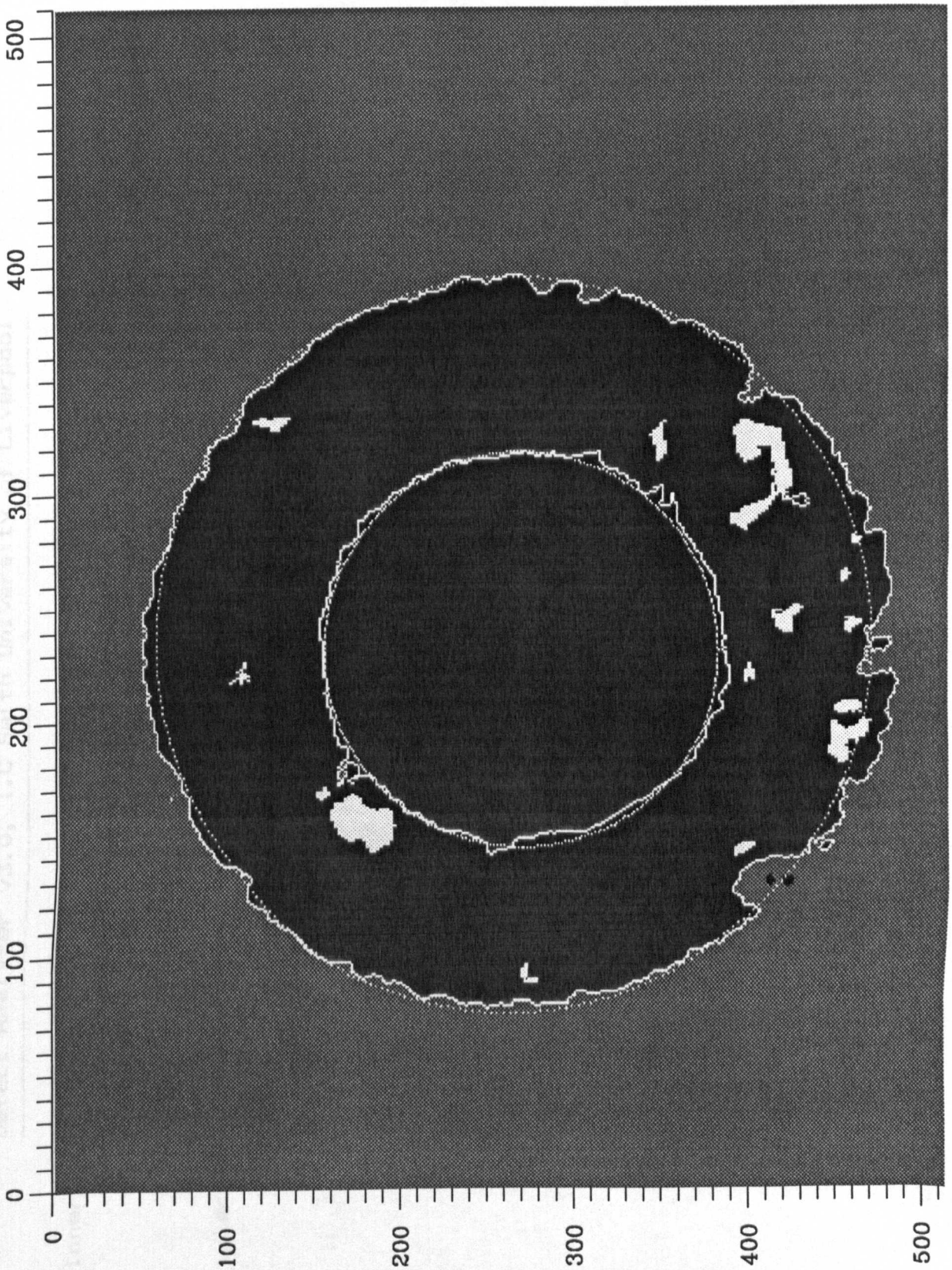


Figure 6.20: RESULT OF PROCESSING IMAGE SHOWN IN FIG. 6.19

```
        Defect Analyser v3.0, I.C Smith University of Liverpool
        -------------------------------------------------------

inner edge parameters:-
          centre at 231,266
          major axis length = 116
          minor axis length = 113
          major axis rotation = -20 degrees

outer edge parameters:-
          centre at 235,262
          major axis length = 212
          minor axis length = 206
          major axis rotation = -17 degrees


          offset between centres = 5
          total chip area = 2266 = 3.2 % of disc area


        all dimensions in pixels normalised w.r.t aspect ratio
                      press any key to continue
```

