A SYSTEMATIC ALGORITHM DEVELOPMENT FOR IMAGE PROCESSING FEATURE EXTRACTION IN AUTOMATIC VISUAL INSPECTION

A thesis presented in partial fulfilment of
the requirements for the degree of
Master of Technology
in
the Department
of Production Technology
Massey University

By

G.X. Xing

1990

ABSTRACT

Image processing techniques applied to modern quality control are described together with the development of feature extraction algorithms for automatic visual inspection.

A real-time image processing hardware system already available in the Department of Production Technology is described and has been tested systematically for establishing an optimal threshold function.

This systematic testing has been concerned with edge strength and system noise information. With the a priori information of system signal and noise, non-linear threshold functions have been established for real time edge detection.

The performance of adaptive thresholding is described and the usefulness of this nonlinear approach is demonstrated from results using machined test samples.

Examination and comparisons of thresholding techniques applied to several edge detection operators are presented.

It is concluded that, the Roberts' operator with a non-linear thresholding function has the advantages of being simple, fast, accurate and cost effective in automatic visual inspection.

ACKNOWLEDGEMENTS

First, I would like to thank my parents and my brother for their constant encouragement and support which enabled me to take the opportunity of obtaining advanced training in technology.

I would like to express my sincere thanks to my supervisors Professor W. Monteith and Dr. Bob Chaplin for their guidance and careful instruction, for their many helpful suggestions throughout the work.

I would also like to thank Professor Bob Hodgson for his invaluable assistance and in sharing his vast experience and information on computer vision.

I would also like sincerely to thank the following people who have contributed to this work. Dr. Roger Browne, Mr. Ralph Pugmire, research officers at the Department of Production Technology, and Dr. D. G. Bailey, director of Image Analysis Unit of Massey University, for their invaluable advice and help during this study.

Fellow postgraduate Wyatt Page and Siew W.H. for their friendship and moral support. Department technicians Ms. Vicki Spagnolo, Mr. Bruce Rapley, and Mr. Eliot Blennerhassett have also provided technical assistance on a number of occasions.

The academic and clerical staff of the Department of Production Technology for their kindness and help. The staff of Massey Library and Photography Department for excellent service.

Department of Production Technology for providing financial support.

Finally, I would like to express my deep appreciation for the confidence, encouragement and support from her heart given to me by my wife Dr. Shuman.

Contents

Summary

This thesis consists of 7 chapters.

Chapter 1 addresses the applications of machine vision technology for quality control. Achieving 100% product quality meeting customers needs, model based automatic visual inspection systems are becoming reality with emerging computer techniques.

Chapter 2 reviews image processing techniques of edge detection, because it is the fundamental processing work in model based automatic inspection.

In chapter 3, a high-speed image processing system used for the experimental work is described, followed by an analysis of feature extraction algorithm improvement requirements. This leads to the formulation of an algorithm development strategy.

Developed software and its edge strength information-noise analysis from systematic plotting of 3-D edge pixel and 2-D contours are presented in chapter 4.

Chapter 5 gives a machine vision thresholding principle which is deduced from Weber's law. Edge processing results of Robert's edge operator with the established non-linear thresholding profiles are presented.

Chapter 6 demonstrates Robert's, Sobel, DIP, RANK and RANGE filter edge detecting performance, and discusses the advantage of thresholding techniques.

The overall conclusions and suggestions for future work are given in chapter 7.

CHAPTER 1

CHAPTER 1 Introduction

1.1 Introduction

Throughout industry today there is a need to improve product and process quality. Good quality will improve productivity and profit.

To achieve tomorrow's quality goal of 100% conformance to requirements, automatic visual inspection and new quality concepts will play a very important part. The real challenge for the development of automatic visual inspection technology is to provide efficient and effective methods to meet customer needs and expectations, at adequate operating speed.

Accurate feature extraction is important in automatic visual inspection, because features are less sensitive to noise in the original gray-scale images and provides data reduction while preserving information required for the inspection. The most important initial feature detection task is that of edge detection, where the goal is to find pixels that lie on the borders between different objects in the scene.

In the past twenty years, many edge detection technologies have been developed [Rob 65; Dav 75; Ros 76; Pra 78; Mar & Hil 79; Mcl et al 84; Hod et al 85; Can 86;]. Because the theory of optimum edge detection algorithms is not well developed, most have limitations for real-time automatic visual inspection applications. This remains the subject of further research.

1.2 Automatic visual inspection in quality control

The benefits that may be realised from quality control include:

1) Improvement in quality of products and services;
2) Increase in the productivity of manufacturing processes;
3) Reduction of manufacturing costs;
4) Determination and improvement of the marketability of products and services;
5) Reduction of "in use" failure;
6) Reduction of consumer prices of products and services;
7) Increase in service life;
8) Improvement in deliveries and availability;
9) Enhancement of the management of an enterprise.

The quality control function has traditionally been performed using manual inspection methods and statistical sampling procedures. Manual inspection is generally a time-consuming task. It requires that parts be removed from the production lines to a separate inspection area. This can cause delays and may become a bottle neck in the production line.

Basic Statistical Quality Control (SQC) philosophy is concerned primarily with the early detection of assignable causes so that product quality may be controlled at the desired level with a minimum of rejects. Table 1 lists the most commonly used statistical tools by category of application.

SQC from its inception has been concerned with economic considerations, and recognises that products must be produced to an adequate quality level at an economical cost. According to J.M. Wiesen [Jur 79] in quality control, advantages of sampling inspection are:

1) Economies due to inspecting only part of the product;
2) Less handling damage during inspection;
3) Fewer inspections, thereby simplifying the recruiting and training problem;
4) Upgrading the inspection job from monotonous piece by piece decisions to lot by lot decisions;
5) Applicability to non-destructive testing, with a quantified level of assurance of lot quality;
6) Rejections by vendors or shop departments of entire lots rather than mere return of the defectives, thereby providing strong motivation for improvement.

The main disadvantages of sampling inspection are:

1) There are risks of accepting "bad" lots and of rejecting "good" lots;
2) There is added planning and documentation;
3) The sample usually provides less information about the product than does 100% inspection.

In reducing time-consuming 100% manual inspection, statistical sampling procedures have been proposed. However, using statistical sampling procedures is an acknowledgement of the risk that some defective parts will slip through. In principle, statistical quality control accepts that something less than 100% quality must be tolerated.

Table 1-1 Summary of Statistical Methods

Application category	Statistical tools
Inspection and test for product acceptance	Lot acceptance sampling By attributes By variables Continuous acceptance sampling Failure distribution analysis
Process control	X and R control charts Precontrol (for variables) P and C charts (for attributes)
Variation research (diagnostic)	Paveto analysis Frequency distribution analysis Process capabilities analysis Multivariate analysis Component swap and Variable swap Measurement accuracy analysis Design of experiments (Taguchi Methods) Analysis of variance Significance testing Evolutionary optimisation Weilbull (and other) failure analysis Scatter design

There are several economic, social, and technological factors at work to modernise the quality control function. The economic factors include the high cost of the inspection process as it is currently performed and the desire to eliminate inspection as a source of costly delay in production.

The social factors include the ever increasing demand by customers for near perfection in the quality of goods, the growing number of expensive product-liability legal cases, and government regulations which require many firms to maintain comprehensive production and quality records.

Another factor in this category is the tendency for some manual inspection tasks to involve subjective judgement on the part of the human inspector. It is considered desirable to try to remove this subjective component from the inspection operation.

The technological factors contributing to modernisation of the visual inspection are the following:

* Extensive automatic visual inspection research and development
* Advances in computer technology
* Faster and more powerful microprocessors
* Cost of computer power continues to drop
* Advances in sensing devices such as charge-coupled devices

All of the above factors are driving the quality inspection function toward 100% automatic inspection. The objectives of automatic visual inspection are:

1. To improve product quality;
2. To increase productivity in the inspection process;
3. To increase productivity and reduce lead times in production.

The strategy for achieving these objectives is basically to automate the inspection process through the application of computers combined with advanced Image Processing technology. Wherever technically possible and economically feasible, inspection should be done on a 100% basis rather than sampling.

100% automatic visual inspections have potential advantages in:

* freeing humans from dull and routine tasks,
* saving human labour costs,
* performing inspection in unfavourable environments,
* reducing demand for highly skilled inspectors,
* analysing statistics on test information and keeping records for management decisions,
* matching high speed production with high speed inspection,
* reducing quality loss,
* ensuring product quality, and in
* increasing profit.

In addition, on-line 100% inspection will introduce opportunities to use the inspection measurements as feedback data to make compensating adjustments in the production process.

There are a number of survey articles and bibliographies published on automatic visual inspection studies of various industrial parts [Chi 82], [Chi 88], [Wal 88].

Chin [Chi 82] has given the following list of automatic visual inspection application categories:

* the inspection of printed circuit patterns;
* the inspection of microcircuit photomasks;
* inspection of integrated circuits on silicon and their alignment for bonding;
* inspection of other electrical and electronic assemblies;
* the inspection of automobile parts;
* visual inspection for metal processing industries;
* visual inspection for fabric processing industries;

The basic components of automatic visual inspection systems are an illumination mechanism, a sensor, an A/D converter, and a processor as illustrated in Figure 1-1.

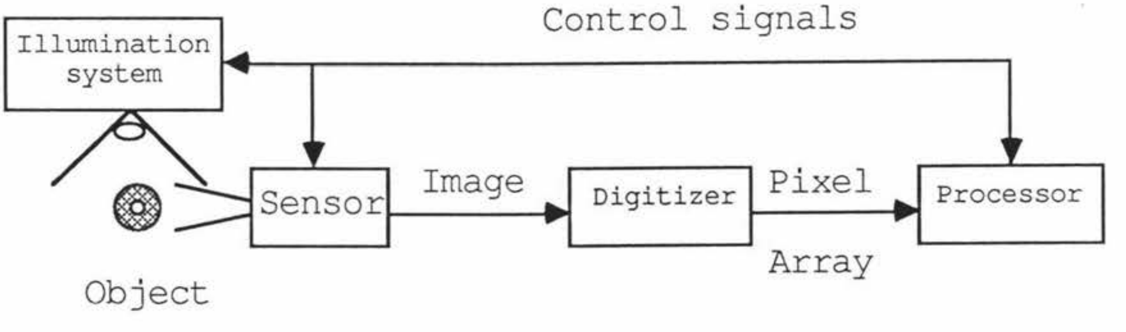


Figure 1-1. Basic components of a automatic visual inspection system [From Ros 85]

The camera-based sensor acquires an image of the object that is to be inspected. The digitizer converts this image into an array of numbers (pixels) representing the brightness values of the image at a grid of points. The processor retains the image for processing to enable the necessary decisions to be made. The processor may also have some degree of control of the sensor, the object, or the environment.

For a particular application it is clear that the design of a machine vision system involves a large number of specialist disciplines related to optics, sensor physics, analog to digital conversion (A/D), direct memory access controller design, algorithm development, computer hardware specification, information display and mechanical control.

1.3 Model based automatic visual inspection

In industry, model-based machine vision inspection algorithms have been in continual development for twenty years. To date, primarily very simple techniques based on 2-D feature models have been applied on real time production lines. However, Chin [Chi 88] has indicated that current model-based systems have weaknesses such as a highl dependence on precision mechanisms and special lighting. More sophisticated techniques need to be developed in order to deal with less structured production environments and to permit more task versatility.

Generally, a model based automatic visual inspection system consists of a modelling phase, a feature extraction phase, and detection phase, as illustrated in Figure 1-2.

Features representing the object are extracted and matched to a pre-defined model. This feature-to-model matching process is the most common technique for detecting defects. It is realised by finding features in the given object that match the model definition of defects. After the detection, a go/no go decision is made to sort the bad parts from the good.

In an automated visual inspection, the first hardware task is to acquire a quality image of the part under inspection. This is often considered to be the most crucial step in the inspection process. The primary objectives are to acquire a quality representation of the object under inspection with the minimum of complexity so as to reduce the subsequent image processing.

Image acquisition involves the design of illumination and optics, and the choice of sensors and their placement. The choice of lighting, optics, and sensors depends largely on the following issues:

1) The type of sensors to be used;
2) The speed of motion of the object;
3) The minimum feature size of the part under examination;
4) The nature of the relevant features for inspection.

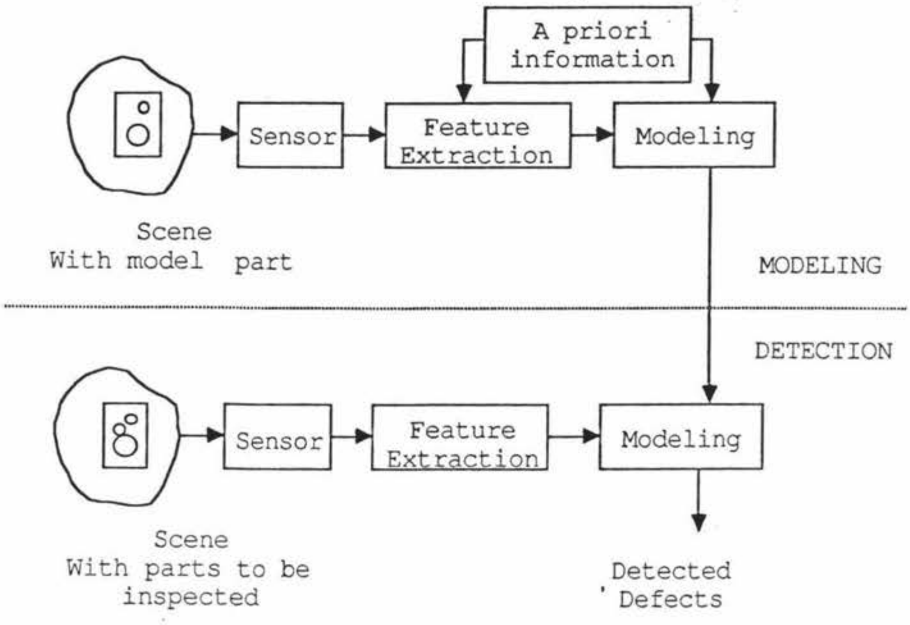


Figure 1-2. Components of a model-based inspection system [From Chi 88]

In some cases, special lighting, such as structured light or polarised light, is employed.

Another basic requirement is the image processor. A common problem in inspection applications is the need to process large sets of predominantly two-dimensional arrays at high speed. Study [Han 86] shown that processing speed of at least one billion operations per second will be required to solve some of the current inspection tasks. This requirement has led to much research in algorithm and special high speed hardware developments [Mcl 84, Nag 87, Elp 87].

High speed algorithms must be realisable in hardware such as general purpose systolic arrays, pipeline architectures, custom VLSI design of specific algorithms, or even more powerful parallel multiple instruction/multiple data architectures to meet the needs of real-time image processing, and image interpretation.

All visual inspection algorithms use "a priori" knowledge. This knowledge is organised into models which provide strategies and standards for the inspection process. Therefore, these systems are referred to as model-based or model-driven systems [Chi 88]. They perform inspection by matching the part under inspection with a set of pre-defined models.

In practice algorithms are developed heuristically by selecting an initial sequence of operations from a large number of possible operations, testing the sequence and then making modifications until satisfactory performance is achieved.

Although research and development in inspection algorithms has increased dramatically over the last twenty years their capability is still very primitive. One reason for this problem is that many manufacturing tasks require sophisticated visual interpretation, yet demand low cost, high speed, accuracy, and flexibility.

1.4 Edge processing in inspection

The main purpose of model based automatic visual inspection is the extraction of the most significant features such as edges, then matching these features with a set of pre-defined models to detect defects.

Accurate feature extraction is important in automatic visual inspection, because it will reduce data and noise while preserving the information required for the inspection.

In edge detection, the goal is to find pixels that lie on the borders between different objects in the scene [Ros 88]. In order to develop more realistic feature extraction algorithms to achieve high speed processing, it is necessary to examine current edge detection techniques.

Edge extractors can be divided into two broad categories. One is based on computation which includes local gradient, digital image filtering processing etc. Other approaches are knowledge based such as adaptive or median filtering techniques.

Traditionally, an "edge" is the first step discontinuity that lies between two regions of uniform but different light intensity. In reality natural image regions are rarely of uniform intensity and the step function must be smoothed into a ramp or similar function.

Extensive research has been devoted to the very earliest stage of vision; extracting discontinuities from the pixel array. When the images are entirely noise free this is simple, but traditional local edge detection methods are very sensitive to additive noise.

CHAPTER 2

CHAPTER 2 Edge Detection Processing

In this chapter, a number of edge detection methods including gradient operators, thresholding techniques and filtering techniques are reviewed.

2.1 Gradient operator principles

The most common edge operators are gradient based, the historically earliest is the Roberts" operator [Rob 65] but a more reliable version is the Sobel [Cro 86].

The Robert's and Sobel edge operators assume that there is a peak in the gradient of the image intensity, but obtaining the slope by simply differentiating the image will amplify the noise as well as the discontinuity. So, the signal to noise ratio has to be improved by smoothing. The smoothing technique works well because the signal is present over an area of the image while the noise appears randomly.

Generally, gradient edge detection methods measure combinations of the gradient values of the pixels in a local region. These operators are simple. However, since these approaches make no use of local characteristics or a priori information, their edge detection performance will be less than optimum. Consequently, while the direct gradient edge operator was very effective for ideal step edge detection, sensitivity to noise is the current limitation. Ramp edges are also extracted as thick lines.