Figure 6.21: ELLIPSE RESULTS FOR IMAGE SHOWN IN FIG. 6.20

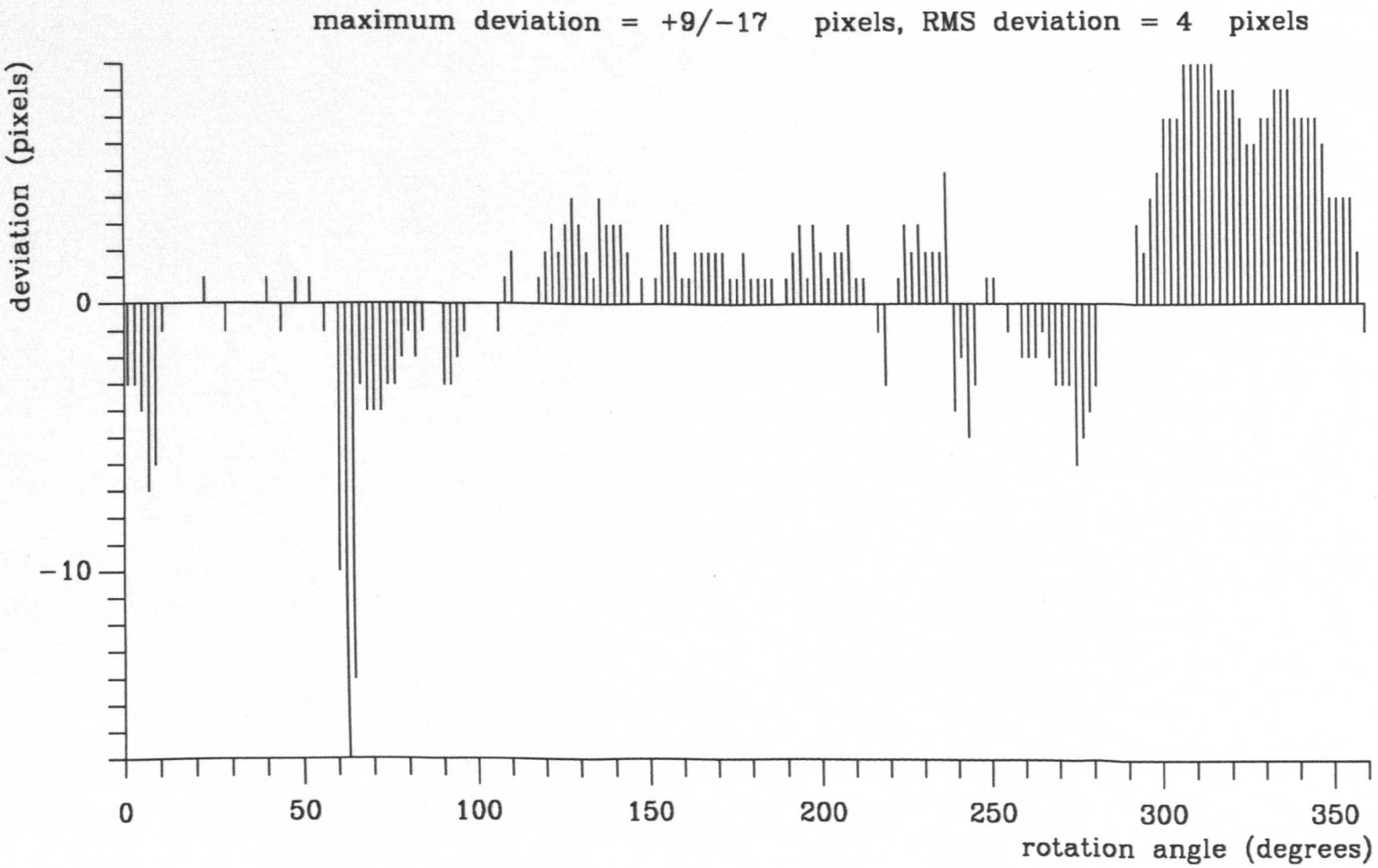


Figure 6.22: INNER EDGE PROFILE FOR IMAGE SHOWN IN FIG. 6.20

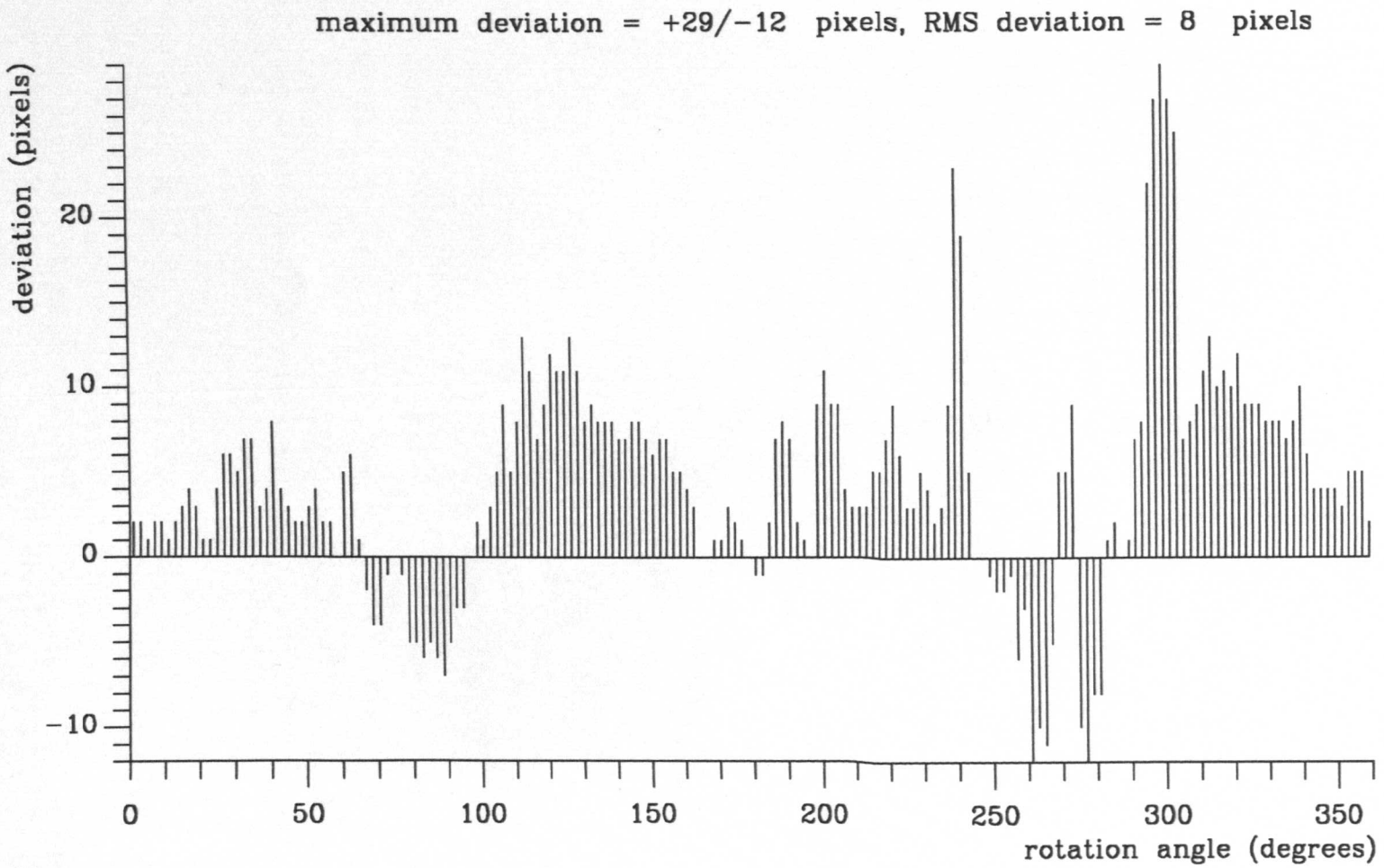


Figure 6.23: OUTER EDGE PROFILE FOR IMAGE SHOWN IN FIG. 6.20

Defect Analyser v3.0, I.C Smith University of Liverpool

chip parameters:-

area	perimeter	P*P/A	position	max r	min r	max / min
129	51	20	219,105	9	1	9
146	48	15	330,123	11	1	11
65	20	6	167,151	4	1	4
900	119	15	154,174	19	8	2
63	29	13	91,269	8	1	8
171	55	17	321,345	13	1	13
700	277	109	289,396	56	3	18
83	27	8	142,392	8	1	8
694	159	36	318,405	28	3	9
69	26	9	220,396	6	1	6

all dimensions in pixels normalised w.r.t aspect ratio
press any key to continue

continued...

Figure 6.24: DEFECT RESULTS FOR IMAGE SHOWN IN FIG 6.20

Defect Analyser v3.0, I.C Smith University of Liverpool

chip parameters:-

area	perimeter	P*P/A	position	max r	min r	max / min
206	60	17	244,418	11	3	3
440	102	23	190,451	19	2	9
148	42	11	204,453	8	1	8
47	17	6	262,451	4	1	4
89	29	9	241,455	6	1	6
30	13	5	278,457	4	1	4

all dimensions in pixels normalised w.r.t aspect ratio
press any key to continue

Figure 6.25: DEFECT RESULTS FOR IMAGE SHOWN IN FIG. 6.20

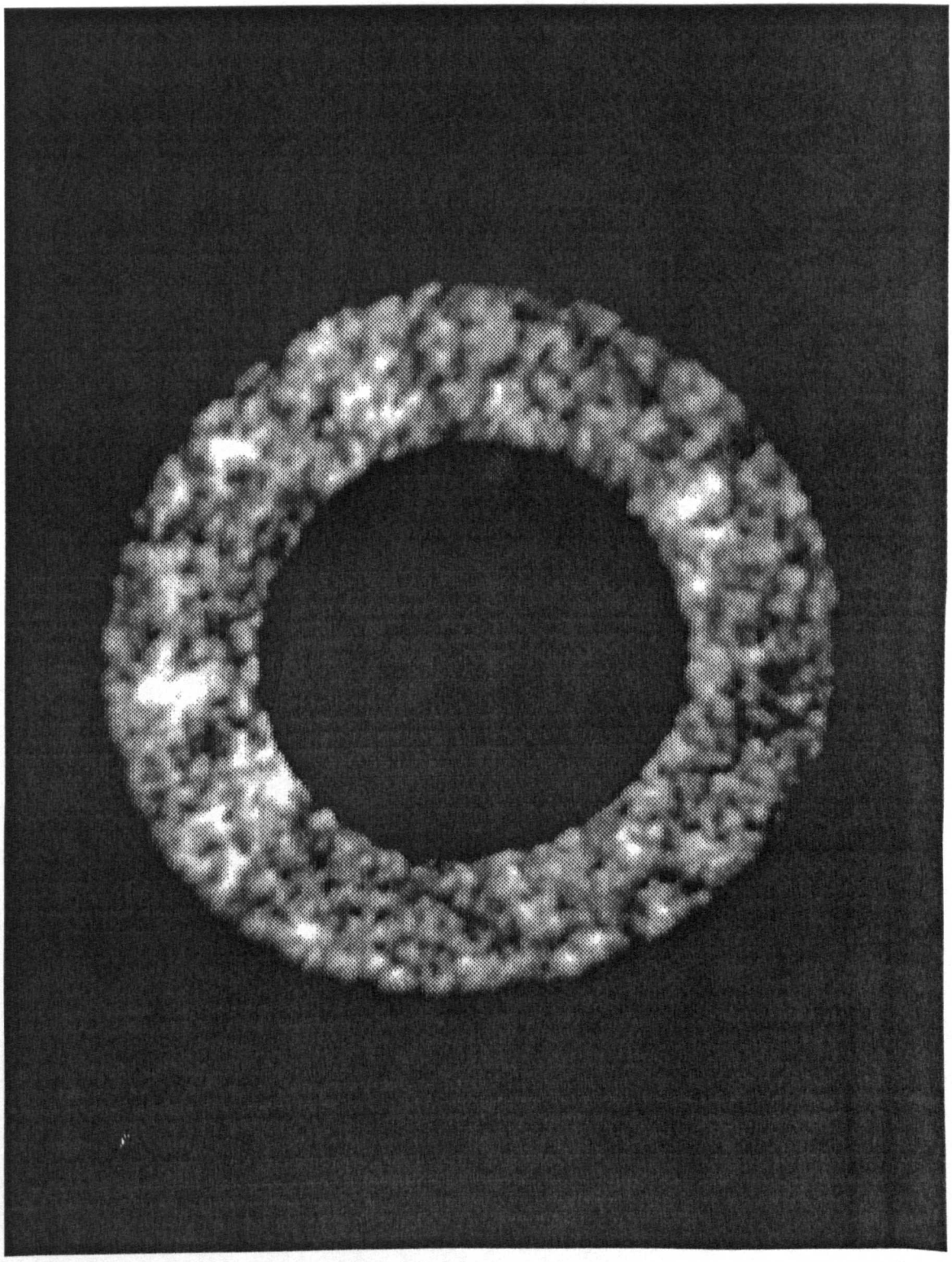


Figure 6.26: FERRITE SAMPLE CONTAINING ACCEPTABLE DEFECTS

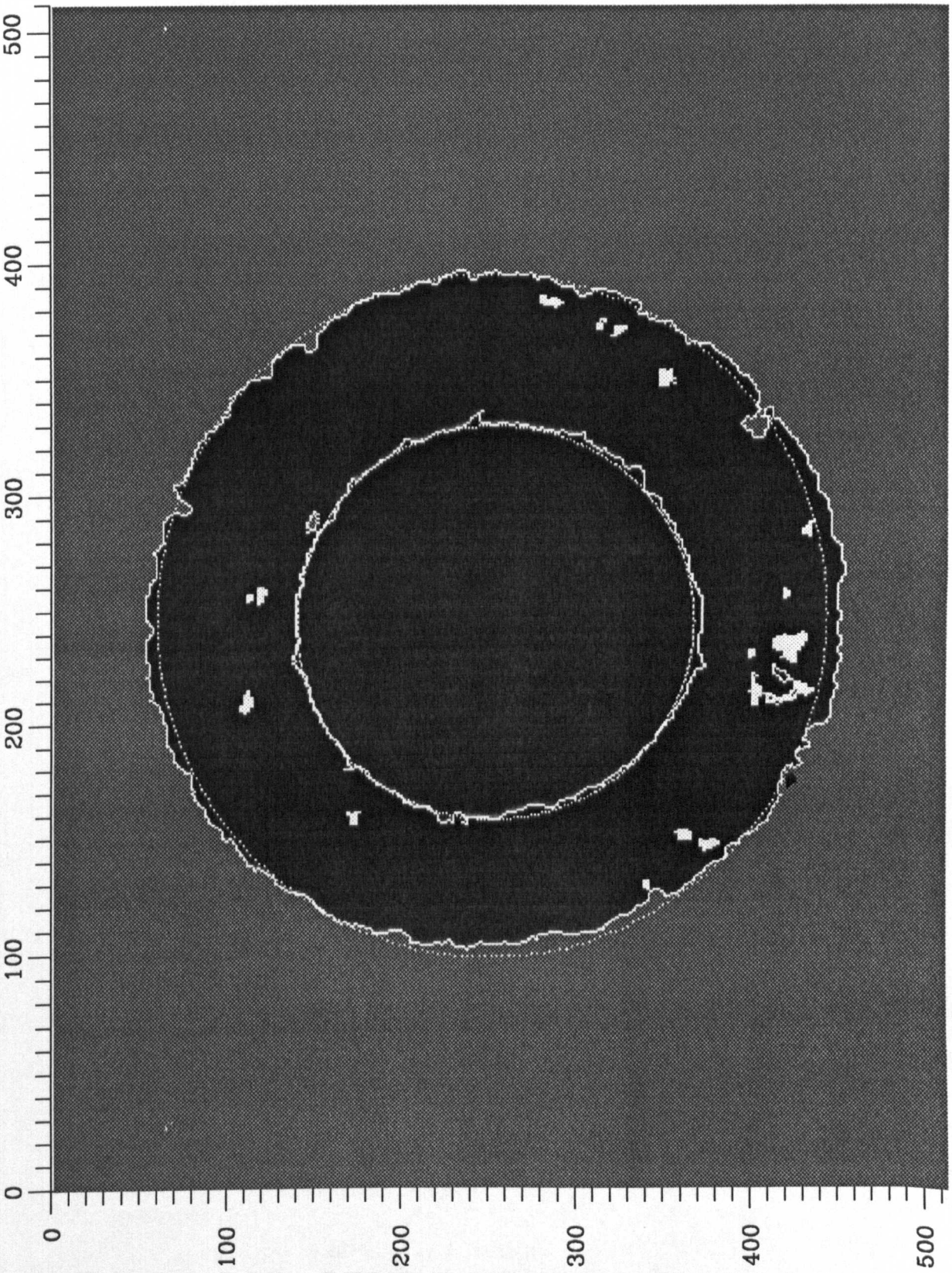


Figure 6.27: RESULT OF PROCESSING IMAGE SHOWN IN FIG. 6.26

```
        Defect Analyser v3.0, I.C Smith University of Liverpool
        -------------------------------------------------------

inner edge parameters:-
        centre at 244,253
        major axis length = 115
        minor axis length = 114
        major axis rotation = -30 degrees

outer edge parameters:-
        centre at 247,251
        major axis length = 198
        minor axis length = 192
        major axis rotation = -23 degrees


        offset between centres = 3
        total chip area = 1172 = 2.0 % of disc area


      all dimensions in pixels normalised w.r.t aspect ratio
                  press any key to continue
```