2.1.1 Roberts operator

With the Roberts' operator [Abr 81], for an image brightness function g(i,j) the gradient magnitude s(i,j) and angle f(i,j) can be computed as:

$$s(i,j) = (\Delta_1^2 + \Delta_2^2) / 2 \qquad (2\text{-}1\text{-}1)$$

$$f(i,j) = \tan^{-1}(\Delta_2/\Delta_1) \qquad (2\text{-}1\text{-}2)$$

where

$$\Delta_1 = g(i,j+1) - g(i+1,j)$$

$$\Delta_2 = g(i+1,j+1) - g(i,j) \qquad (2\text{-}1\text{-}3)$$

Equations (2-1-3) describe only one of many possible difference operators. Other methods of measurement of gray-level intensity difference could be used along either horizontal or vertical directions.

Figure 2-1 (a,b,c) shows three different Roberts operators. A rather simple Roberts operator employed in this study is shown in Figure 2-1 (a). Other forms such as in Figure 2-1 (b,c) could also be employed.

$$s(i,j) = |g(i,j)-g(i+1,j+1)| + |g(i,j+1)-g(i+1,j)| \qquad (2\text{-}1\text{-}4)$$

This operator involves only four points and is therefore argued to be sensitive to noise and surface irregularities [Dav 75].

Δ2		Δ1		
0	1	1	0	(a)
-1	0	0	-1	
1	-1	1	0	
0	0	-1	0	(b)
1	-1	1	1	
1	-1	-1	-1	(c)

Figure 2-1 Robert operators.

2.1.2 Sobel operator

The Sobel gradient magnitude is calculated in a similar way as Roberts operator:

$$\Delta 1 = [g(i-1,j+1)+2g(i,j+1)+g(i+1,j+1)] - [g(i-1,j-1)+2g(i,j-1)+g(i+1,j-1)]$$

$$\Delta 2 = [g(i+1,j+1)+2g(i+1,j)+g(i+1,j-1)] - [g(i-1,j+1)+2g(i-1,j)+g(i-1,j-1)]$$

$$S(i,j) = (\Delta_1{}^2 + \Delta_2{}^2) / 2 \qquad (2\text{-}1\text{-}5)$$

This operator gives a better result due to the local averaging implicit in the larger template.

Sobel operator configuration is shown in Figure 2-2.

Δ2

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

Δ1

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

Figure 2-2 Sobel operator.

There are many other edge filters operating in the local gradient approach; most of them utilise 3 by 3 [Pra 78], 4 by 4. To improve the edge detection performance some operators such as the Marr operator have been reported using 24 by 24 pixel masks [Mar & Hil 79].

2.1.3 DIP operator

Ryoo and Kim [Ryo & Kim 88] have proposed a difference of inverse probabilities (DIP) operator for detecting edges. This approach arises from an understanding of fundamental human visual characteristics.

For example the human viewer is more sensitive to the edges and valleys in "bright" regions than those in very dark regions. In principle, valleys are the regions which are composed of local intensity minima, they are universal in the visual perception of object shape. Therefore, to perceive and analyse an object in a manner akin to the human visual system, one must extract sketch features (valleys and edges) subject to the local intensities.

The DIP operator extracts sketch features by using local intensity in a quite simple manner: firstly a DP operator is defined as:

$$DP = \frac{f_m(i,j)}{\bar{f}(i,j)} - \frac{f(i,j)}{\bar{f}(i,j)} \qquad (2\text{-}1\text{-}6)$$

where $\bar{f}(i,j)$ is the sum of intensities and $f_m(i,j)$ is the maximum intensity in a 3x3 window. The DP not only responds to valleys and edges owing to the difference being taken, but also depends on local intensities f(i,j). Therefore, DP values are larger in dark regions than in bright regions, under the same conditions of rate of intensity change.

DIP operation is introduced to improve the DP performance in extracting edges in low local intensity regions:

$$DIP = \frac{\bar{f}(i,j)}{f(i,j)} - \frac{\bar{f}(i,j)}{f_m(i,j)} \qquad (2\text{-}1\text{-}7)$$

Equation (2-1-7) is only applied when the difference of $f_m(i,j)$ and $f(i,j)$ is greater than a given threshold. Each $\bar{f}(i,j)$ and $f_m(i,j)$ of equation (2-1-7) are much the same in valleys and their neighbours, while $f(i,j)$ are smaller in the valley than in their neighbours. Thus, the values of the DIP are so much larger in the valleys themselves that it extracts valleys thinly. Ryoo and Kim's [Ryo & Kim 88] results have shown that the DIP has high values in valleys and edges and low values in smooth regions. It can therefore extract sketch features efficiently.

2.2 Thresholding for edge detection

Thresholding techniques fall into three categories: fixed threshold, automatic global threshold, and local adaptive threshold. Fixed threshold is a common technique used with gradient operators, but finding a proper threshold level is difficult in low contrast images. Although extensive research work remains to be done, the last two new techniques have shown that they may be employed in finding an optimum thresholding profile for better edge detection.

2.2.1 Fixed thresholding

The simplest way to produce a binary image is with a fixed or manually adjustable threshold [Ros 76]. However, a fixed thresholding approach cannot handle variations in illumination or variation in surface reflectance of the light and dark regions of the scene.

Better thresholding levels may be achieved iteratively, but the testing and adjusting process generally is slow and tedious, therefore improvement of this processing is required.

2.2.2 Global thresholding

Global thresholding [Wes 78] uses the gray level information in a scene to compute a binary threshold appropriate for that scene, usually by histogram analysis. The advantage of this approach is that it allows the system to handle large variations in illumination intensity or surface contrast. However, the extensive processing time required presents an impediment for the technique to be applied widely in high speed image processing.

2.2.3 Adaptive thresholding

This non-linear thresholding approach computes a threshold on a pixel-by-pixel basis, using the grey level information in a small region surrounding a pixel. The developed technique can handle the effects of illumination and surface reflectance of non-uniformities [Mon and Xin 89]. Because the algorithm is so simple real-time processing has been achieved with a low cost hardware system [Mon et al 89].

2.3 Filtering for edge detection

Filtering is a very general notion of transforming the image intensity in some way so as to enhance or de-emphasise certain features. Filtering techniques have been studied extensively in image processing [Dav 75, Sha et al 79 and Pra 78].

The main goal of filtering in edge detection is to change image gray levels to enhance the appearance of objects. Most often this means using transformations that make the intensity discontinuities between regions more prominent.

2.3.1 Linear filters

For linear filters, the output value is a linear combination or weighted average of the input pixel values within the selected window. A linear low-pass filter is often used to reduce noise, but it will blur edges, because, low-pass filtering reduces the high-frequency components of the processed signal, and edges or details of an image usually contribute to the high-frequency components.

High-pass filtering emphasises the high-frequency components of a signal while reducing the low-frequency components. Because the high-frequency components of a signal generally correspond to edges or fine details of an image high-pass filtering often increases the local contrast and thus sharpens the image.

Since high-pass filtering emphasises the high-frequency components, it tends to increase background noise power when it is applied to degraded images. This is a major limitation of high-pass filtering for image enhancement.

2.3.2 Median filters

Median filtering [Pra 78] is a non-linear process in reducing impulsive salt-and-pepper noise. In using a median filter, a window slides along the image, and the median intensity value of the pixels within the window replaces the intensity of the pixel being processed. For example, when the pixel values within a window are 20,25,30,35, and 40, and the pixel being processed has a value of 40, it value is changed to 30, which is the median of the five values.

Like low-pass filtering, median filtering smooths the image and, therefore, is useful in reducing noise. Unlike low-pass filtering, median filtering can preserve discontinuities in a step function and can smooth a few pixels whose values differ from their surroundings without affecting the other pixels.

A.C. Bovik, T.S. Huang and D.C. Munson have proposed an effective median filtering technique for edge estimation and detection [Bov et. al 87], they indicated that an important design parameter in using a median filter is the size of the window used. If the window size is smaller, the pixels with impulsive values are not significantly affected.

Median filtering has been successfully applied to many image processing tasks. R M Hodgson et al [Hod et al 85] and Bailey [Bai & Hod 85] have proposed RANK and RANGE filter for edge detection and noise reduction.

The RANK filter is one of a class of two dimensional non-linear filters. These filters replace each pixel of the image by a value determined by some non-linear function of the neighbouring pixels.

The filter output is the pixel value selected from a specified position in a ranked list, of the pixel values within a window area of N pixels:

$$(f_1,f_2,f_3,...,f_n) \qquad (2\text{-}4\text{-}1)$$

where

$$f_1<=f_2<=...f_n \qquad (2\text{-}4\text{-}2)$$

the output is selected as:

$$rank(i)=f_i \qquad 1<=i<=N \qquad (2\text{-}4\text{-}3)$$

Once this is done for all possible window positions, it is represented as:

$$g=R_i(f) \qquad (2\text{-}4\text{-}4)$$

where f is the input image, g is the processed image and i is the rank position selected. i.e. filtered image pixel intensity is determined by the ranked number's R(i) pixel intensity f(i) in the selected window.

Range filter, takes a range of intensities which is a straightforward derivative of the rank filter. For a given neighbourhood, the range filter extracts the absolute difference between the Mth and Nth entries of an ordered list of the intensities in the neighbourhood.

A range filter may be implemented within a selected window from:

$$(f_1,f_2,f_3,...,f_n) \qquad (2\text{-}4\text{-}5)$$

where

$$f_1<=f_2<=...f_n \qquad (2\text{-}4\text{-}6)$$

then subtracting the intensity values for two selected position (i and j)

$$range(j,i_0=f_j-f_i) \quad 1<=i<=j<=N \qquad (2\text{-}4\text{-}7)$$

When this is performed over the whole image, it may be represented by

$$g = Ra_{ij}(f) \qquad (2\text{-}4\text{-}8)$$

where g is the output image. Thus the range filter is a extension of another local non-linear rank filter defined as

$$rank(i) = f_i \qquad 1 <= i <= N \qquad (2\text{-}4\text{-}9)$$

When performed over the whole image, this may be represented by

$$g = R_i(f) \qquad (2\text{-}4\text{-}10)$$

Therefore, combining equations (2-4-7) to (2-4-10) gives

$$Ra_{i,j}(f) = R_j(f) - R_i(f) \quad i < j \qquad (2\text{-}4\text{-}11)$$

i.e. filtered image pixel intensity is determined by the difference of ranked R(j) pixel intensity f(j) and ranked R(i) pixel intensity f(i) in the selected window.

CHAPTER 3

CHAPTER 3 Real-time Edge Detection

3.1 Image processing system

The high-speed image processing hardware system for the current study was designed and constructed in 1984 [Mcl & Mon 84]. The design was based on the concept that image intensity change is a primary characteristic in human visual perception, and therefore the detection of these changes, or edges, was chosen as the function to be performed by the hardware system.

High operating speed is achieved by implementing a modified Roberts' differential operator in a pipelined parallel processor which can by-pass the data processing bottle-neck of greyscale image capture and processing.

The system based on mid 80s Fairchild Advanced Schotky Transistor (FAST) devices operates at 10 MHz pixel rate which can accommodate images resolution of 256x292 pixels at 25 frames/second.

Development was based on a computer simulation of combining the Robert's and Sobel operator with thresholding technique. To achieve high speed the computationally efficient Roberts' differential operator was chosen. To improve system edge extraction performance, non-linear thresholding was implemented [Mcl & Mon 84].

The system consists of a TV camera, A/D converter, Roberts' Operator & Local average brightness calculation (LAB), edge picture frame store, and Digital Equipment Corporation's Micro PDP-11/23 Plus host Computer units as shown in Figure 3-1.

The TV camera is a standard Panasonic WV-1500 series vidicon tube TV camera with 1 volt of peak-peak composite video signal. Signal-to-noise ratio is more than 40dB with 4.5 MHz filter.

The A/D converter is a Flash mode device capable of 60 million conversions/second. The digital values from the A/D converter are passed to the input buffers of a two-line video store. The 8-bit gray-scale digital values are split into odd and even picture lines and passed to the appropriate RAMs for storage.

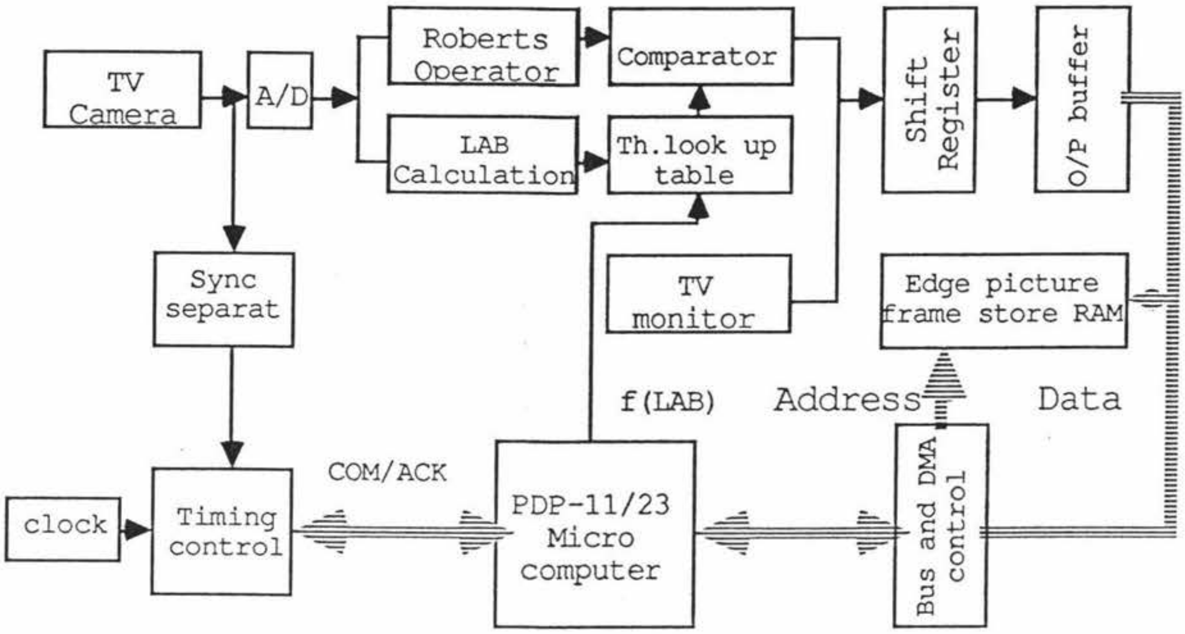


Figure 3-1 High speed image processing
hardware system block diagram

There are two parallel data paths in the arithmetic processing stage: one calculates the Robert's operator (RP), the other calculates the local average brightness LAB. LAB value is used to establish a look-up table with programmable threshold value f(LAB). Comparing Robert's operator output and f(LAB) value will generate a single bit edge/no-edge signal. The result is then transferred to the edge picture frame-store.

To carry on processing, two distinct control levels in the image processor are required.

1) Dedicated hardwired logic for the generation of the timing and control signals required for system operation.

2) Supervisory control over the complete system by software running in the Micro PDP-11/23 Plus computer.

Figure 3-2 shows the basic control interface in the system. Supervisory programming control for threshold function setting, Edge picture frame grab, Edge picture print and Edge picture pixel count will be presented in chapter 4.

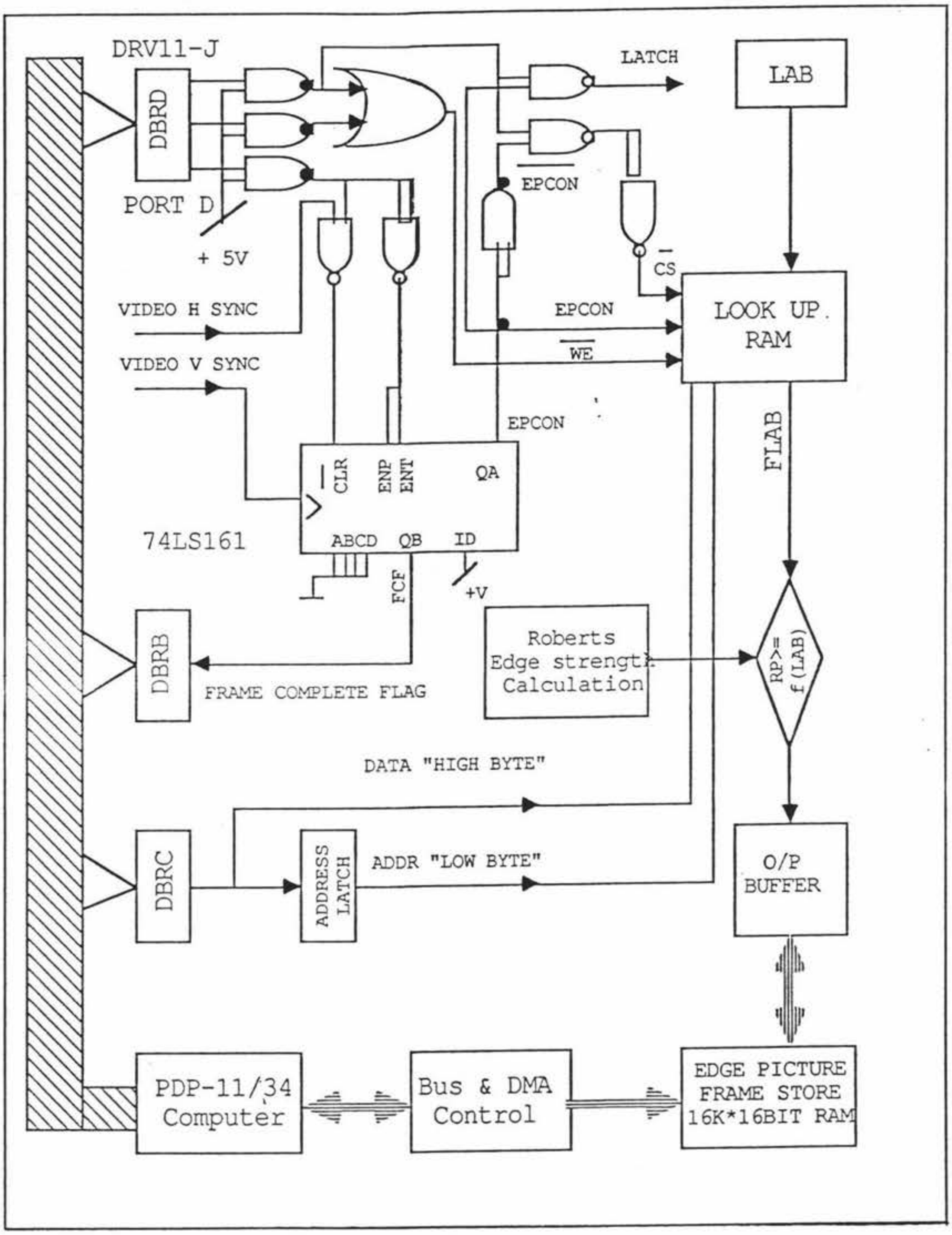


Figure 3-2

Real-time Image Processing System Control Interface

3.2 Algorithm improvement requirements

In automatic visual inspection the primary requirement has been that the system should be applicable of real-time tasks, therefore investigation should focus on fast and reliable image processing techniques.