Figure 6.28: ELLIPSE RESULTS FOR IMAGE SHOWN IN FIG. 6.27

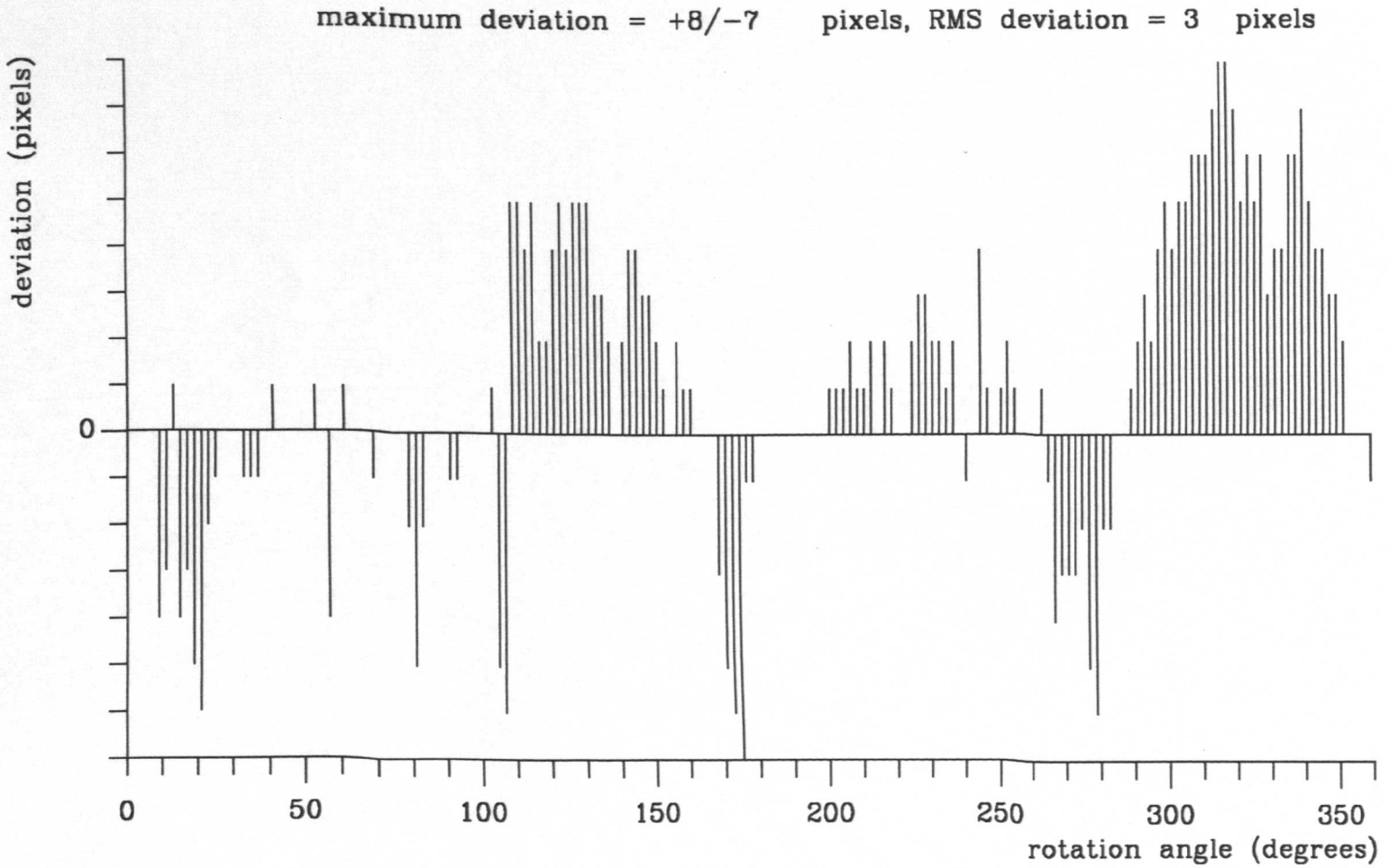


Figure 6.29: INNER EDGE PROFILE FOR IMAGE SHOWN IN FIG. 27

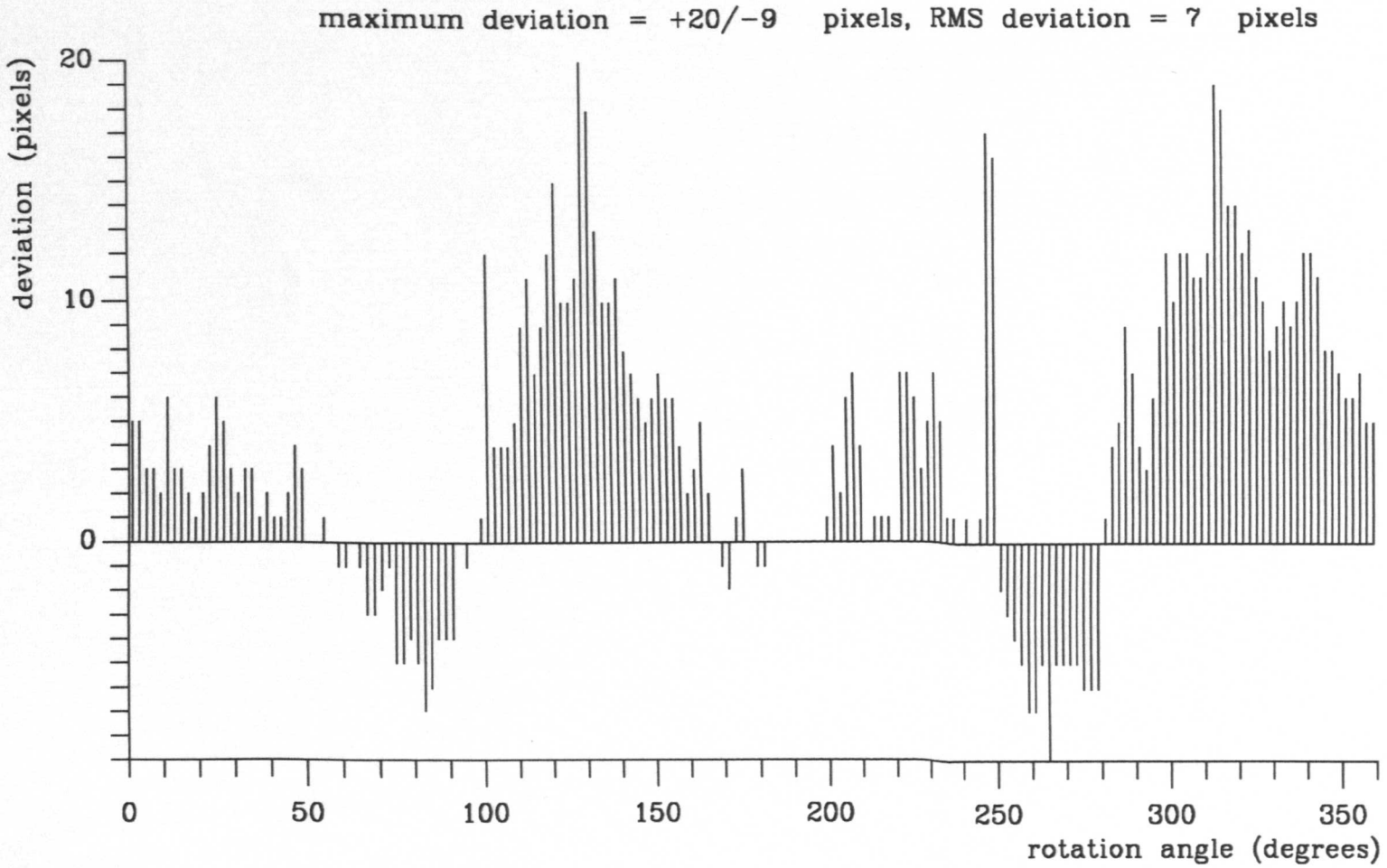


Figure 6.30: OUTER EDGE PROFILE FOR IMAGE SHOWN IN FIG. 27

Defect Analyser v3.0, I.C Smith University of Liverpool

chip parameters:-

area	perimeter	P*P/A	position	max r	min r	max / min
102	31	9	209,109	8	1	8
27	10	3	255,111	3	1	3
66	23	8	257,117	5	1	5
55	23	9	158,171	5	1	5
103	33	10	383,283	8	1	8
37	15	6	373,311	5	1	5
61	23	8	371,322	5	1	5
33	13	5	128,339	3	1	3
118	38	12	350,349	6	3	2
70	23	7	150,361	5	2	2

all dimensions in pixels normalised w.r.t aspect ratio
press any key to continue

continued...

Figure 6.31: DEFECT PARAMETERS FOR IMAGE SHOWN IN FIG. 6.27

Defect Analyser v3.0, I.C Smith University of Liverpool

chip parameters:-

area	perimeter	P*P/A	position	max r	min r	max / min
94	29	8	145,375	6	1	6
27	10	3	229,399	3	1	3
377	164	71	212,402	33	2	16
288	64	14	233,421	12	4	3
30	11	4	256,419	2	1	2
139	62	27	214,428	20	1	20
58	23	9	283,431	6	1	6

all dimensions in pixels normalised w.r.t aspect ratio
press any key to continue

continued...