To improve image processing performance, advanced theoretical and practical techniques are required. Established requirements are that optimum edge detection operators should be able to provide accurate localisation of edge elements and all real edges should have a high probability of being detected, while very few non-edge points should be marked as edges.

In the literature, fundamental limitations of existing gradient edge extraction techniques are: firstly, they are localised operators and thus do not take into account noise characteristics; secondly, the gradient operation itself is quite noise sensitive.

Early research work [Mcl & Mon 84] suggested that a non-linear threshold function using the simple Roberts' operator is a development opportunity which could enable complex tasks and environments to be accommodated by establishing an optimum non-linear threshold function.

Selecting a general optimum threshold is not a simple task. For example, in simple flat thresholding, if the threshold is set too low, many spurious candidate edge points are produced due to noise and textural fluctuations, whereas if it is set too high, insufficient candidates are produced to provide a contiguous set of points that may be identified as an edge segment.

The automatic generation of an optimum thresholding profile is a challenging research topic. Therefore a systematic development has been conducted to understand basic principles and to test image system signal/noise characteristics numerically.

3.3 Strategy of algorithm development

To meet algorithm development requirements, a strategy of four phases has been conducted:

1) Literature review;

More than 20 papers on feature extraction were located, and reviewed as outlined in chapter 1 and chapter 2. In the literature, no one algorithm could fully meet general industrial inspection requirements. Therefore, research and innovation were required in developing a fast, reliable and cost effective inspection system.

2) Generation and investigation of possible methods;

Early work [Mon & McI 84], [Xin & Mon 88] showed that non-linear edge detection using adaptive thresholding has potential for achieving better edge detection performance. Thresholding functions might have various forms, every such form representing a possible solution. Two methods have been used in determining the optimum thresholding profile. One is based on experiments to estimate noise in the system. The second approach will be a hypothesis which is based on locating the significant signal region and to achieve maximum signal to noise ratio.

3) Systematic testing of algorithms.

Noise estimation requires systematic testing, so computer aided testing was designed for this purpose. The testing is reported in chapter 4.

4) Examination thresholding performance.

Comparison was undertaken to understand different edge detection and thresholding techniques.

CHAPTER 4

CHAPTER 4 Edge Information and Noise Analysis

In this chapter, experimental design is given in section 4.1, including consideration of the test sample and illumination. Then, the software functions needed for testing edge information and noise are described in section 4.2 with detailed flow charts. Information and noise analyses for selecting the thresholding profile are presented in section 4.3.

4.1 Experimental considerations

To facilitate effective design, repeatable test conditions must be ensured and appropriate adjustments must be made: This implies 1) test sample selection; 2) illumination adjustment; 3) camera adjustment; 4) analog to digital board adjustment; 5) program preparation for frame total edge pixel counting, data transfer to VAX computer.

A set of plastic samples was used in the experiments as shown in Figure 4-1. The samples contain both round and rectangular forms. The test samples are of all 50 mm square sheet of yellow perspex with two group of holes. One set is of full holes of diameter 8, 4 and 2mm whilst the second set has the same size but are blind.

Considering practical production line illumination conditions, laboratory fluorescent lighting was chosen as the lighting source for the experiments. As illustrated in chapter 3, a Panasonic model WV-1500/B TV camera was used as shown in Figure 4-2. The vidicon camera's F1.6mm lens can be focused from 52cm to infinity, mechanical focus adjustment can be done by adjusting a screw at the back of the camera.

Figure 4-3 shows an analogue to digital converter (A/D) circuit board used to digitize the video signal, the TRW Flash A/D performance can be adjusted by setting reference levels for the maximum and minimum voltage outputs of the camera.

Tests on the A/D board indicated that the board had no DC restore, therefore video information on the low brightness region may be lost, and circuit adjustments were necessary.

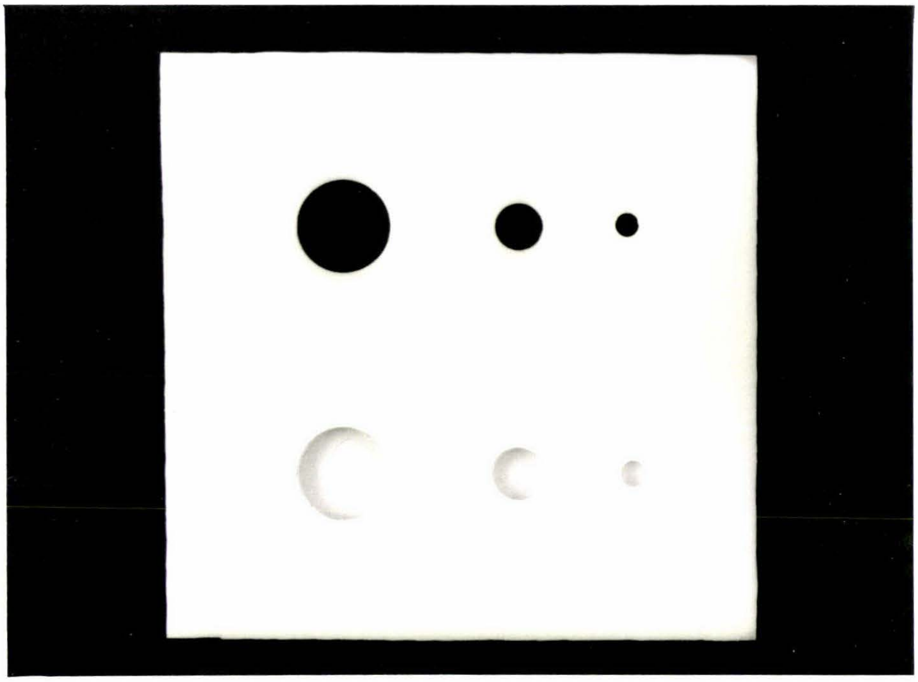


Figure 4-1
A six hole sample

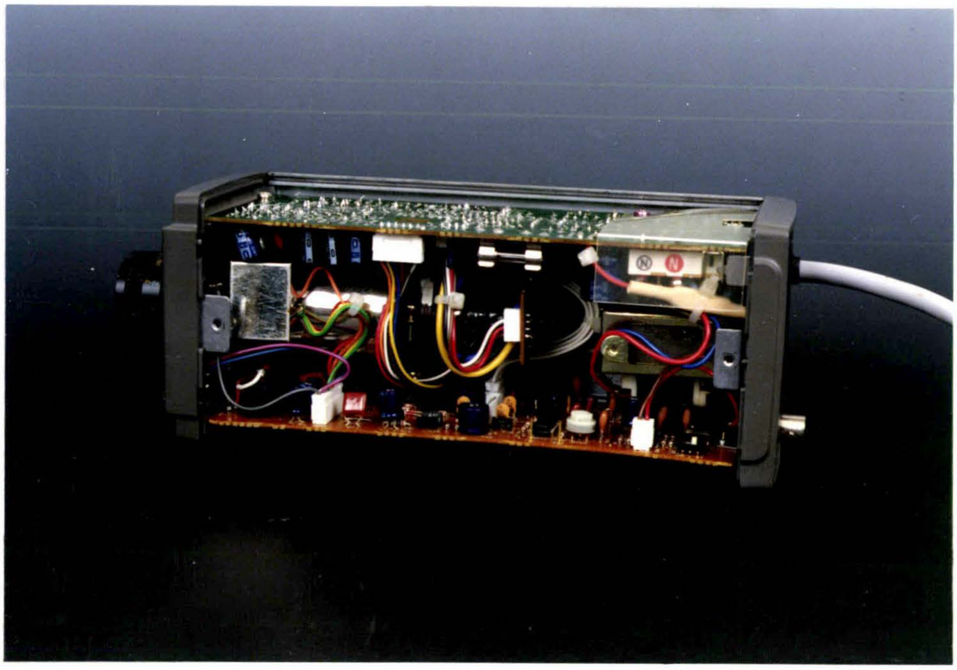


Figure 4-2
Vidicon camera mechanism

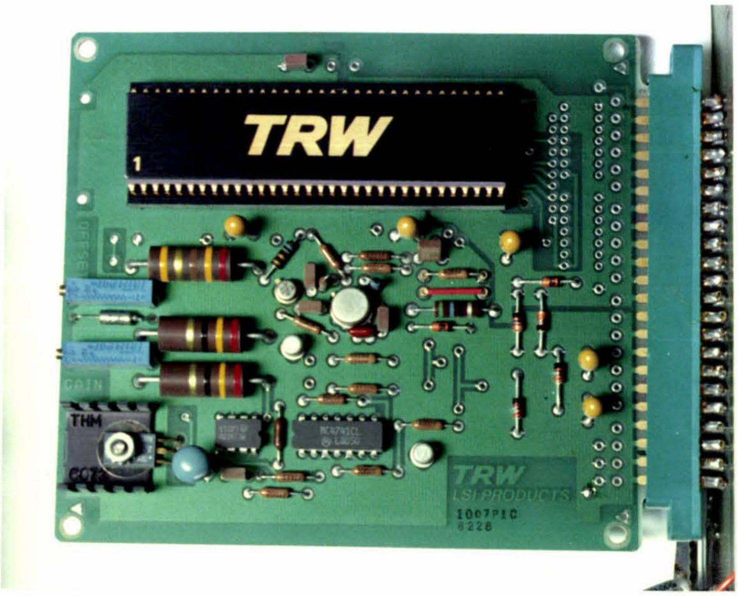


Figure 4-3
Image processing system A/D circuit board

4.2 Programs for analysis of edge information

A Macro-11 assembly language program previously developed to permit an interactive visual assessment of the performance of the hardware system was modified to a multiple function program.

The initial purpose of the program was to examine the relationship between good edges and most likely threshold level at each individual local brightness point. In the program, the generated single value edge strength was called "notch" as demonstrated in Figure 4-4.

Here LAB is the local average brightness, FLAB is a function of the local average brightness, and W is the width of the notches at the various brightness levels.

Early study suggested that several threshold functions could achieve better performance. Example profiles are shown in Figure 4-5. which are a function of local average brightness F(LAB) where the edge detection algorithm is given by : If RP>= FLAB then edge registered [Mon 84].

Initially simple flat thresholding was thoroughly tested by adjusting the threshold level until good edges occurred, but to obtain reliable edge detection results observation or estimation of edge magnitude over the entire the LAB region are required.

Because of the uncertainty of image processing system noise, subject to the scene content, systematic investigation of noise and edge strength are necessary.

Systematic testing requires more software support. To do the testing an edge pixel count program was written. The program was written in Macro-11 assembly language. This automatically calculates one complete frame total edge pixel numbers for a given single value threshold 'notch'.

For convenient display and analysis, the resulting pixel count data is transferred from the DEC Micro PDP-11/23 computer to a DEC Micro VAX II computer. This transfer is accomplished by using an RS-232 interface and a terminal interface program LNK, written by Mr. Ralph Pugmire of the Department of Production Technology.

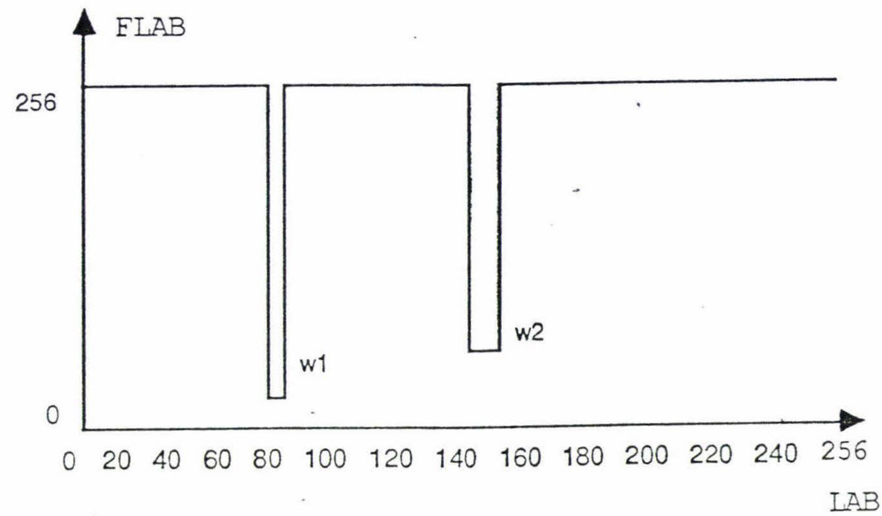


Figure 4-4 Selective "notch" as threshold function

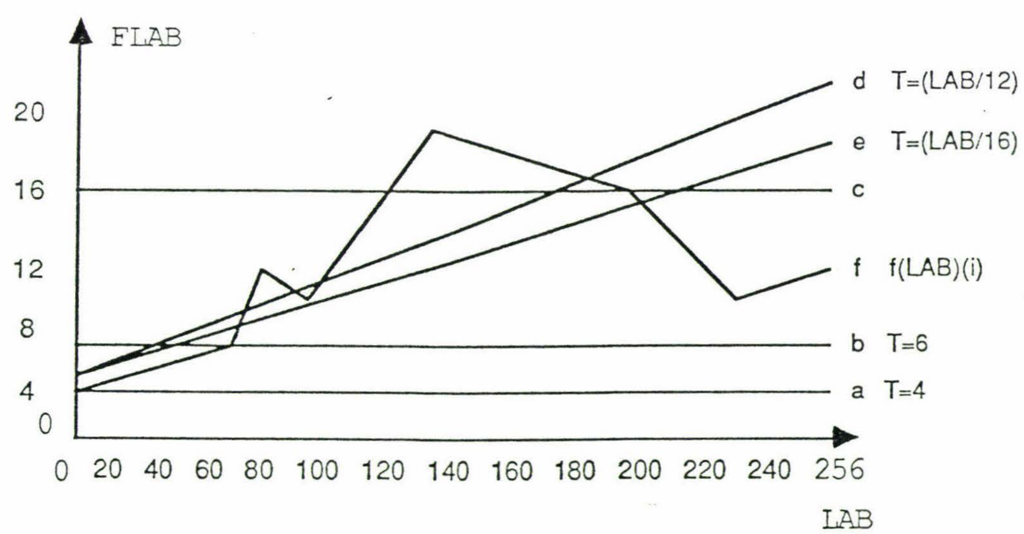


Figure 4-5 Non-linear threshold profiles

By using the MATLAB package available on a Digital Equipment Corporation Micro VAX II computer in the department, the data was then processed, and displayed on a three-dimensional mesh plot. The RP-LAB plane formed the base axes of the plot and the edge pixel number formed the vertical axis.

The 3-D plots for the test sample by this method are shown in Figure 4-6. In principle the plots of the edge pixel number would effectively build up the "best" threshold profile for a given image. In practice this is still inconvenient as there was no direct two dimensional view of edge strength information and local average brightness. If horizontal slices are taken through the RP-LAB plane in the 3-D edge pixel plots, then contours are generated for each pixel number level. Figure 4-7 shows 2 contours for the six hole sample image of Figure 4-1.

When 3-D plot and contour techniques were used, selecting edge magnitude and locating the most significant edge region for each "notch setting became relatively easy.

As the "best" thresholding profile selection was the goal of this whole exercise, the strategy was to minimise noise effects and to examine the edge signal magnitude.

To determine the system noise, a black image was used which was achieved by covering the camera lens. The black image was processed by applying a serial "notch" threshold to generate a binary image, and corresponding detected edge pixel data were generated by the pixel counter program automatically. With these edge data, then noise investigation can be achieved by generating the 3-D plot and contours at required pixel number level.