Figure 6.32: DEFECT PARAMETERS FOR IMAGE SHOWN IN FIG. 6.27

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 INTRODUCTION

The previous chapters of this work have described the development of vision based systems for NDT applications and typical results from the systems have been presented. In all cases the systems have been shown to provide results which are at least comparable with those of manual tests. The systems are, at present, only semi-automatic in that samples have to be loaded and unloaded manually by the operator. In this chapter improvements aimed at providing full automation are suggested.

The use of a general purpose personal computer has kept the cost of each system, less than £5 000, to a minimum however this has been at the expense of processing speed. Typical cycle times are, approximately, 15 s for the Shore system, 30 s for the Vickers system and 90 s for the quality assessment system. Much faster operation can be achieved using a transputer based system and it is expected that any future work will advance in this direction. In all of the systems it has been necessary to make a

compromise between accuracy and processing speed and in this chapter image analysis algorithms are discussed which should give improvements in accuracy but were too slow to be used in the present systems.

7.2 SHORE HARDNESS TEST

The Shore system has been shown to provide accurate results over the range in which the test is linear. If the system was required to produce results which are directly comparable to a commercially available Shore Scleroscope, measurements would have to be made using the Scleroscope and compared to those obtained from the vision system to give a non-linear calibration curve. The calibration data could then be stored as a look-up table in the host computer.

The accuracy of the system is determined directly by the frame store resolution and capture rate; the latter being the most important at present. The faster the capture rate, the higher the effective sampling rate and the more faithful the indenter record path produced by the system will be to the actual indenter motion. The worst case error occurs when a frame is captured half a frame period before the top of the indenter trajectory and again half a frame period after the top of the trajectory. In this case the top of the trajectory will not be present in the path record but instead two "snap shots" taken either side of this point. At present the frame capture rate is limited by the use of a standard closed-circuit television (CCTV) video camera to the CCTV standard of 25 frames/sec. However, video cameras are available in which the frame capture rate can be set arbitrarily by the user. In particular, charge-coupled device (CCD) cameras

are available with fast, externally controlled, electronic shutters. Another device, known as an image dissector, allows for complete control of scanning rate and scanning path.

In the present system only four rows of image data are extracted from the frame store for each captured frame, so that most of the frame store image memory is actually superfluous. Since the cost of frame stores is dictated mainly by their resolution, and hence memory size, the present system is clearly not cost effective. Instead of using a commercially available frame store, a circuit may be designed to digitise and store image data from only a few given lines of each video signal field. Since the amount of data to be stored in each frame would be significantly reduced it may be possible to write the image data directly to the host computer base memory allowing faster access for the microprocessor. Using this set up, real time analysis may be possible giving faster execution speed.

Similar considerations also apply to the video camera. Since it is only the medial axis of the guide tube which is of interest in the scene viewed by the video camera use may be made of a line scan camera focused on the centre of the guide tube. The line scan camera is likely to be cheaper than a conventional CCTV camera and may provide greater spatial resolution.

In the present system, the operator is required to release the indenter manually and then press a key on the computer to start the analysis program. Clearly this process is not suited to an automated system and a means of drawing the indenter back up the guide tube after each measurement and holding it at the 250 mm position is required. One solution to this may be to apply a vacuum to the top of the guide tube to raise the indenter. A fine wire gauze can be imbedded in the tube at the 250 mm point so that the indenter is held at this position. An electrically operated valve may then be used to

shut off the vacuum so that the indenter is released to begin a test. Automation can then be achieved by interfacing the vacuum valve to the host computer to allow for software control.

It should be noted that the vision system described in this work provides just one means of automating the Shore hardness test. One alternative is to place a photo-detector array along the side of the guide tube with the indenter again illuminated with laser light so that the incident light is scattered onto the detector array. By scanning the array for the detector having the largest output the position of the indenter may be deduced. Another alternative is to calculate the time of flight for a pulse of light to be transmitted by the laser and reflected, by the indenter, back to the source. Since the speed of light is known, the distance from the laser to the indenter can be calculated and position of the indenter at any given instant determined.

7.3 VICKERS HARDNESS TEST

As indicated in Chapter 6, the precision of the Vickers system is determined by the ability of the analysis software to accurately locate the edges of the indentation. These do not appear as sharp black to white transitions but as gradual changes from black through grey to white taking place over several pixels. These grey areas reduce the image contrast and thus the reliability of the automatic thresholding process leading to uncertainty in the exact position of indentation edges.

To increase the precision of the system, a more sophisticated threshold calculation algorithm is required to take account of edge blur; one such method is to analyse only

the grey levels of pixels lying close to the object/background interface. The grey level histogram for these pixels should contain two narrow, symmetrical, peaks separated by a sharp well defined "valley" which is ideally suited to automatic determination of thresholds. The shape of the histogram can be explained by the fact that in a given window containing the border, the frequency of light (background) pixels will be high as will the frequency of dark (object) pixels, since the probability of a pixel belonging to the object is equal to that of it belonging to the background. This gives rise to the two sharp peaks in the histogram. The frequency of pixels with any given intermediate grey level will be low, since there is a gradual change from light to dark at the object/background border; this gives rise to the sharp valley. A problem with this approach is that the position of the object/background border is not known prior to segmentation however, the location of border pixels can be found from consideration of the digital gradient of the image.

As indicated in section 2.4.3, the magnitude of the first order derivative of image intensity is high at points marking the object/background boundary (these points being referred to as edges) but is low for homogeneous regions. Thus, by choosing a suitable threshold for the gradient value, a decision can be made on which pixels are suitable for histogram analysis.

An extension to this scheme makes use of the Laplace operator (second order derivative of image intensity) in conjunction with the first order gradient operator to give a tri-level image. The image is segmented according to:-

$$s(x,y) = \begin{cases} 0, & \text{for } G[f(x,y)] < T \\ +, & \text{for } G[f(x,y)] \geq T \text{ and } L[f(x,y)] \geq 0 \\ -, & \text{for } G[f(x,y)] \geq T \text{ and } L[f(x,y)] < 0 \end{cases}$$

where s(x,y) is the value assigned to the tri-level image at position (x,y) and f(x,y) is the original image intensity at (x,y). G represents the value of the digital gradient at (x,y) and L the output of the Laplace operator at (x,y) whilst T is a fixed threshold.

In the case of a dark object on a bright background the Laplace operator output changes from a large positive value to a large negative value at the object/background border (see section 2.4.3) so that a given scan line in the tri-level image has the following structure:-

$$(...)\,(-,+)\,(0 \text{ or } +\,)\,(+,-)\,(...)$$

where (...) represents any combination of +,-, or 0. The innermost parenthesis contain the object pixels which can easily be identified by noting the location of any sequence (0 or +) bracketed by (-,+) and (+,-). Thus a clear delineation can be made between object and background pixels.

The use of a more sophisticated thresholding algorithm in the analysis software would lead to an additional improvement in analysing small indentations. At present, when an image of a small indentation is analysed, the mode corresponding to the object in the grey level histogram is small in comparison to that of the background and may be "swamped" by modes corresponding to surface blemishes etc. In this case, erroneous threshold limits may be produced by the threshold calculation algorithm, leading to rejection of the component or recourse to region labelling (which is time consuming). However, the more powerful thresholding algorithms described in this chapter are insensitive to changes in object size and therefore do not give rise to these problems. A disadvantage with using these algorithms is the need to compute the digital gradient

of the entire image before thresholding can be attempted. This is a time consuming operation which may add 30 seconds to the cycle time of the present system.

One drawback of the present system is the need for samples to be placed manually under the microscope and the microscope then focused by the operator. To see how full automation can be achieved, it first is necessary to examine the operation of a conventional Vickers test machine. Here the sample is placed under the indenter and a lever pressed on the machine to apply the indenter load (which is set by the operator adding weights to a hanger on the back of the machine); after approximately 10 seconds the load is automatically removed and the user is required to turn a handle, controlling the table position, to locate the sample under the microscope. The microscope is then focused by the operator and a reading taken.

Control of the load application could easily be achieved by the use of an actuator interfaced to the host computer whilst either a stepping motor or servo motor, again interfaced to the computer, could be used to control the lateral movement of the table and sample. As a refinement, two drives could be used allowing the table to be moved in both the X and Y directions so that a series of indentations can be made over the entire area of the sample. The indentations could then be placed one by one under the microscope for analysis so that a complete profile of the surface hardness is developed. This would be particularly useful in micro-indentation hardness testing. The problem of altering the load applied to the indenter is the most difficult one; the most obvious solution would be to use a robot to load and unload the weights however, this seems hardly cost effective.

In order that different sized samples can be analysed automatically some means of adaptively focusing the microscope is required. The microscope focus can be varied by

altering the height of the lens above the table using a stepping or servo motor under the control of the host computer. A sensor is also required to determine when the camera image is in focus. One method may be to use an ultrasonic or optical range-finder to measure the distance between the lens and the sample surface. If the focal length of the lens is known, then the lens can be positioned at the correct height. Another method may be to vary the height of the lens and analyse the degree of image blur at each position using image processing so that the focal point can be found. The actual image processing required is at present uncertain however, one method may be to calculate the sum of the digital gradient values over the entire image for each captured frame. When the image is in focus, there should be sharp contrast between the indentation and background leading to large gradient values. The sum of the gradient values should then be at a peak at the focal position.

Further software will have to be written to give a fully automated system. This will need to include control algorithms for the drive motors and actuators in addition to software for analysing sensor data. The usefulness of the system may also be improved by the addition of software giving hard copy of results, statistical data for groups of measurements and data archiving. A useful facility would also be to provide a means of allowing the user to specify the number and position of measurements to be taken on any given sample.

7.4 QUALITY ASSESSMENT SYSTEM

Currently, the results presented by the quality assessment system are in the form of pixels and not physical units. For practical applications calibration is therefore

necessary. One method would be to calculate the size (inner and outer edge radius) of several samples and compare the average value in pixels to the average value as measured by a micrometer. The accuracy of this method is determined by both the number of samples used and the micrometer accuracy and a better method may be to use a calibration block of very precisely known dimensions. If the calibration block is of the same thickness as a typical sample it can be analysed directly using the vision system, since the camera will still be in focus and no alteration of the optical hardware is necessary. The size of the block, in pixels, can be measured using two, software generated, cursors (as in the Vickers system) so that the camera scale factors can be measured. Any calibration procedure should take account of geometric distortion within the camera lenses.

The present system if capable of analysing defects appearing on the flat surfaces of a component but not on the rim. By placing the component on a spindle, driven by motor, a series of images of the rim can be captured as the component is rotated. Two types of defect need to be analysed; non parallel faces on the component and rim cracks. In the first case any non parallelism is shown up by variations in rim thickness as the component is rotated. The thickness of the rim at any point may be calculated by analysing a scan line in the image which is co-linear with the spindle axis and counting the number of object pixels in the scan line. Since only one row of data is extracted from each captured frame a complete profile of the rim may be developed in the same manner as the path record in the Shore system.

Fully penetrant rim cracks may be detected in a similar manner to that used in section 5.3.4. If the disc does not contain a crack the image will contain only one large bright region corresponding to the image of the rim whilst if a crack is present then this region will be split into two by the image of the crack. In the former case border tracing leads

to only one large contour while in the latter case two non-intersecting contours are formed. This condition could easily be detected by the image analysis software.

As indicated in Chapter 6, the distinction between chip and crack features is somewhat arbitrary at present. In order to give a more meaningful description of defects it is necessary to incorporate more "intelligence" into the system so that the present software acts merely as a front end processor. This "intelligence" can take one of two forms, a neural network or an expert system.

In the former case the network is supplied with the outline of a defect as input and given a description of the defect (e.g. "chip" or "crack"). After analysing many samples the neural network "learns" to identify the different defects and can distinguish between them. The length of the learning period needed is determined by the complexity of the input data (in this case the amount of data needed to describe the outline of the crack) and the complexity of the network (degree of interconnection). A disadvantage of this approach is that there is no way of determining how the neural network arrived at a decision so that the "intelligence" cannot easily transferred to an incompatible system.

In contrast, expert systems rely on a complex set of rules, held in data base, to arrive at a decision. The rules are arrived at by the examination of data supplied by real experts in the field. This may take the form of physical laws or rule of thumb decisions based on past experience. Since the rules are well defined it relatively easy to transfer the "intelligence" to other systems.

An expert system for analysing weld cracks has been developed by the Welding Institute [58]. By responding to questions posed by the system the user supplies information about the crack. The system responds by indicating which cracking mode is most likely

to have induced the problem. This type of output is common to all expert systems; instead of supplying the user with a "hard" decision based on the input data the results are presented as a series of possible judgements, each with it's own probability. As the system is provided with more (correct) information the likelihood of a given judgement increases whilst the likelihood of other judgements diminishes so that a "soft" decision can be arrived at. In this application it will be necessary to calculate many more shape parameters for defects so that these may be input into the expert system. A series of rules governing which parameter values lead to which defects is also necessary to form the knowledge base of the system.

REFERENCES

[1] Halmshaw R., Non-Destructive Testing, Edward Arnold, 1987.

[2] Halmshaw R., Industrial Radiology: Theory and Practise, Applied Science, 1982.

[3] Handbook on the Ultrasonic Examination of Welds, The International Institute of Welding, 1977.

[4] Coffey J., "Non-Destructive Testing - the Technology of Measuring Defects", *CEGB Research*, pp 36-42, 1988.

[5] Hochschild R., *Prog in NDT*, vol. 1, pp 57-110, 1958.

[6] Magnaflux Ltd., "NDT Equipment and Materials", South Dorcan Estate, Swindon, Wiltshire SN3 5H3, UK.

[7] Dorothy R.G., "Social and Economic Considerations of Automated Inspection", *Proc 5th Int Conf on Automated Inspection and Product Control*, Stuttgart, 1980.

[8] Morris E. and Lucas J., "Microcomputer Controlled Robotic Equipment for Precision TIG Welding", *Met. Constr. Trans.*, vol. 9, pp 81-94, 1987.

[9] Lucas W., "TIG and Plasma Welding in the 80's", *Met. Constr. Trans.*, vol. 14, pp 592-9, 1982.

[10] Fenn R., "Sensors Present and Future", *Weld. Met. Fabr.*, pp 313-5, 1984.

[11] Ho S.K., White R.M. and Lucas J., "A Vision System for Automated Crack Detection in Welds", *Meas. and Sci. Technol.*, vol. 1, no. 3, 1990.

[12] Tabor D., "The Hardness of Metals", Chap 8, Oxford University Press, 1984.

[13] O'Neil H., Hardness Measurement of Metals and Alloys", Chap 2, Chapman and Hall, 1967.

[14] Batchelor G.G., Marlow B.K., Smith D.V. and Werson M.J., "A Research Laboratory for Automated Visual Inspection", *Proc. 5th Int. Conf. on Automated Inspection and Product Control*, Stuttgart, 1980.

[15] Mohs F., Grundiss der Mineralogie, Dresden 1822.

[16] Shore A.F., *J. Iron and Steel*, vol. 2, pp 59- , 1918.

[17] Brinell J.A., *2nd Cong Int Methodes d'Essai*, Paris 1900.

[18] Whalberg A., *J. Iron and Steel Inst.*, vol. 59, pp 425-, 1918.

[19] Smith R. and Sanderland G., *J. Iron and Steel*, vol. 2, pp 59- ,1918.

[20] Rockwell, S.R., *Trans. Amer. Soc. for Steel Treating*, vol. 2, pp 1023-, 1922.

[21] British Standards Institute, BS427 Part 1: Vickers Hardness Test - Testing Of Metals, 1961.

[22] British Standards Institute, BS427 Part 2: Vickers Hardness Test - Verification, 1962.

[23] Gifkins R.C., Optical Microscopy of Materials, Pitman 1970.