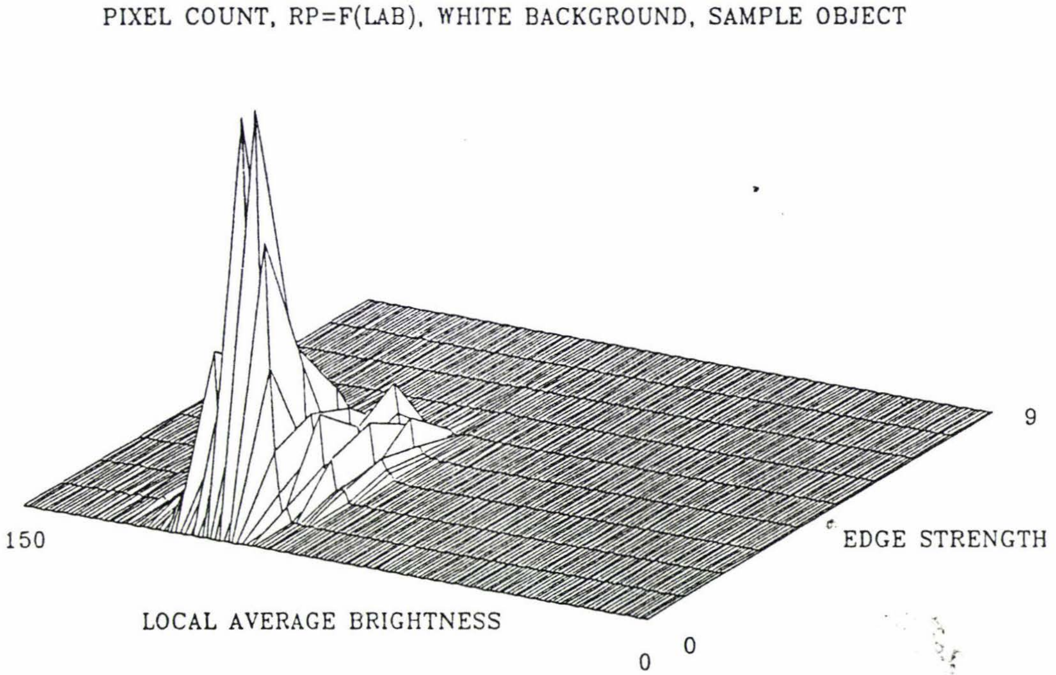


Figure 4-6 Statistically processed six hole sample image 3-D plot

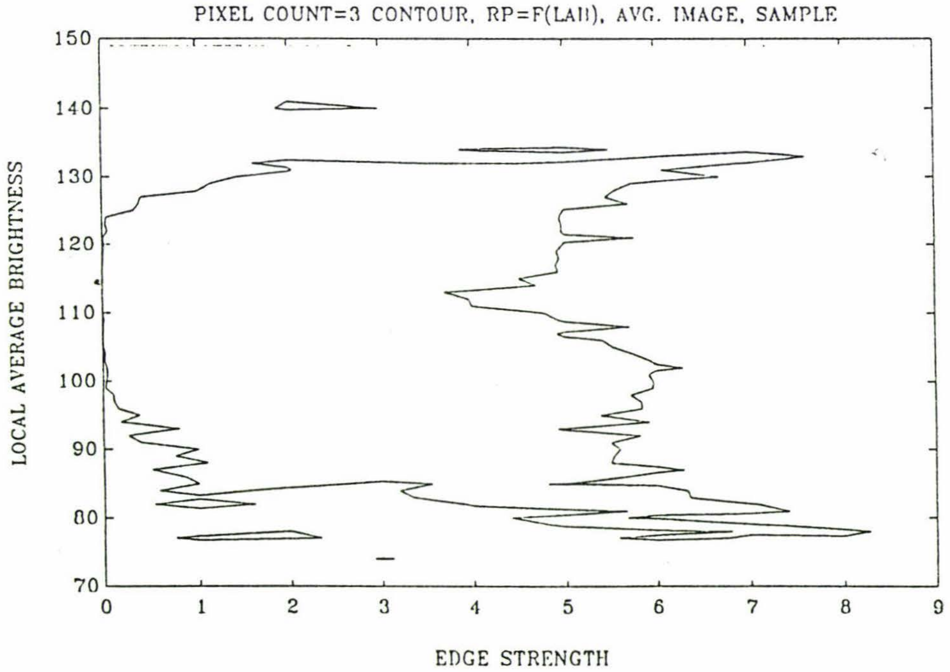


Figure 4-7 Statistically processed six hole sample image contour

To implement the above mentioned task, a multiple function program is required, the program functions should include:

1) Automatic threshold level generation with FLAB, LAB, and W parameters (PIXCON);

2) Send threshold profile to RAM (SNDPRF);

3) Frame grab for the specific threshold profile setting (FRMGRB);

4) Edge picture printing on an EPSON printer (EPSNPV);

5) One complete frame total edge pixel count (PIXCT);

6) Convert double word (32-bit) to decimal ASCII (CONV).

Detailed flow charts are shown in Figure 4-8, and the full program listing is given in Appendix A.

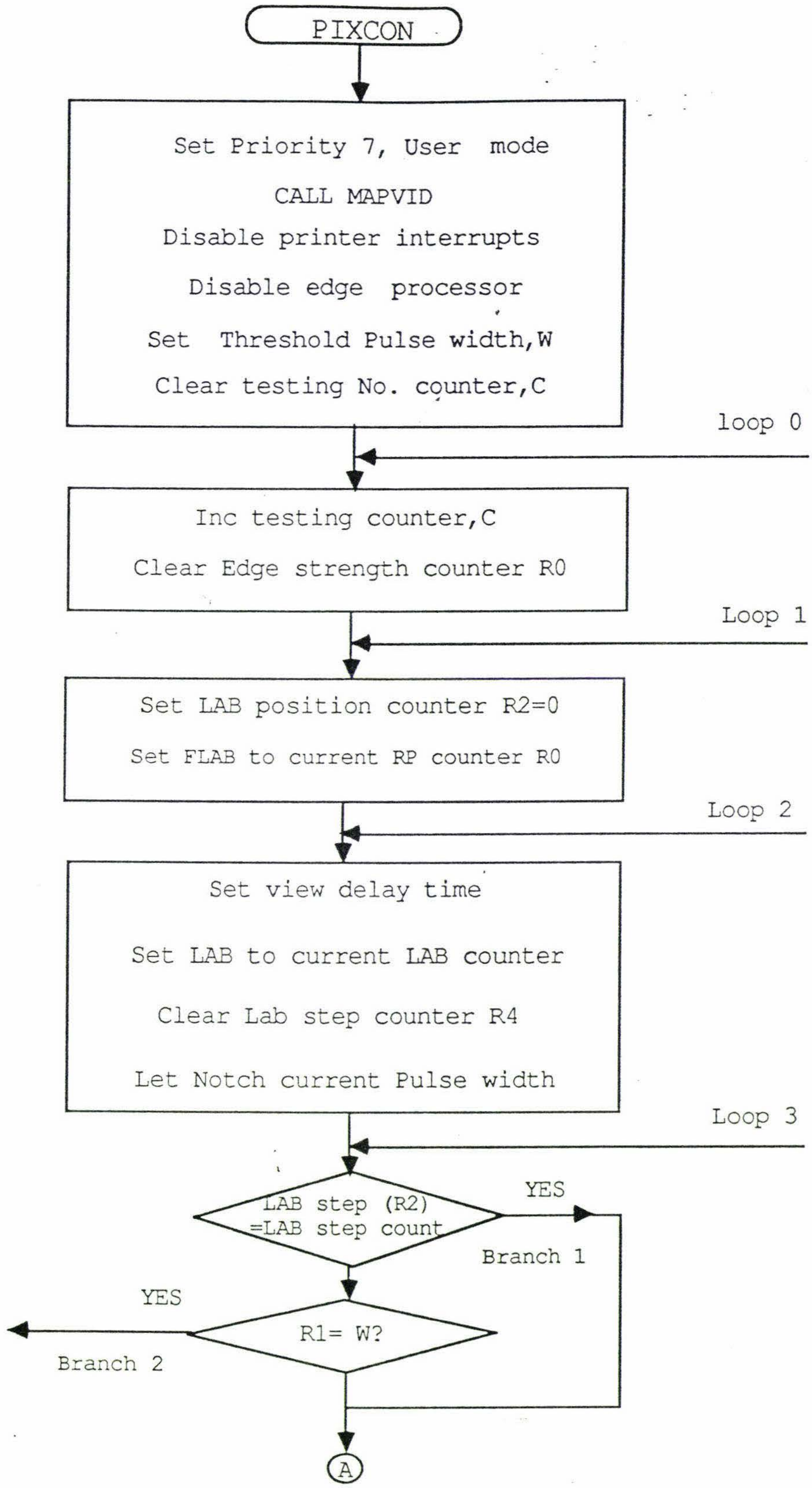


Figure 4-8 (PIXCON)
Edge pixel count master program

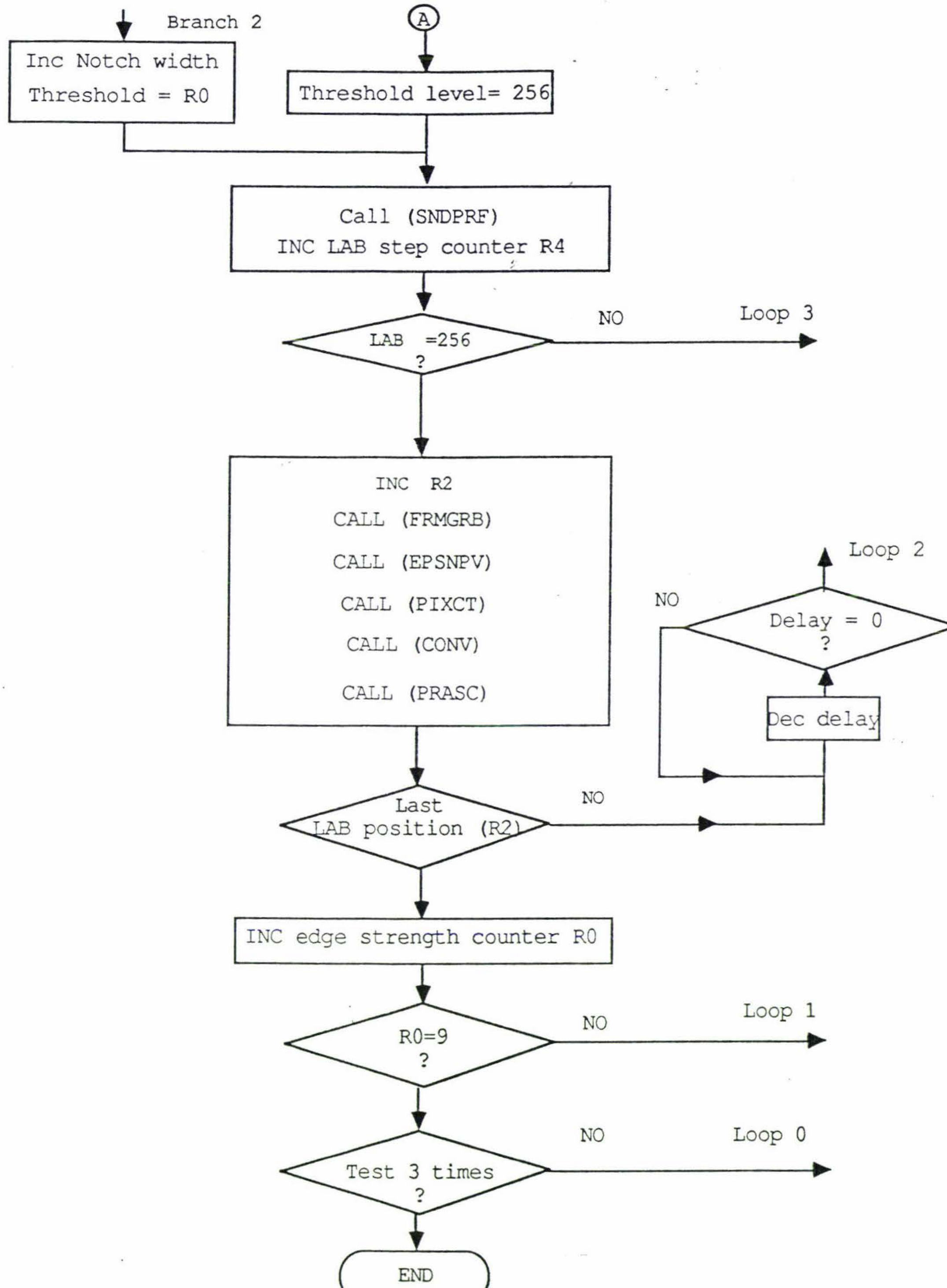


Figure 4-8 (PIXCON continue)