[24] Rosenfeld A. and Kak A.C., Digital Picture Processing, Academic Press, 1976.

[25] Woods R.E. and Gonzalez R. C., "Real Time Digital Image Enhancement", *Proc IEEE*, vol. 69, no. 5, pp 643-645, 1981.

[26] "Photographic Image Enhancement by Superposition of Multiple Images", *Photogr. Sci. Eng.*, vol. 7, no. 4, pp 241-5, 1963.

[27] Huang T.S. and Yang G.Y., "A Fast 2D Median Filtering Algorithm", *IEEE Trans. Acoust. Speech and Signal Process.*, vol. ASSP-27, pp 13-18, 1979.

[28] Unger S.H., "Pattern Detection and Recognition", *Proc. IRE*, vol. 47, no. 10, pp 1737-1752.

[29]Sklansky J., "Image Segmentation and Feature Extraction", *IEEE Trans. Syst Man. Cybern*, vol. SMC-8, no. 4, April, 1978.

[30] Brice C. and Fennema C., "Scene Analysis using Regions", *Artif. Intel.*, vol. 17, pp 285-348, 1970.

[31] Frei W., "Fast Boundary Detection: A Generalisation and a New Algorithm", *IEEE Trans Comput*, vol. C-26, no. 10 (October), 1977.

[32] Agin G.J., "Vision Systems", from Handbook of Industrial Robotics, edited by Nof S.Y., John Wiley and Sons, 1985.

[33] Freeman H., "Computer Processing of Line Drawings" , *Comput. Surv.*, vol. 6, pp 55-97, 1974.

[34] Nahim P.J., "The Theory and Measurement of a Silhouette Description for Image Processing and Recognition", *Pattern Recognition*, vol. 6, no. 2, pp 95-95, 1974.

[35] Kernighan B.W. and Ritchie D.M., The C Programming Language, Prentice Hall, 1978.

[36] ASM86 Language Reference Manual, Intel Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051, USA.

[37] Melles-Griot, 1 Fredrick St., Aldershot, Hants GU11 1LQ, UK.

[38] Hitatchi Densi UK, Ltd., Garrick Industrial Estate, Garrick Road, Hendon, London NW9 9AP, UK.

[39] Data Translation Inc., 100, Locke Drive, Marlboro, MA 01752-1192, USA.

[40] Opus PC Users Guide, Opus Technology Ltd., 53 Ormside Way, Holmethorpe Industrial Estate, Redhill, Surrey RH1 2LW, UK.

[41] Lattice C Compiler, Lattice Inc., P.O. Box 3148, Elen Ellyn, IL 60138, USA.

[42] MS-DOS Users' Guide and Users' Reference, Microsoft Corporation, 16011 NE 36th Way, Box 97017, WA 98073-9717, USA.

[43] Microsoft ASM86 Macro Assembler, Microsoft Corporation, 16011 NE 36th Way, Box 97017, WA 98073-9717, USA.

[44] Plink 86plus, Phoenix Technology Ltd., 320 Norwoord Park South, Norwood MA 02062, USA.

[45] Pfix 86plus, Phoenix Technology Ltd., 320 Norwood Park South, Norwood MA 02062, USA.

[46] Halo Users' Manual, Halo Associates Ltd., 1651 Third Avenue, New York NY 10128, USA.

[47] IBM Technical Reference Manual, IBM Personal Computer Sales and Service, P.O. Box 1328-C, Boca Raton, Florida 33432, USA.

[48] Pavlidis, T., Algorithms for Computer Graphics and Image Processing, Springer-Verlang, 1976.

[49] Fu K.S., Gonzalez R.C. and Lee C.S.G. Robotics, Control, Sensing, Vision and Intelligence, McGraw-Hill, 1976.

[50] Foith J.P., Eisenbarth C., Enderle E., Geisselmann H., Ringhauser H. and Zimmermann G., "Real-time Processing of Binary Images for Industrial Applications", from Digital Image Processing Systems, edited by Bolc L. and Kulpa Z., Springer-Verlang, 1981.

[51] Veillon F., "One Pass Computation of Morphological and Geometrical Properties of Objects in Digital Pictures", *Signal Process.*, vol. 1, 1977.

[52] Kruse B., "A Parallel Picture Processing Machine", *IEEE Trans. Comp.*, vol. C-22, no. 12, pp 1075-1087, 1977.

[53] Agrawala A.K. and Kulkarni A.V., "A Sequential Approach to the Extraction of Shape Features", *Computer Graphics and Image Process.*, vol. 6, pp 538-557, 1977.

[54] Porter R. I., Further Elementary Analysis, Bell and Sons, 1969.

[55] Adobe Systems Inc., PostScript Language Reference Manual, Addison-Wesley 1985.

[56] Apple Computer Inc., 20525 Mariani Ave., Cupertino, CA 95014, USA.

[57] Knight K. L. and Valaski W., Using AutoCAD, Que Corporation, 1989.

[58] Weldcrack Expert, The Welding Institute, Abington Hall, Abington, Cambridge CB1 6AI, UK.

APPENDICES

APPENDIX I

Proof that the initial descent of the indenter should always be visible in a path record produced by the analysis program.

Consider an object falling vertically from rest from a height a height h. We wish to calculate the time t_h for the indenter to fall through this distance. Using the equation:

$$s = ut + \frac{1}{2}at^2$$

where:

u = initial velocity = 0
s = distance = h
a = accceleration = g
t = time = t_h

yields:

$$h = \frac{1}{2}gt_h^2$$

$$t_h = \sqrt{\frac{2h}{g}}$$

$g = 9.81 ms^{-2}$ and for the test rig used $h = 0.25m$ therefore:

$$t_h = \sqrt{\frac{2 \times 0.25}{9.81}}$$
$$= 0.226 \text{ s}$$

Since the frame capture time is 40 ms (≡ 25 frames/sec) the image of the indenter on it's initial descent should be present within at least 5 captured frames, therefore the initial descent of the indenter should always be present within the path record.

APPENDIX II

Derivation of a worst-case estimate of the right hand window limit location.

In Chapter 3 it was explained that the analysis program locates the top of the indenter trajectory, on the first rebound, by first finding the rising and falling parts of the trajectory. The right hand window limit must therefore be placed at least as far right as the point corresponding to the second impact. In temporal terms, this point occurs at the time t_3 in Fig. AII.1. Clearly, t_3 will be at a maximum when the indenter impact is totally elastic (i.e. rebound height = drop height). From symmetry:

$$t_3 = 3 \times t_1$$

and from Appendix I:

$$t_1 = 0.226s$$

therefore:

$$t_3 = 3 \times 0.226 = 0.678s$$

Since the frame capture time is 40 ms ($\equiv$ 25 frames/sec) the time t_3 corresponds to 17 captured frames. Furthermore, since 4 rows of pixels are extracted from each captured frame this corresponds to $4 \times 17 = 68$ rows. Therefore:

co-ordinate of right hand window = co-ordinate of left hand window + 64

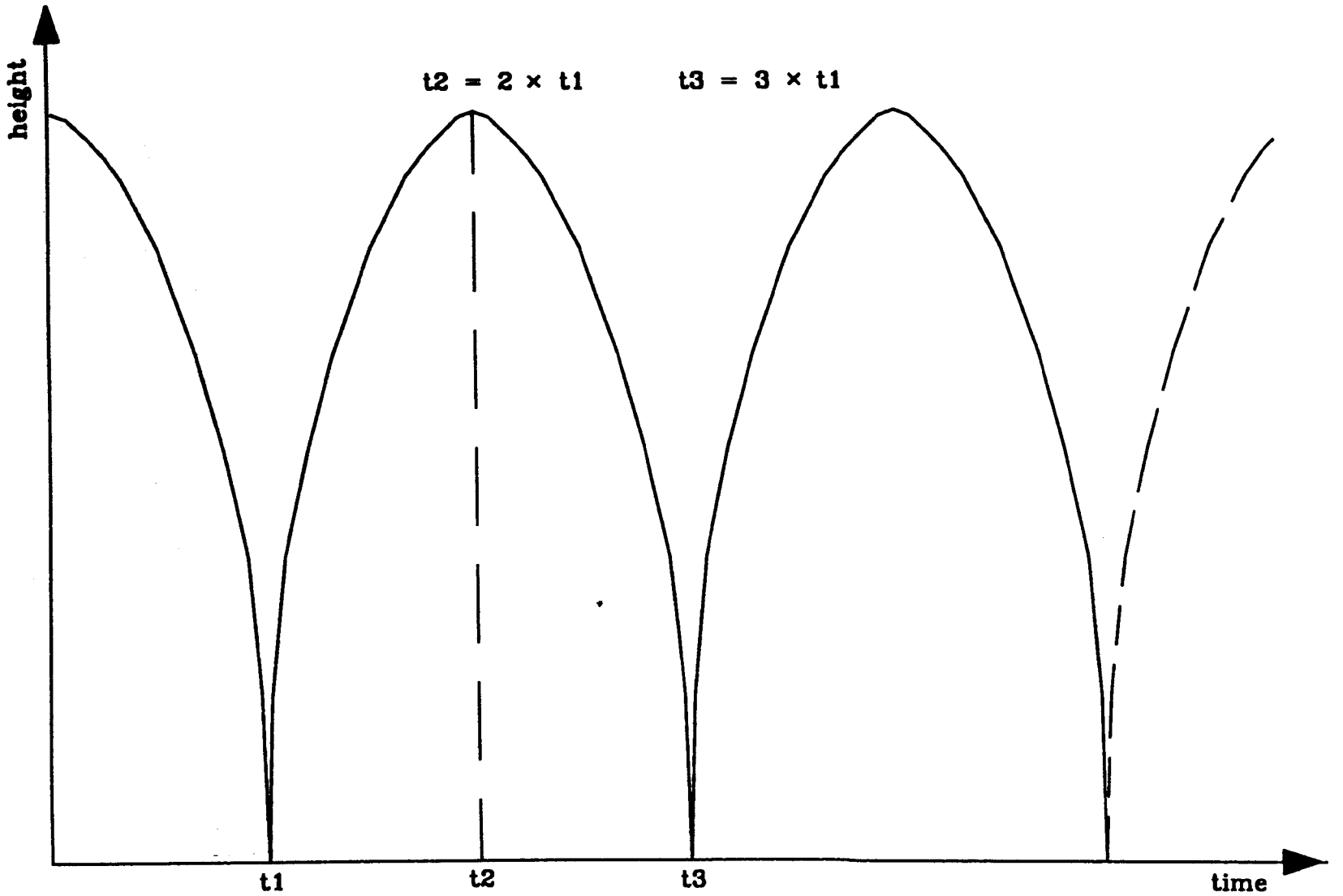


Figure AII.1: INDENTER PATH WITH ELASTIC IMPACT

APPENDIX III

Worst case estimate of rebound height error cause by finite sampling frequency

The worst case error, Δh, occurs when frames are captured at equal time intervals before and after the point at which the top of the trajectory is reached i.e.:- times t_1 and t_2 in Fig. AIII.1. Using the equation:

$$s = ut + \frac{1}{2}at^2$$

where:

u = initial velocity = 0
s = distance = Δh
a = acceleration = g
t = time = Δt

yields:

$$\Delta h = \frac{1}{2}g\Delta t^2$$

$g = 9.81 ms^{-2}$ and for the frame store used the sampling rate is equivalent to 50 Hz (i.e. 50 fields/sec). Therefore:

$$2\Delta t = \frac{1}{50}$$
$$\Delta t = 0.01 \ s$$

$$\Delta h = 0.5 \times 9.81 \times 0.01^2$$
$$= 4.905 \times 10^{-4} \quad m$$

Therefore the worst case error introduced by limited sampling is approximately 0.5 mm.

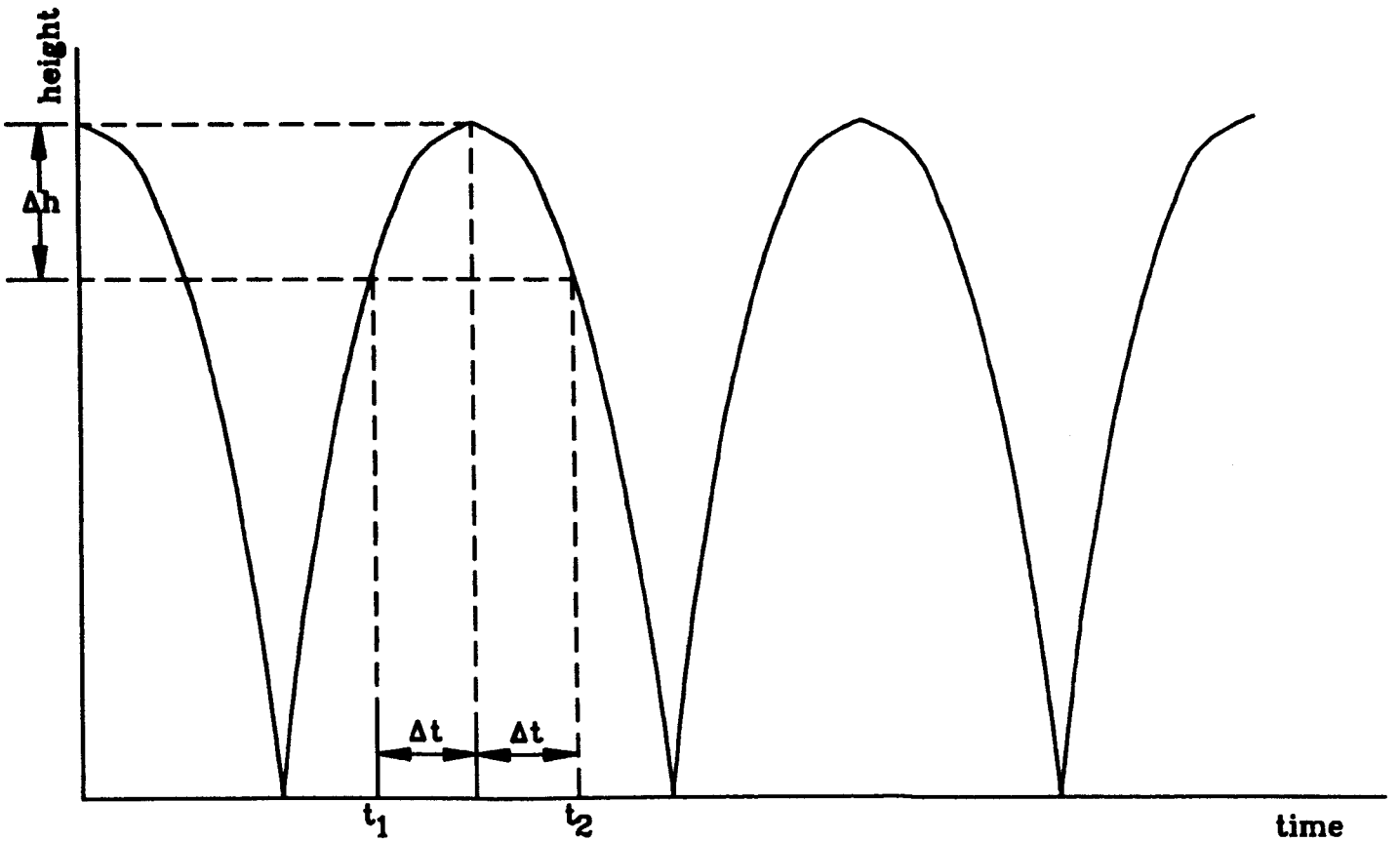


Figure III.1: WORST CASE ERROR ESTIMATE

APPENDIX IV-VIII

It is not possible to provide a full listing of the programs here due to space limitations. The interested reader should refer to the supplementary volume of this thesis for the relevant software.