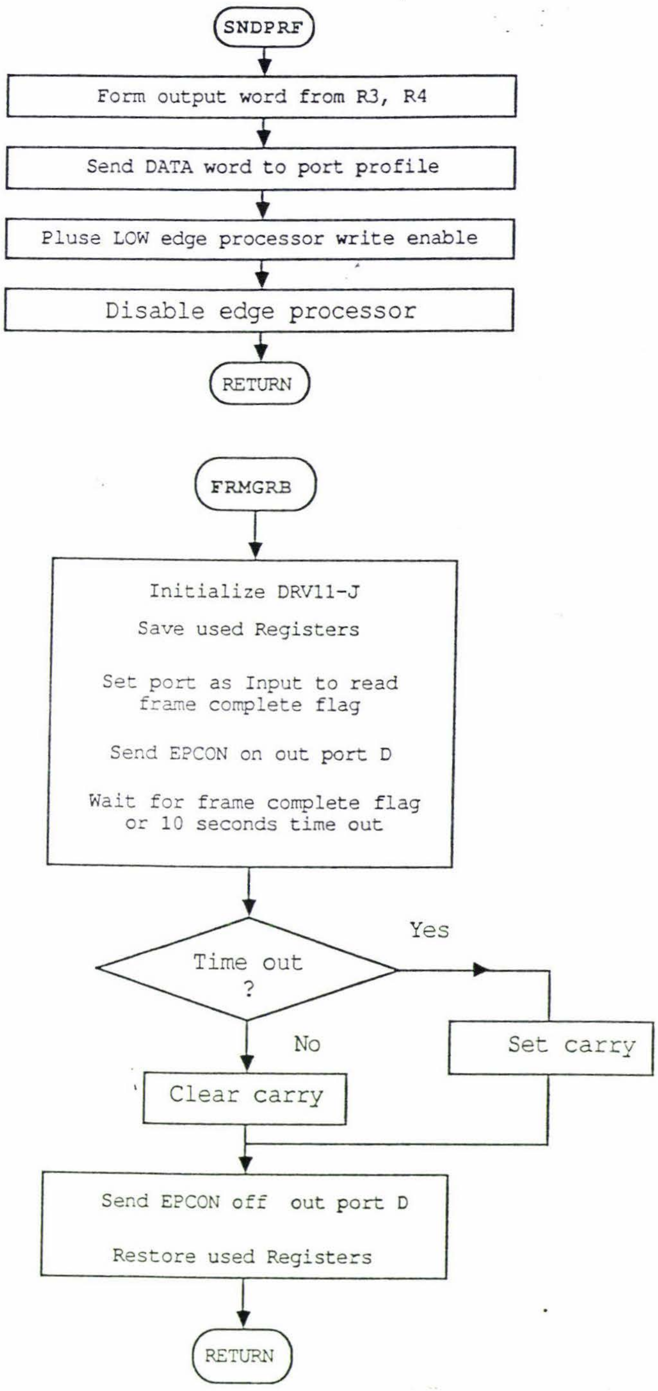


Figure 4-8 (SNDPRF & FRMGRB)
Send threshold profile to RAM and grab a frame

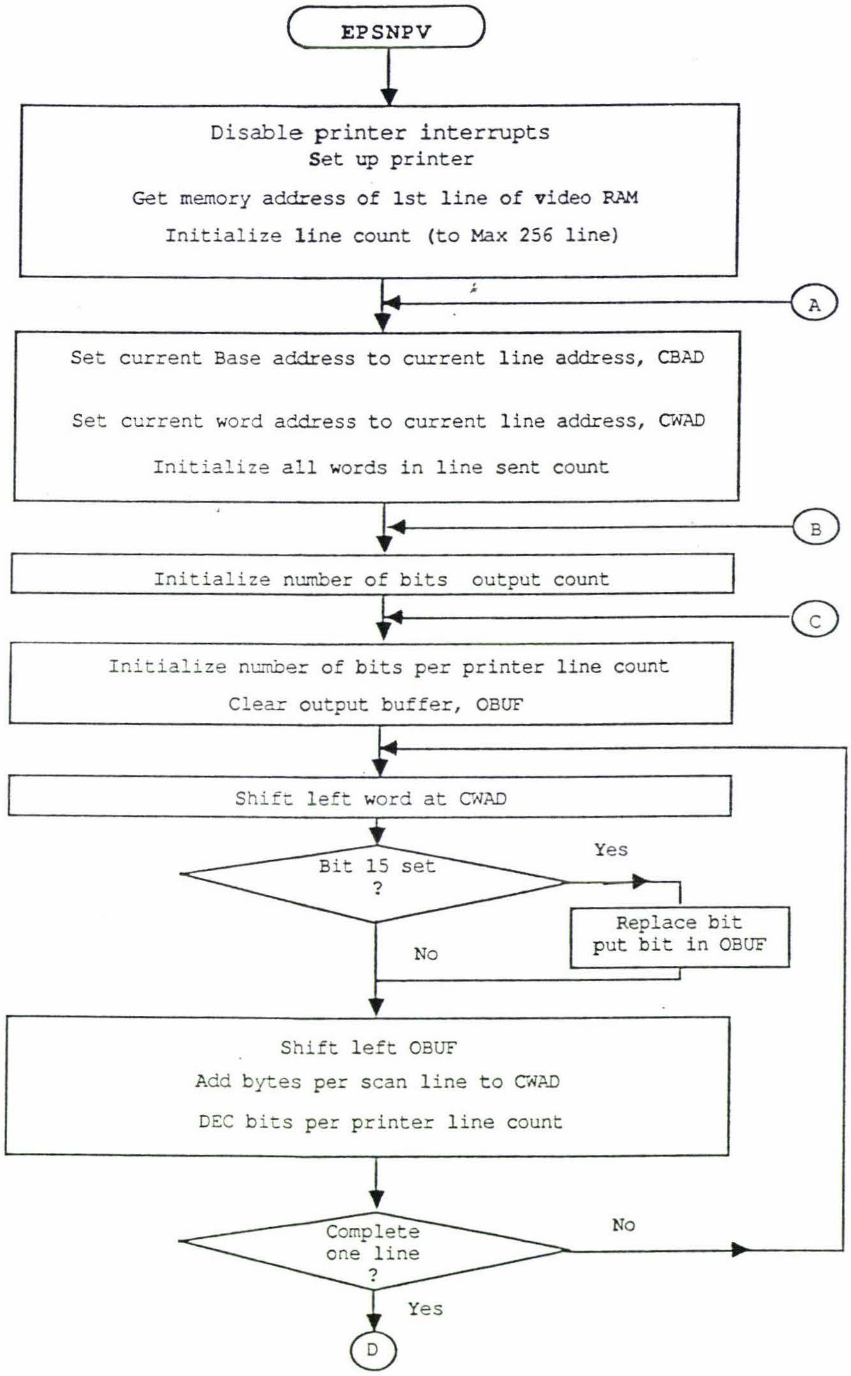


Figure 4-8 (EPSNPV)
Edge picture printing on an EPSON printer

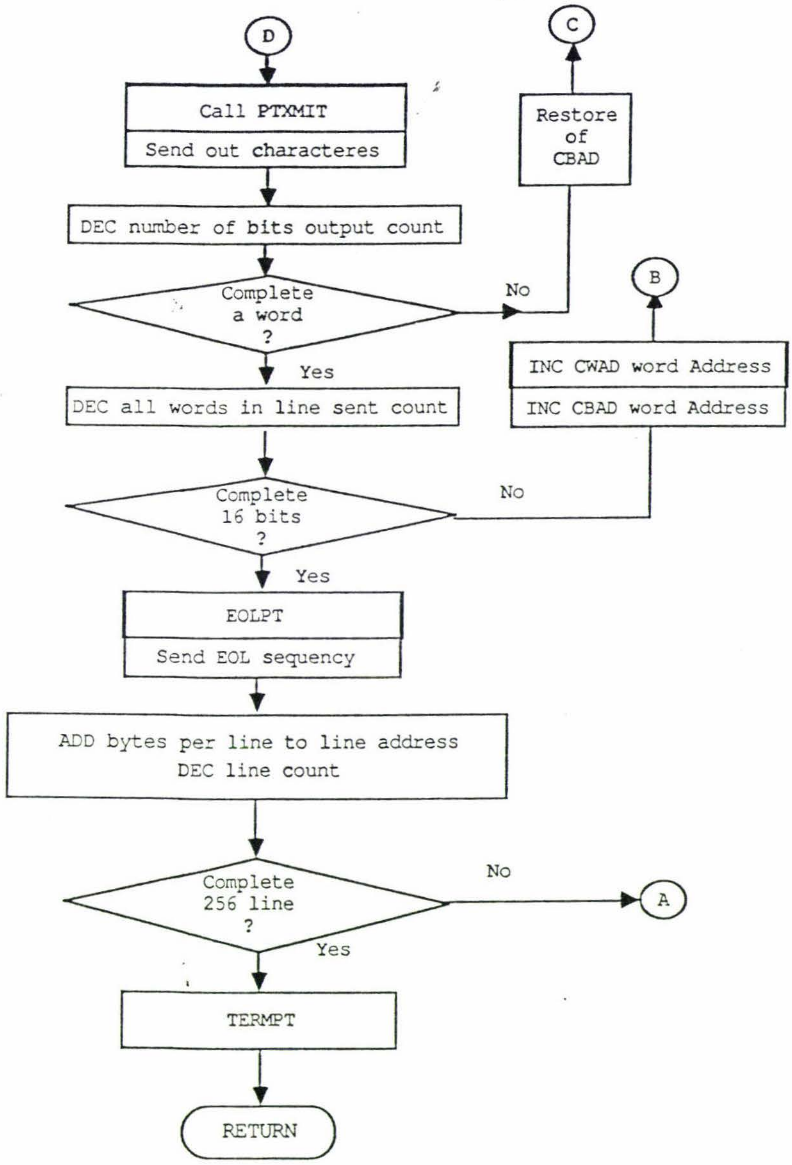


Figure 4-8 (EPSNPV continue)

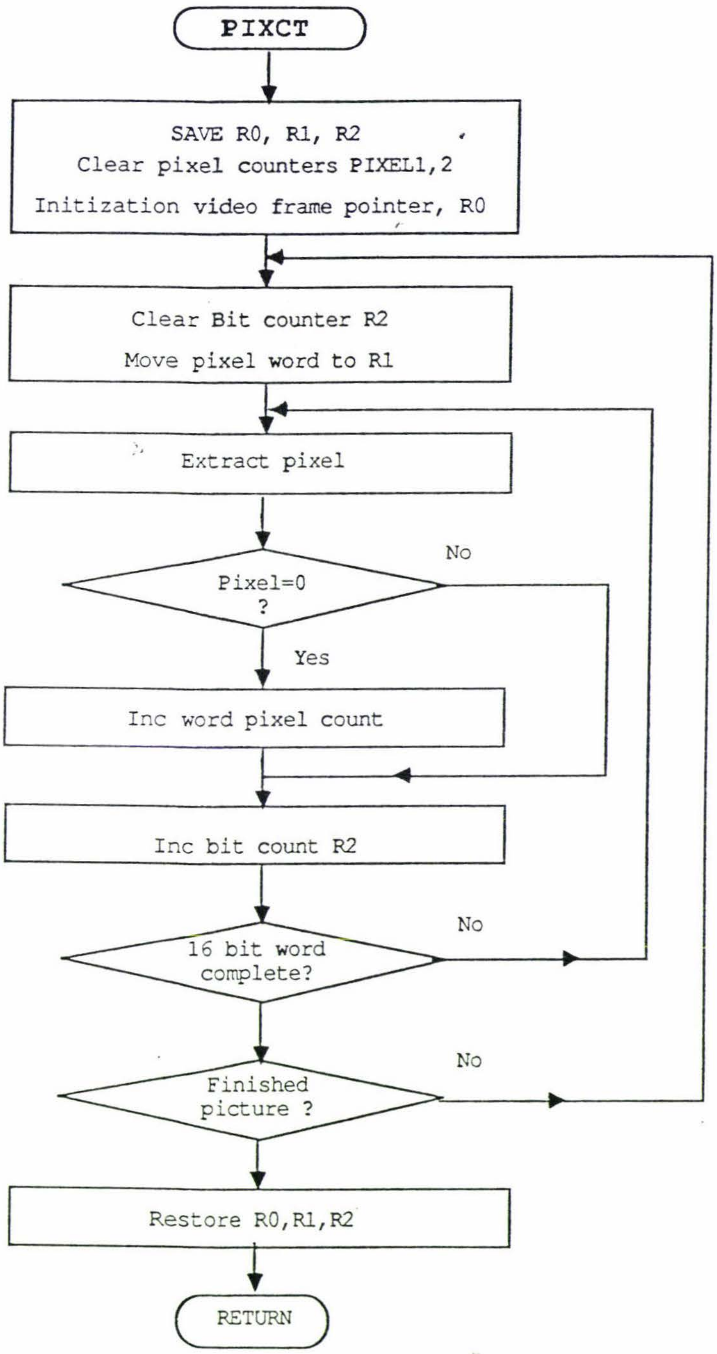


Figure 4-8 (PIXCT)
One complete frame total edge pixel count

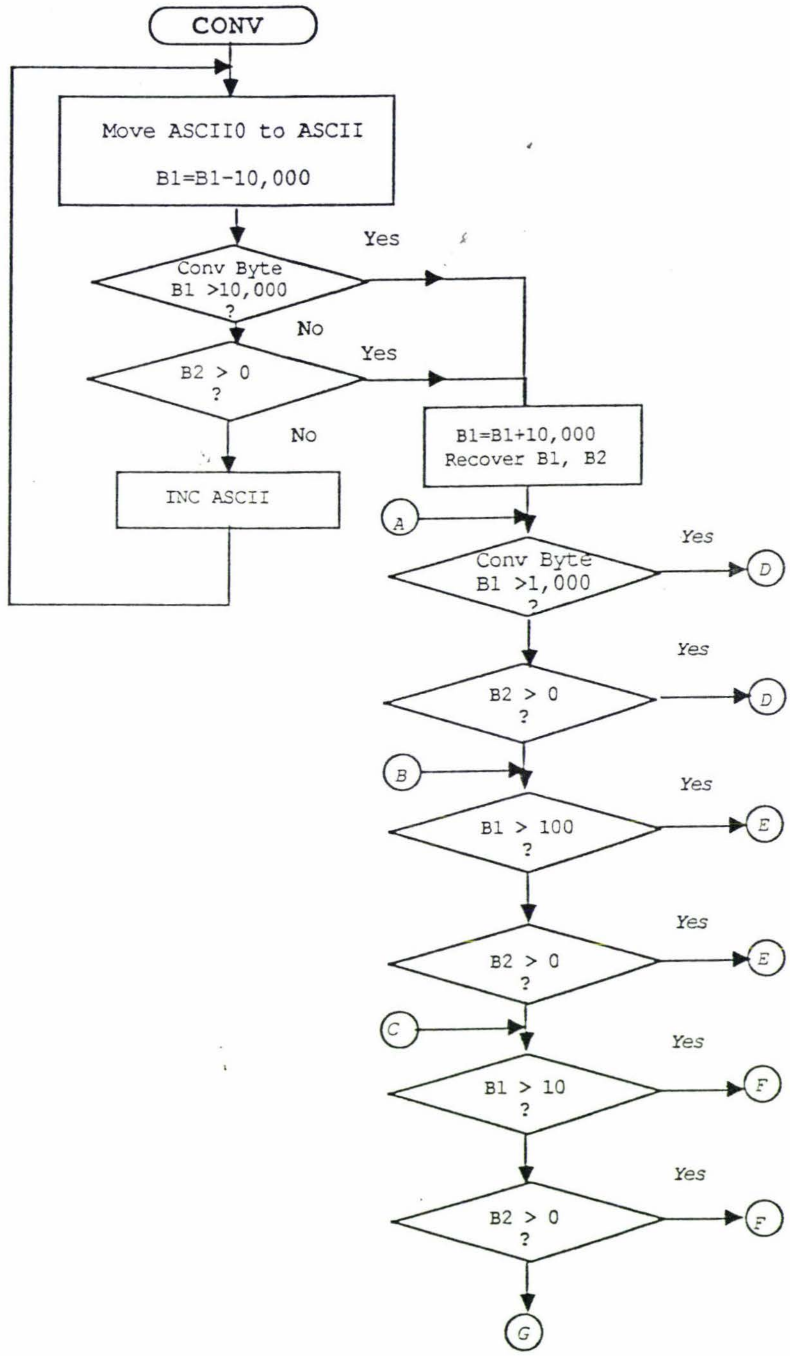


Figure 4-8 (CONV)

Convert double word (32-bit) to decimal ASCII

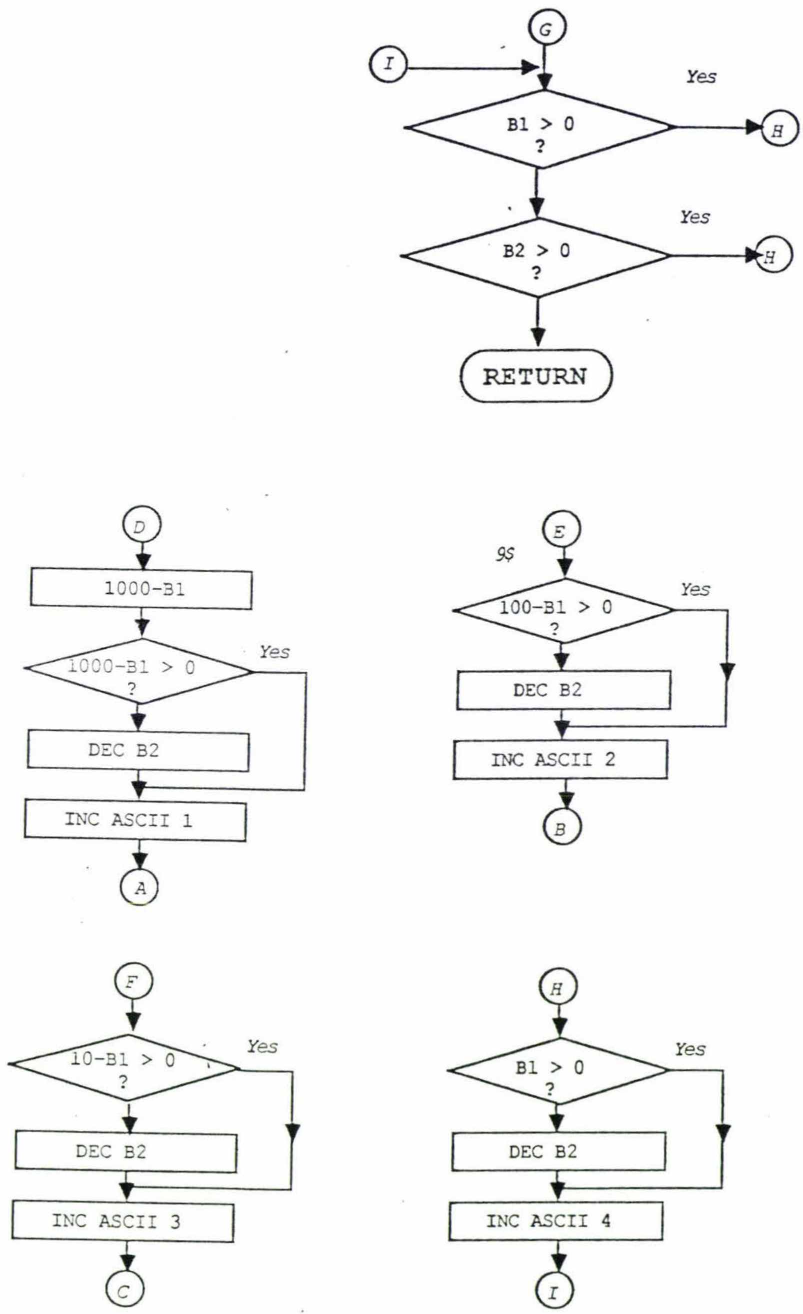


Figure 4-8 (CONV continue)

4.3 3-D contour analysis for selecting threshold

Most vision systems use thresholding at some stage to extract useful information, to reduce the data flow and reduce noise in the image. Because threshold processing achieves a simple form of image partitioning, and simplifies later processing steps many threshold techniques have been presented in the literature [Ros 84] and [Wes 78].

The concept of thresholding is that pixels in a coherent region of the image have an intensity greater than (or less than) a certain value. Thresholding transforms gray-level image pixels into a binary image in which each pixel is either black or white according to the specific point threshold level.

Taking simple flat thresholding as an example, once an intensity threshold is chosen, then pixels whose intensity level is below this threshold are assigned zero value ('black'), and all those above this threshold are assigned maximum value (white').

A numerical method has been developed to evaluate edge signal information and noise, in order to determine the optimum threshold profile for edge processing. Analysis of the edge pixels with 3-D plotting and contours is simple, so system noise distribution can be established without too much difficulty.

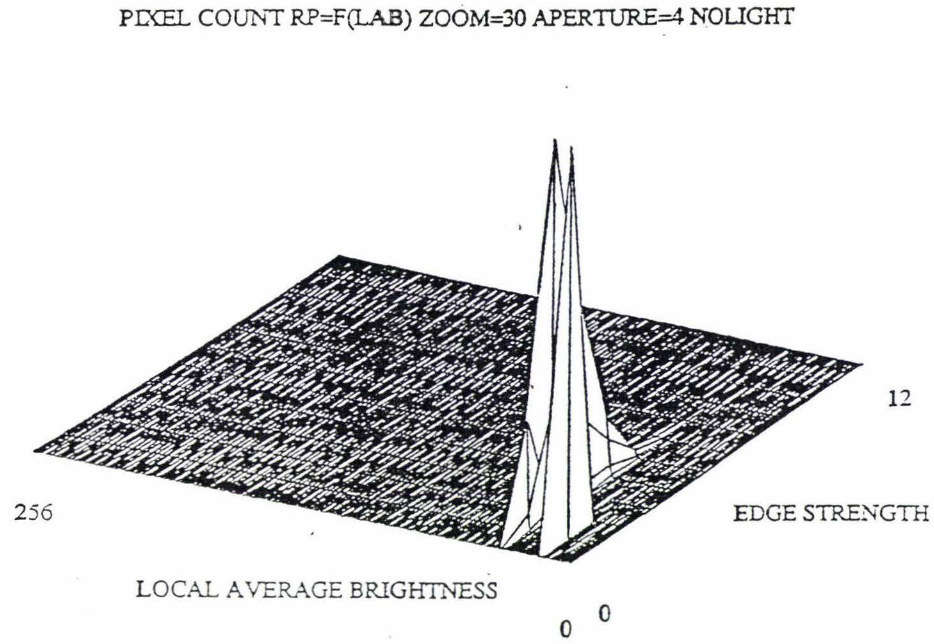
To determine the noise characteristic of the system, two zero edge information images were captured and processed. The first is a black image, and for the second a white paper was chosen.

Figure 4-9 (a,b) and Figure 4-10 (a,b) show the 3-D plots and contour plots of these two images. From Figure 4-9 (a,b) we can see that the black image has one peak edge pixel response occurring at a local average brightness of 40 and maximum edge strength of 8. Figure 4-10 (a) showed clearly, with the white input image the noise region expanding to 5-100 but the maximum edge strength still remains the same as for the black input image at an edge strength of 8.

For the black input image (camera lens covered) which is evenly bright we would expect a zero edge pixel count, but there are significant numbers of noise edge pixels in the system. This noise may arise in the vidicon tube camera or in the associated A/D electronics devices.

Since the image intensities are affected by noise, the choice of optimum thresholding to reduce noise is important. If the threshold is too low at the noise peak region, many spurious edges will be generated by this noise. On the other hand, if it is too high, the sensitivity of the system will be reduced.

The measured noise characteristics do not agree well with a Gaussian noise probability density function. The noise edge contour random shapes indicate that finding a universal thresholding profile from the resulting contour of a fixed pixel level is not very effective. However, the technique does provide useful information about the significant edge sensitive regions and the noise level which is advantageous in helping the researcher to consider the signal and noise relationship in a systematic way.



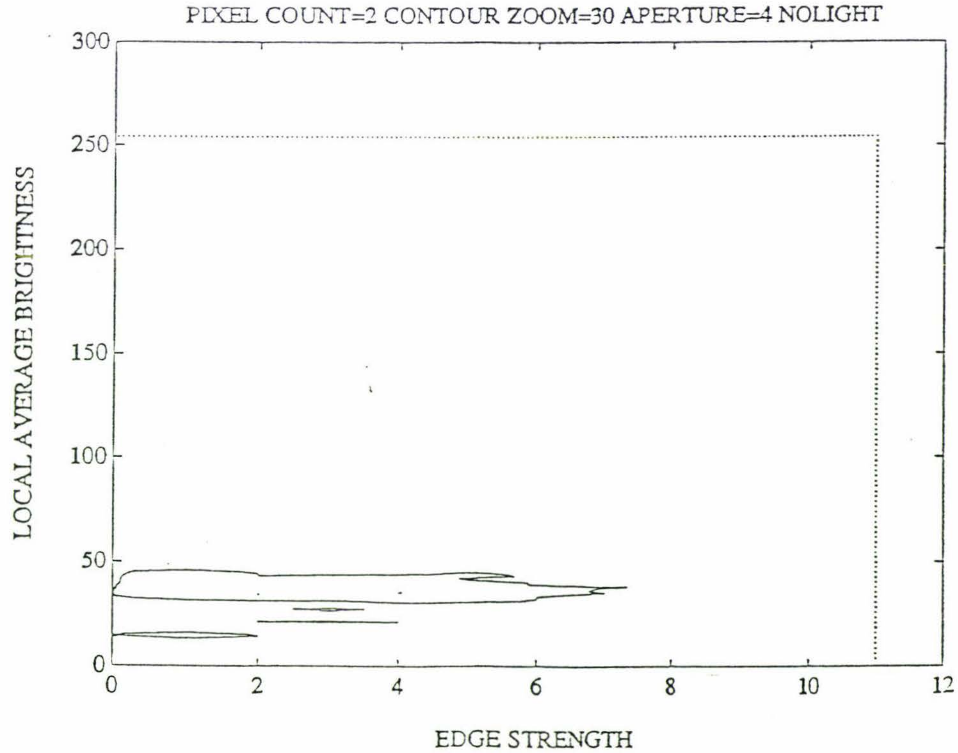
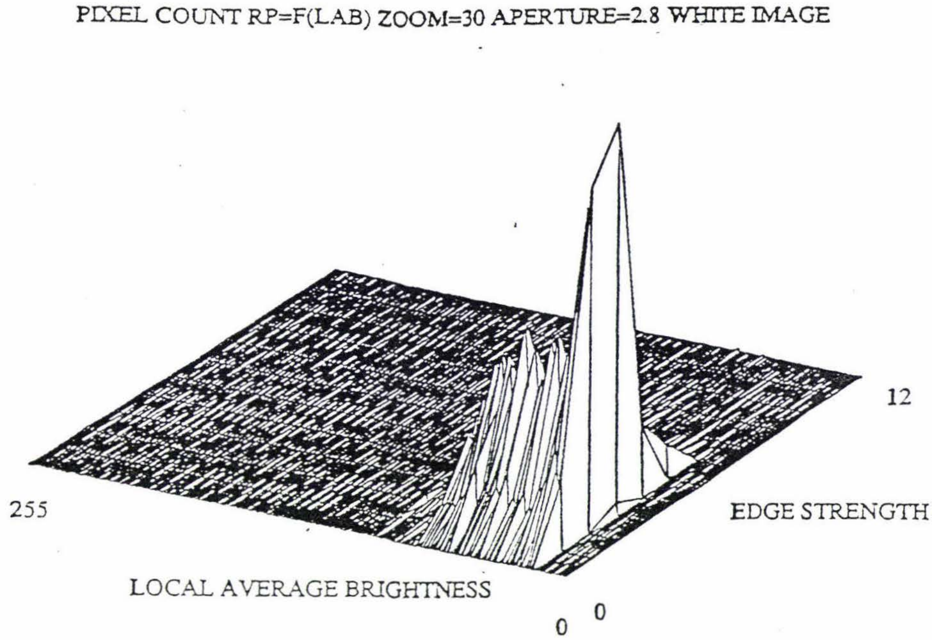


Figure 4-9 (a,b)

black image edge pixel 3D and contour plot



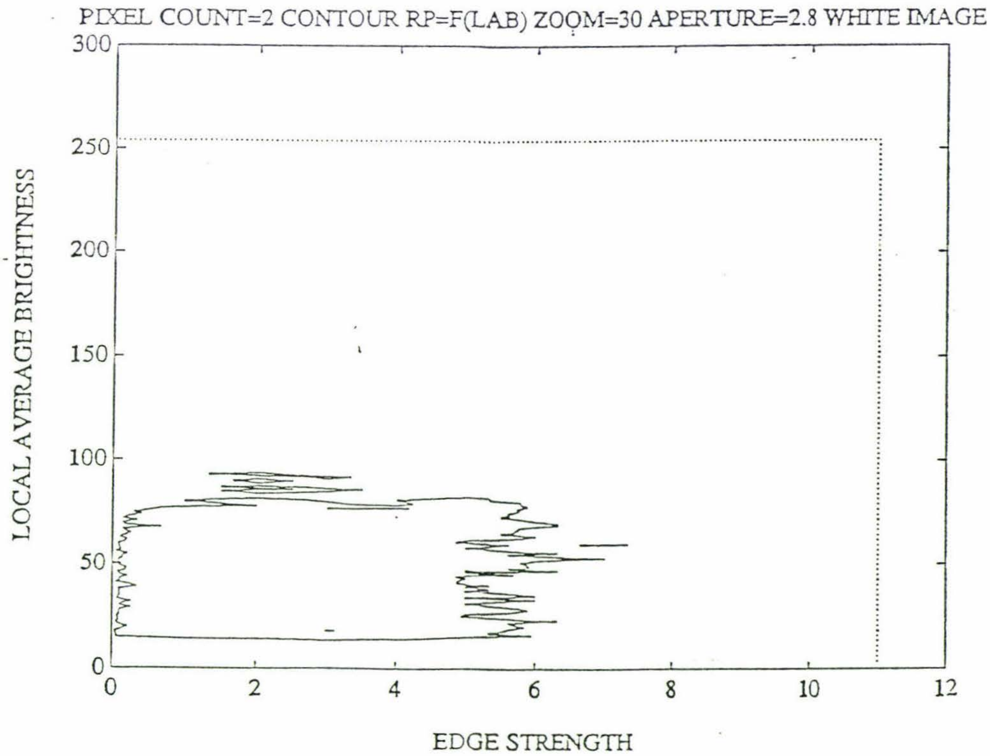


Figure 4-10 (a,b)

White image edge pixel 3D and contour plot

CHAPTER 5

CHAPTER 5 Feature Extraction Experiments

5.1 Experimental arrangement

In this stage, experiments were focused on stationary images, although moving target feature extraction has had a pilot run as reported earlier in a conference paper [Mon et al 89].

The light sources in the experiments are standard fluorescent tubes. To give a general view, Figure 5-1 shows the vision laboratory experimental equipment and its illumination condition. The machine vision laboratory dark window curtain enabled repeatable test runs during the day time.

Figure 5-2 taken from the real-time image processing hardware system TV monitor screen shows the illumination effects of the shadows and diffusion caused by surrounding fluorescent tubes, and the problem of non-uniform brightness distribution on the sample.

It would be ideal to have an appropriate illumination system to simplify the image processing, however in a real situation it is difficult to obtain uniform illumination on the production line such as Figure 5-2 shows.

Research work by Paulsen & McClure [Pau & McC 86], Batchelor, Hill and Hodgson [Bat et al85] indicated that the illumination for an automatic visual inspection application can make the difference between a success and failure. Because, problems such as glinting, shadows and poor object to background contrast can introduce inspection errors in the real-time application.

A poorly designed illumination system can produce glare which may saturate the camera, shadows which may hide defects, low contrast or uneven illumination making the inspection task unnecessarily difficult to perform. So, the illumination system can either enhance the feature being inspected or completely obscure what is actually being searched by the camera.

In developing an appropriate illumination system, consideration should be given to the light source. There are two basic types of light used in automatic visual inspection, incoherent or white light and conherent light which is typically from a laser.

Figure 5-1
Real-time image processing system.

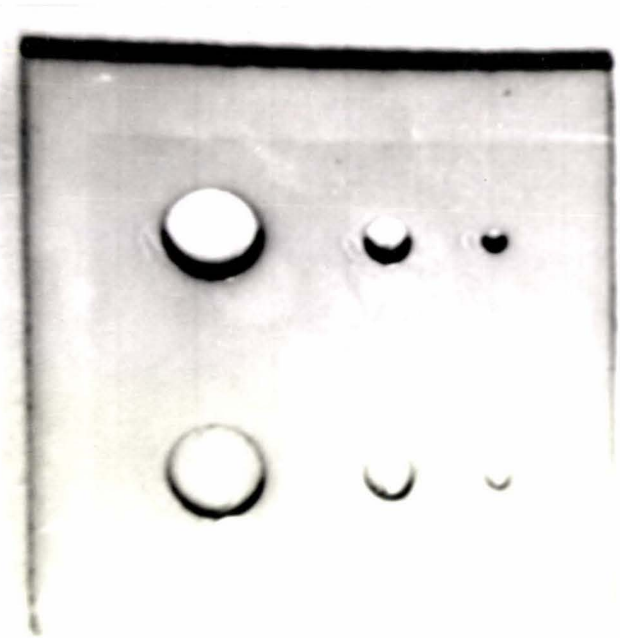


Figure 5-2
Six holes sample picture on TV monitor

Good illumination systems also require a suitable form which can give high contrast for the image information of interest. There are many forms of light for machine vision application, for example diffuse front lighting, directional lighting, back-light, polarised light, and structured lighting.

To obtain high quality images, the level of illumination required may also depend on camera parameters such as f-number, sensor sensitivity, the field-of-view of the system, and the reflectivity of the subject. However, it is difficult to find a general principle for determining the required light level for an application. In practice, the camera should be adjusted by the operator according to personal judgement of image contrast on a TV monitor.

In our experiments the lighting was the standard fluorescent ambient lighting, making feature extraction algorithm development match the realistic production environment.

What has been discussed is only a basic consideration of illumination for automatic visual inspection. Future automatic visual inspection system designers should be aware that to fit a specific application a good illumination system design is necessary.

5.2 A principle for image feature extraction

An extensive literature search was performed to investigate the principles of thresholding for the edge detection. Two principles were located. One is referred to as learning theory [Has & Siy 88]. The method uses linear digital filters as a human learning mathematical model. It starts with an estimate of the background and then the thresholding is achieved by scanning gray-level images on a line by line basis with an updated segment.

Another principle is Weber's law [Hec 24], Weber's law has been discovered many years ago in the field of biology. However, recent developments in understanding and application of Weber's law in machine vision have been reported. Early work is documented in [Mcl & Mon 84] but problems of direct application in machine vision still need further examination.

An analysis is now presented to map the $\Delta I/I$ - I plane of Webers' law to the ΔI - I plane internally used in the real-time hardware system. ΔI corresponds to RP and I to LAB in the hardware system.

Weber's law indicated that human eye sensitivity was a non-linear function of brightness [Pra 78]. The function is illustrated in Figure 5-3. From Figure 5-3, it is clear that in the low intensity region (dark) the human visual sensitivity threshold level $\Delta I/I$ decreases as the intensity I increases; In the middle intensity region, the ratio of $\Delta I/I$ is nearly constant at a value about 0.02 to 0.05; In the high intensity region $\Delta I/I$ increases with increasing I.

To work out the thresholding profile in the RP and LAB plane, assume Weber's law in the indicated three low-middle-bright regions can be represented by three straight lines as shown in Figure 5-4 (a).

In the low intensity region I_0- I_1, the line function would be:

$$\Delta I/I = (A_1 + A_2 I)$$

therefore

$$\Delta I = (A_1 + A_2 I)\ I;$$

where A_1, A_2 are constants but A_2 is negative.

Figure 5-4 (b) is a plotting of a curve of the second degree of I in the ΔI and I plane. This curve is an approximation of the sensitivity threshold profile in the low intensity region. It should be noted that there is an undefined threshold in very low intensity region, so in the ΔI and I plane, the threshold is set to maximum for I less than I_0.

In the middle intensity region I_1 - I_2, this function can be writen as:

$$\Delta I / I = B$$

therefore

$$\Delta I = BI \qquad \text{where B is constant}$$

Figure 5-4 (c) shows the middle intensity region threshold profile as a straight line at the slope of B in ΔI - I plane.

In the high intensity region I_2 - I_3 the line function is:

$$\Delta I / I = (C_1 I + C_2)$$

therefore

$$\Delta I = C_1 I^2 + C_2 I$$

where C_1, C_2 are constant.

Figure 5-4 (d) is the second degree curve in the ΔI and I plane. There is an uncertain threshold in the very high intensity region, so in the ΔI and I plane the maximum ΔI level should be used for $I > I_3$.

The combination of the three lines is a continuous non-linear threshold profile as shown in Figure 5-4 (e).

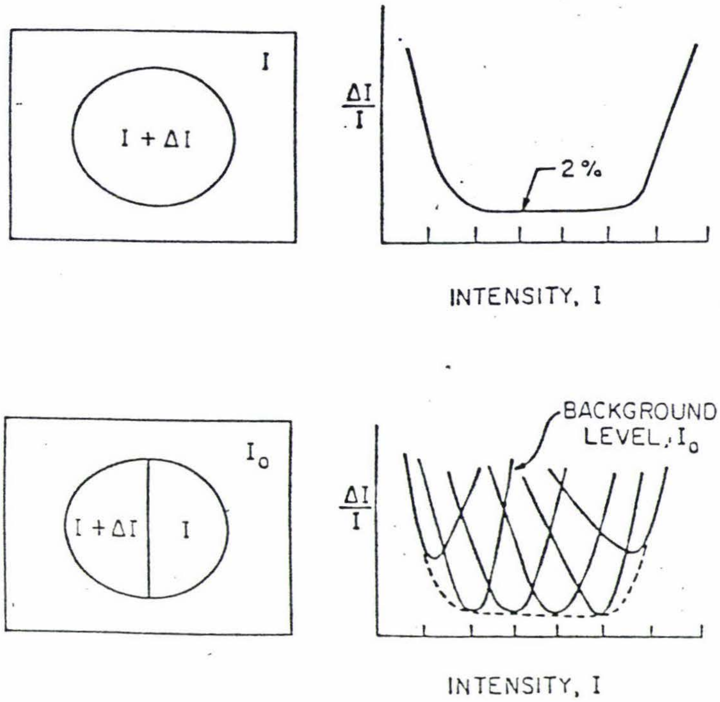


Figure 5--3
Webers' law in $\Delta I/I$ and I plane

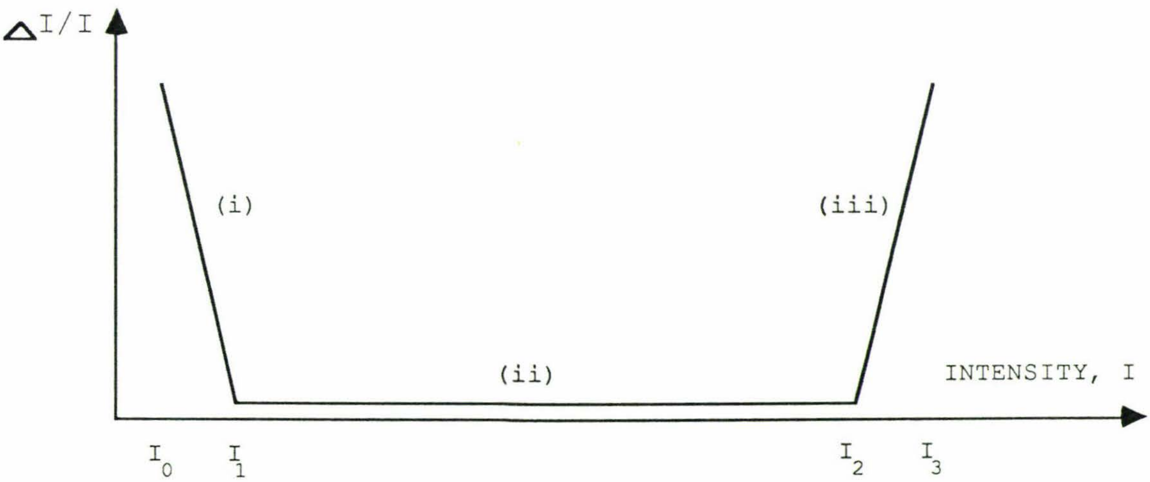


Figure 5-4 (a)
Approximate linear Webers' law in $\Delta I/I$ and I plane

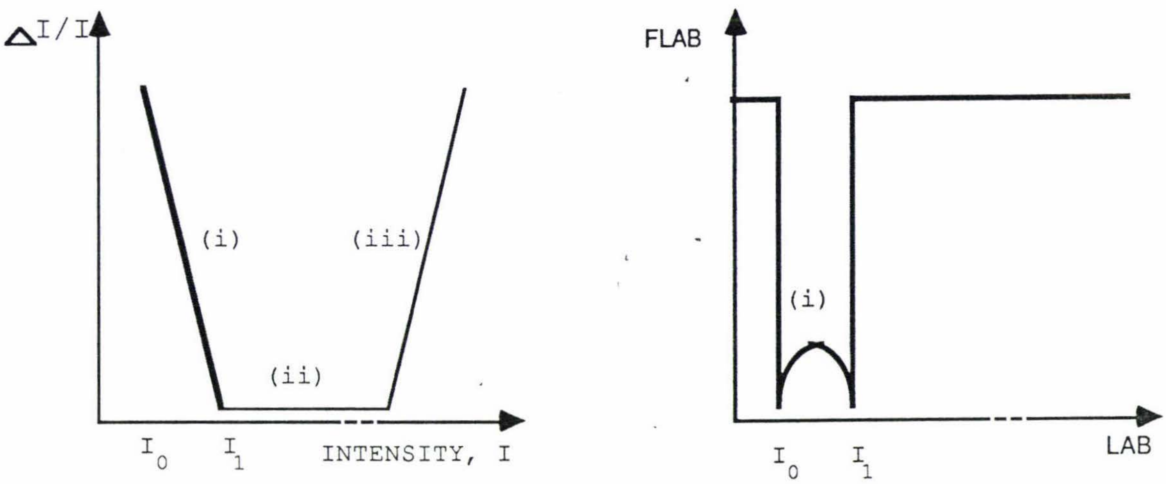


Figure 5-4 (b)
Low intensity region Webers' law in ΔI/I-I and ΔI -I planes

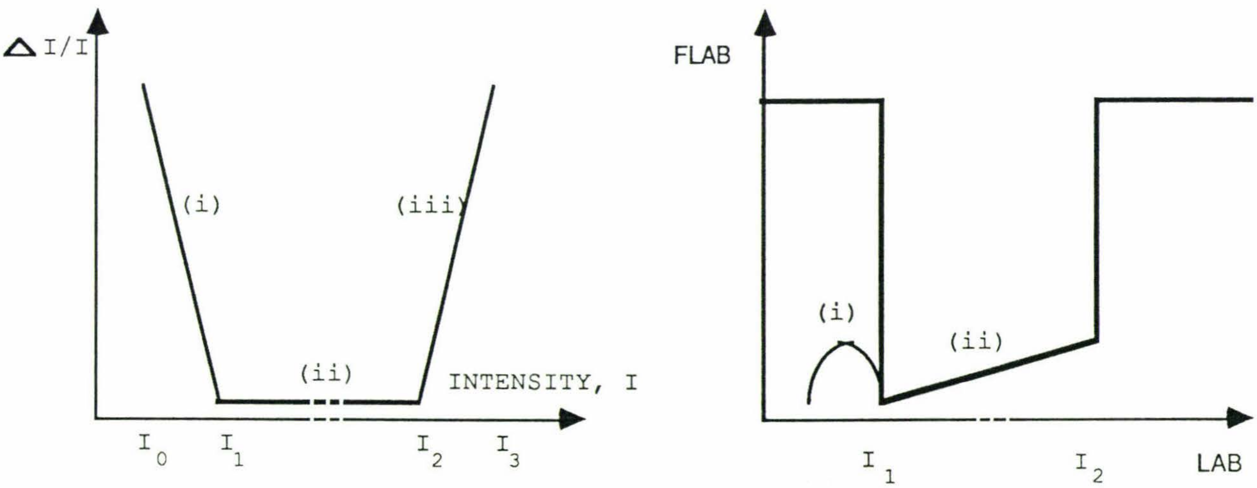


Figure 5-4 (c)
Middle intensity region Webers' law in ΔI/I-I and ΔI -I planes

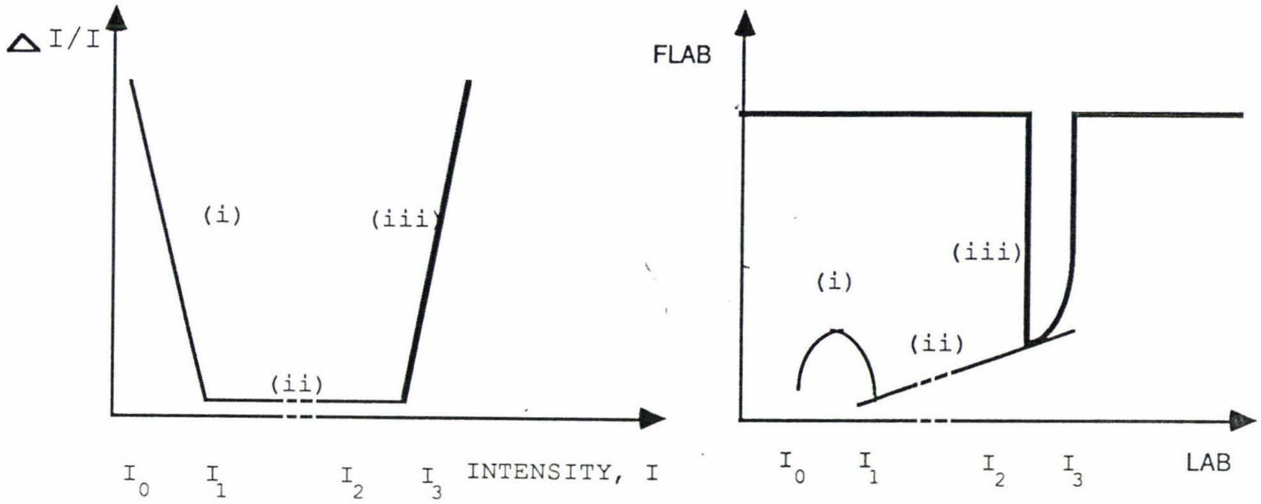


Figure 5-4 (d)

High intensity region Webers' law in ΔI/I-I and ΔI -I planes

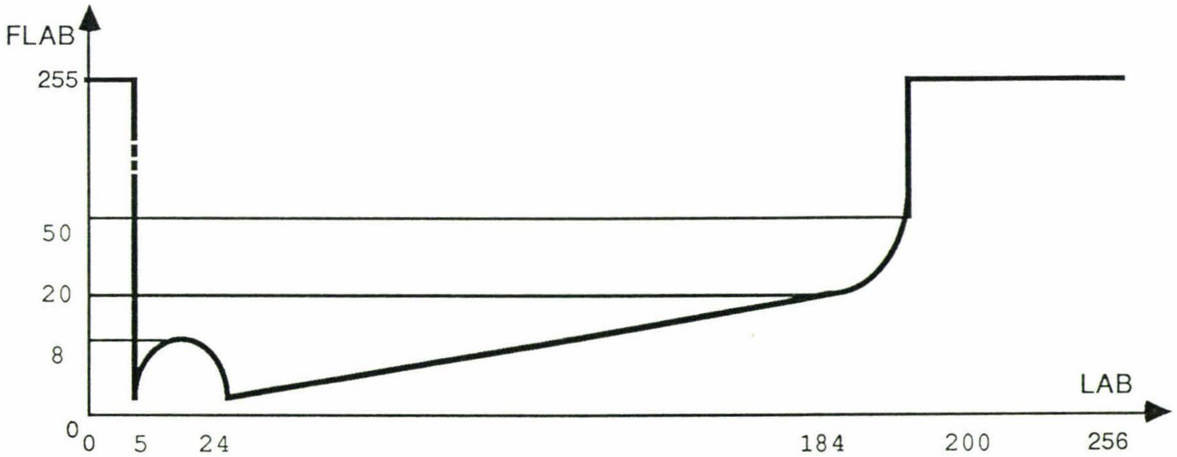


Figure 5-4 (e)

Overall Webers' law in ΔI-I plane

In the plotted profiles on ΔI - I plane, it is clear that in the dark region a W window thresholding profile is like a bandpass filter as shown in Figure 5-5. In the second region the broad U shape threshold is more like a linear filter allowing more information to be registered, as shown in Figure 5-6. In the very bright region, a split V window will perform as a high pass filter to reject noise and only allow significant signal through, as shown in Figure 5-7.

The deduced principle is now very simple, but requires testing on the chosen sample with the standard laboratory illumination. The tested image size is 256x292, processed with the real-time image processing system to generate binary edge pictures.

To demonstrate the non-linear threshold performance, the W shape threshold profile in Figure 5-5 was used to detect dark region edges. Results are shown in Figure 5-8(a). The local intensity range LAB has been chosen from 5 to 24 and intensity change (RP) level varying between 1 and 8. It should be noted that to be confident in choosing parameters, early system noise and edge strength testing has been a vital contribution.

Secondly, a wide window is constructed for LAB from 24 to 183 with the F(LAB) level from 1 to 21 as shown in Figure 5-6. The results in Figure 5-8 (b) demonstrated that the processed image contains both continuous and thin edges.

Finally, the V shape profile LAB from 184 to 199, F(LAB) from 20 to 50 as shown in Figure 5-7 was applied to the test sample. The results in Figure 5-8 (c) indicated that only a few edge pixls were registered but with no noise pixels.

Weber's law has a biological basis, but the principle has application potential in machine vision. Results of this project, demonstrate that non-linear thresholding principles of human origin are capable of good performance machine vision edge processing.

It should be understood that system noise affects the presented results. Such noise may be attributed to the several different optical and electronic devices, so further work may be required to more fully evaluate the behaviour of the principle.

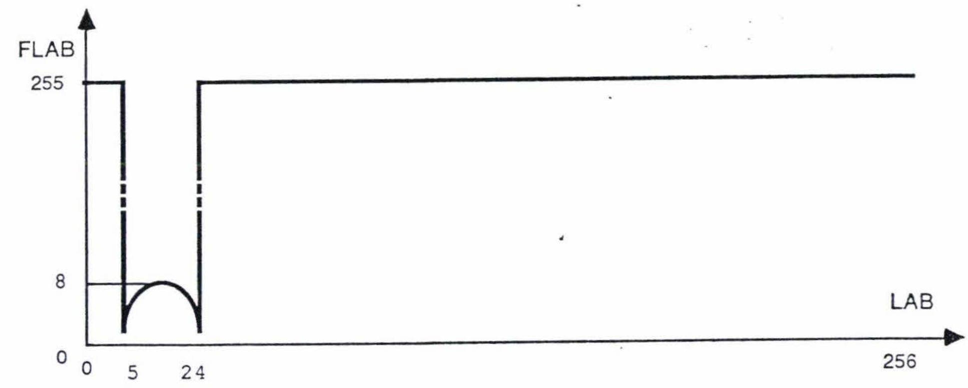


Figure 5-5
Low intensity region W thresholding profile

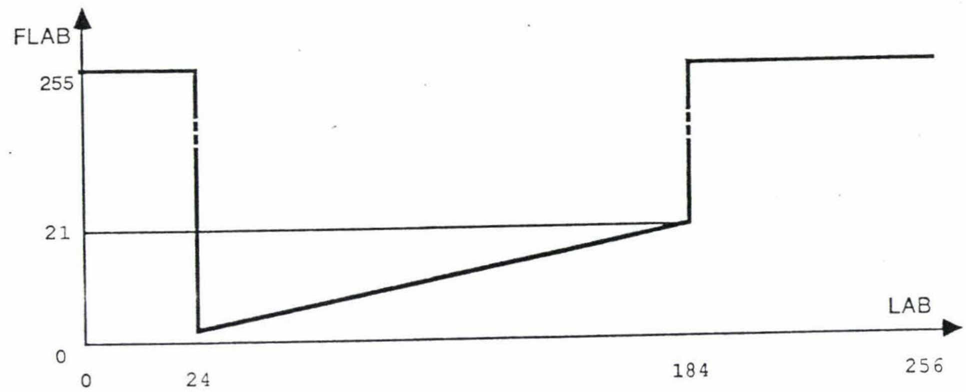


Figure 5-6
Middle intensity region U thresholding profile

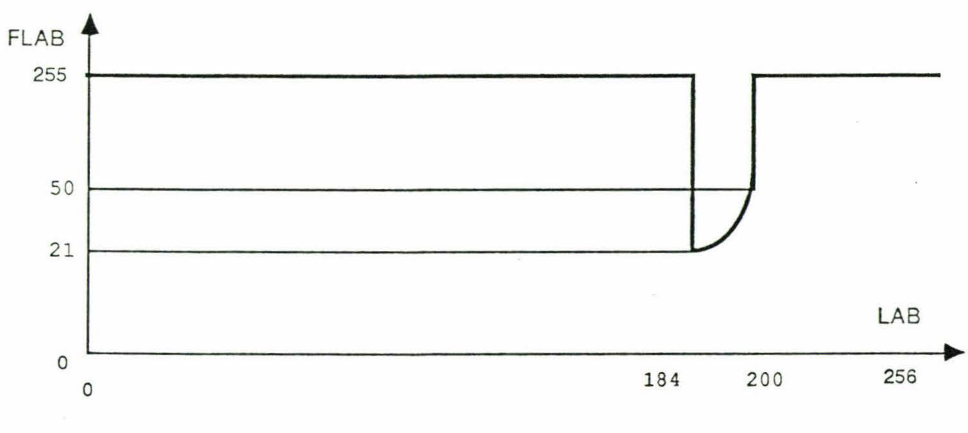


Figure 5-7
High intensity region V thresholding profile

5.3 Image features extraction

The non-linear W, U, V shape thresholding profiles were used for extraction of features in the six hole test sample. The sample is shown in Figure 4-1 (a). The edge detection results are presented in Figure 5-8 (a-c).

The nonlinear threshold profile has demonstrated the ability to select different features in a six hole test image. In the experiment, the W threshold profile of Figure 5-5 can get most edges but they are thick; the U threshold profile in Figure 5-6 registers continuous edges of the full holes only; the V threshold profile in Figure 5-7 will result in true edges but broken in the low brightness region.

From the results, the ability to register features with low noise has showed clearly, however, the overall thresholding profile still has some inability to register weak edges with the test sample, therefore improvement in illumination and the optical system could be a cost effective enhancement.

Results of the experiments have indicated the possibility of using the principle directly to extract general features, and special algorithms can be developed for extracting desired features by combining a priori brightness and edge strength information.

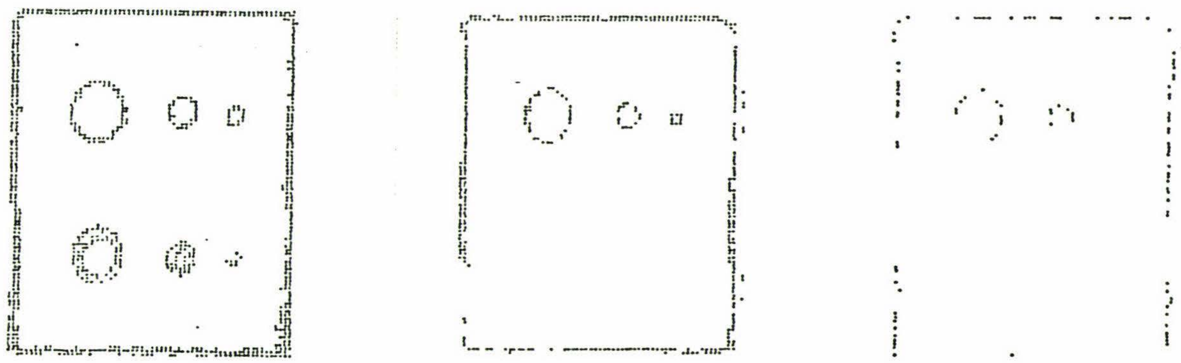


Figure 5-8(a,b,c)
W - U - V thresholding profiles performance

5.4 Discussion & conclusions

It is clear that an appropriate illumination level and good camera is a prerequisite to good image processing. For example if the input image becomes too bright or too dark, the resultant camera image loses or obscures details. In our experimental results, double edges do exist, mainly because ceiling fluorescent tube arrangement cannot give an even lighting of the surface of the sample, so shadows have occurred.

Adaptive thresholding has made the Roberts' operator less sensitive to noise, because it utilises the estimated background noise information in selecting a safe threshold level to minimise noise edge response. It also uses background information such as brightness to select the useful edge data in the image. Therefore, this approach has achieved good feature extraction performance by maximum Signal to Noise ratio in the processing.

The advantages of using deduced W U and V thresholding profiles from Weber's law are firstly they are very simple and have the advantage of low computational expense, and secondly by using different thresholding profiles not only can the noise be reduced, but also the different brightness edges can be separated.

Two possible approaches to desired feature extraction have been considered, the first one uses the different brightness ranges to isolate the desired and unwanted information. Another approach was to utilise the edge strength and the defined thresholding level to select the high contrast edges only. However, very low contrast edges detection would need a more knowledge based algorithm to find true edge from noise images.

In conclusion, the new approach is simple, and enables machine vision system development to follow the principle of imitating the natural system. Feature extraction by using the Roberts' operator and W U V thresholding profiles has advantages for edge detection. The principle and experiments have demonstrated that the simple Robert's operator and non-linear thresholding would be an appropriate approach in high speed feature detection applications.

CHAPTER 6

CHAPTER 6 Comparison With Other Edge Detection Results

To demonstrate the performance of thresholding for edge detection, an extensive comparison of different operators is presented. This work requires a wide range of operations and the system must be highly interactive. These requirements are beyond the developed real-time hardware image processing system design capacity. "VIPS", a VAX-based image processing system originally developed at the University of Canterbury, New Zealand has provided an extremely flexible interactive tool.

6.1 Environment in VAX Image Processing System

VIPS, standing for VAX Image Processing System. It was developed by a group of applied image processing specialists led by Dr. R M Hodgson.

The group [Bai 85], [Bai & Hod 85], [Hod et al 87] have demonstrated that VIPS has considerable ability in automatic visual inspection algorithm development. The VIPS system includes a Micro VAX host computer as the central processing unit to execute the software algorithms for various operations.

Two main peripheral devices of VIPS system are:

* A terminal which is the interface between the user and the system.
* A DMA interface (Currently using a Digital Equipment DR11W) for fast image data transfer between the host computer and the image capture and display subsystem.

The image capture and display subsystem is a Matrox MIP-512 image processing board. The programmable image resolutions are: 128x128 pixels, 256x256 pixels and 512x512 pixels. The image is standard 8-bit grey-level. The colour monitor is a 14 inch high-resolution type.

VIPS system terminal, colour monitor, camera and the lighting facility is shown in Figure 6-1. Because a CCD (charge coupled device) camera and adjustable lighting are used in VIPS, with high computing power there is a considerable saving in noise removal pre-processing.

VIPS provides a wide selection of image processing commands and operators. Information on all of these commands and their associated parameters is available on-line through the HELP command, or by using a special HELP key on the terminal.

VIPS is a command-based interactive image processing system. VIPS enables commands to be developed in a high-level language such as PASCAL.

Figure 6-1 VIPS system

6.2.1 Roberts' operator performance

Roberts' operators implemented in VIPS system utilises 2x2 windows. The operator is an enhancement/threshold operator. The input image and processed edge image with a simple estimated magnitude threshold scheme are shown in Figure 6-2 (a-d). Simple flat thresholds are applied again at the levels of 1 and 2 respectively in Figure 6-2 (c-d).

To evaluate thresholding, three sets of different threshold scales, 4-6-8-10, 15-20-25-30 and 40-50-75-100 were applied to the Roberts' filtered image. From the results of Figure 6-3 (a-p), it is easy to see the tendency of noise reduction with increasing threshold level. There is a noise free threshold level from 10 to 15, which agrees well with the test results of Figure 4-7 where the noise edge strength is 7. The results also indicate that with increasing threshold weak edges are missed too, but detected edges are thin.

A square 3 by 3 window linear convolutional smoothing filter was utilised in smoothing processing. Weightings used in the 3 by 3 mask are 1,1,1,1,8,1,1,1,1. As demonstrated in Figure 6-4 (a-b), there is a significant reduction of noise in the final result, but smoothing also causes blurring. Therefore, some enhancement filter may required.

By using different filter in image processing as far as the output is concerned, smooth filtering may reduce noise significantly, and enhancement filter may improve edge result well, but the cost is more processing time required, and an increased trend to loss true edges.

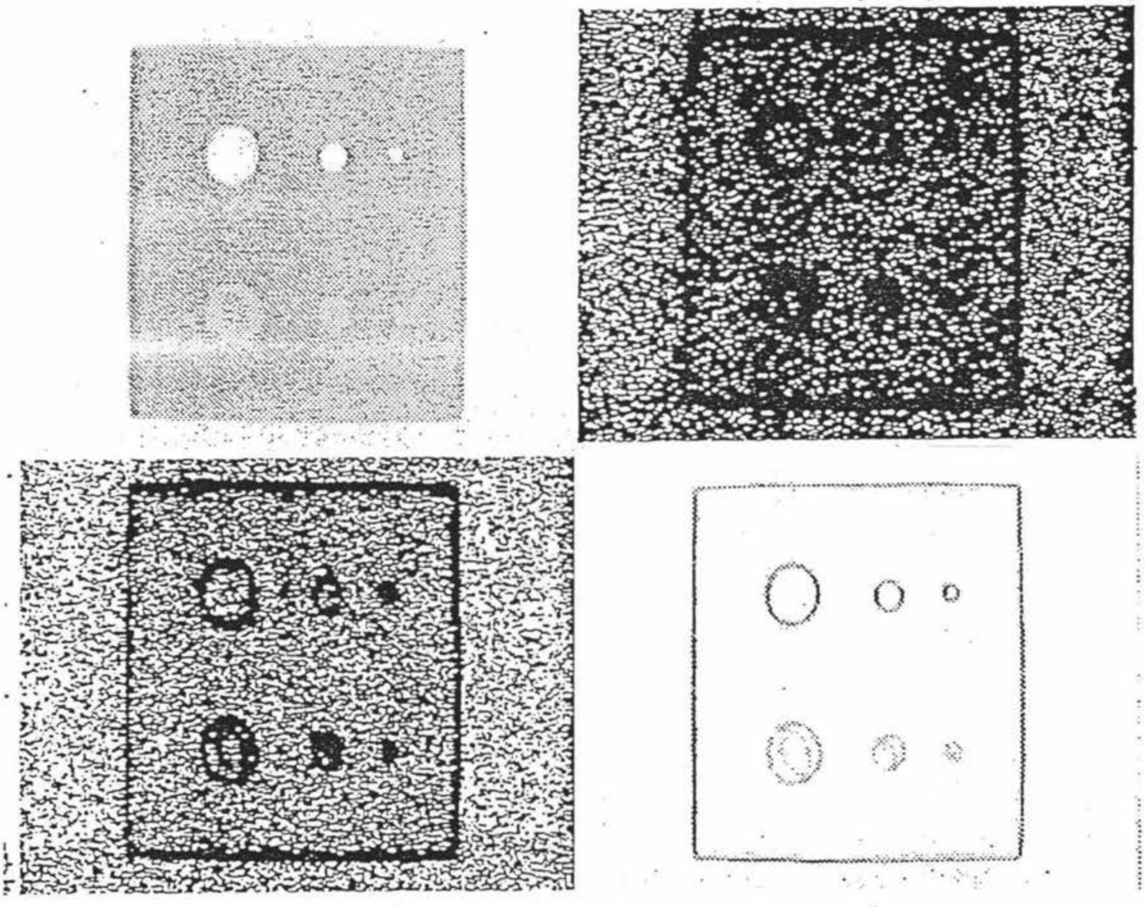


Figure 6-2 (a-d)
(a): Input image (b): Threshold at 1
(c): Threshold at 2 (d): Threshold at 25

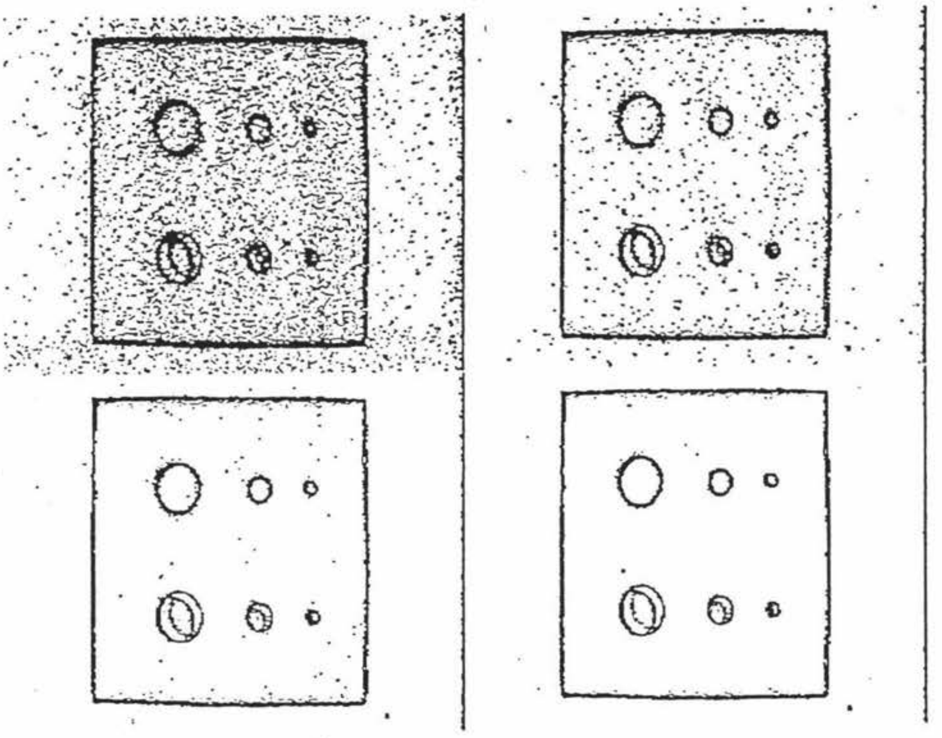


Figure 6-3 (a-d)
(a): Th=4; (b): Th=6; (c): Th=8; (d): Th=10

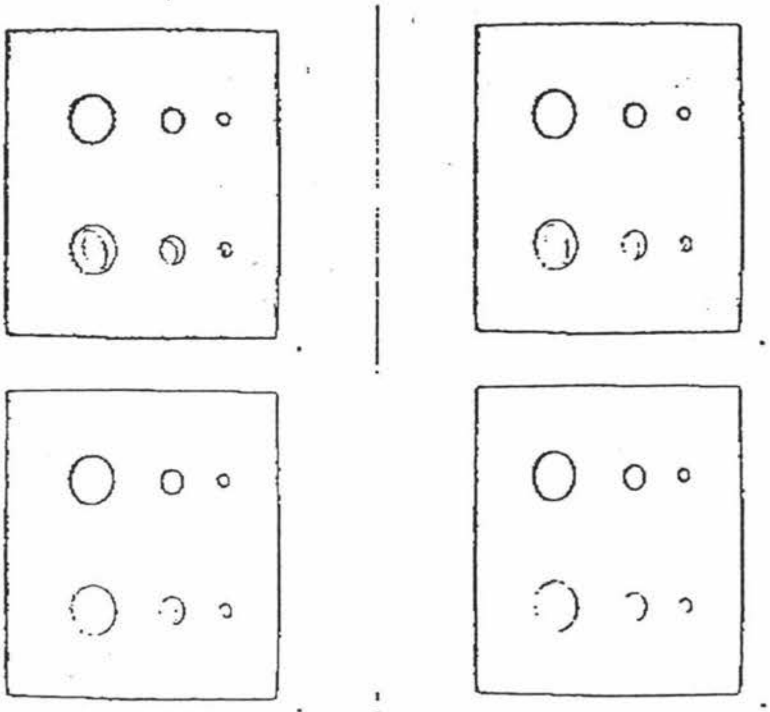


Figure 6-3 (e-h)
(a): Th=15; (b): Th=20; (c): Th=25; (d): Th=30

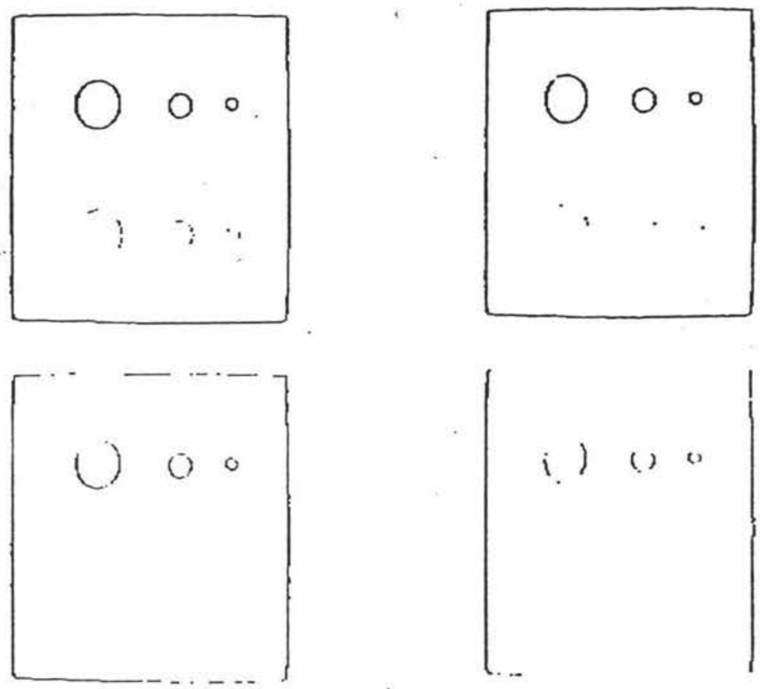


Figure 6-3 (i-l)
(a): Th=40; (b): Th=50; (c): Th=75; (d): Th=100

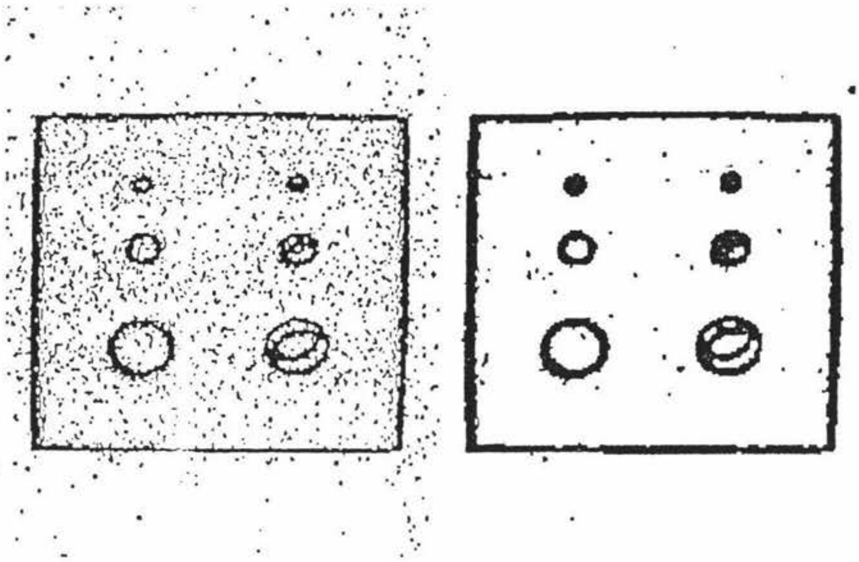


Figure 6-4(a-b) Linear filter noise reduction
(a): RP filtered image; (b): RP filtering Smoothed input image;

6.2.2 Sobel operator performance

The 3 by 3 masked Sobel operator was applied to the input image. This operator is an enhancement/threshold operator. An input image and processed edge image with a simple estimated magnitude threshold scheme are shown in Figure 6-6 (a-b). The resulting performance with different thresholds of Sobel edge detection is shown in Figure 6-6 (c-l).

Results show that edges become thin as the threshold increases from 5 to 50. Compared with Roberts' operator, the Sobel operator is less noise sensitive, as shown in Figure 6-7 (a-b). By using the same 3 by 3 linear filter as reported in 6.2.1, weightings in the mask of 1,1,1,1,8,1,1,1,1. there is a tendency to blur edges.

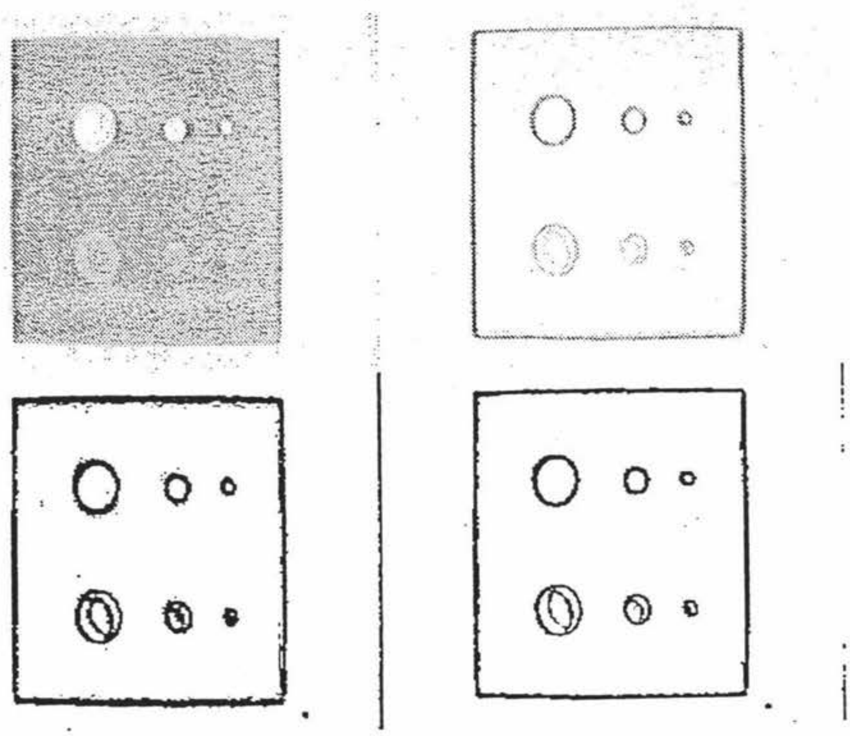


Figure 6-6(a-d) Sobel operator edge detection
(a): Input image; (b): Sobel filtered image;
(c): Thresholding at 5; (d): Thresholding at 10.

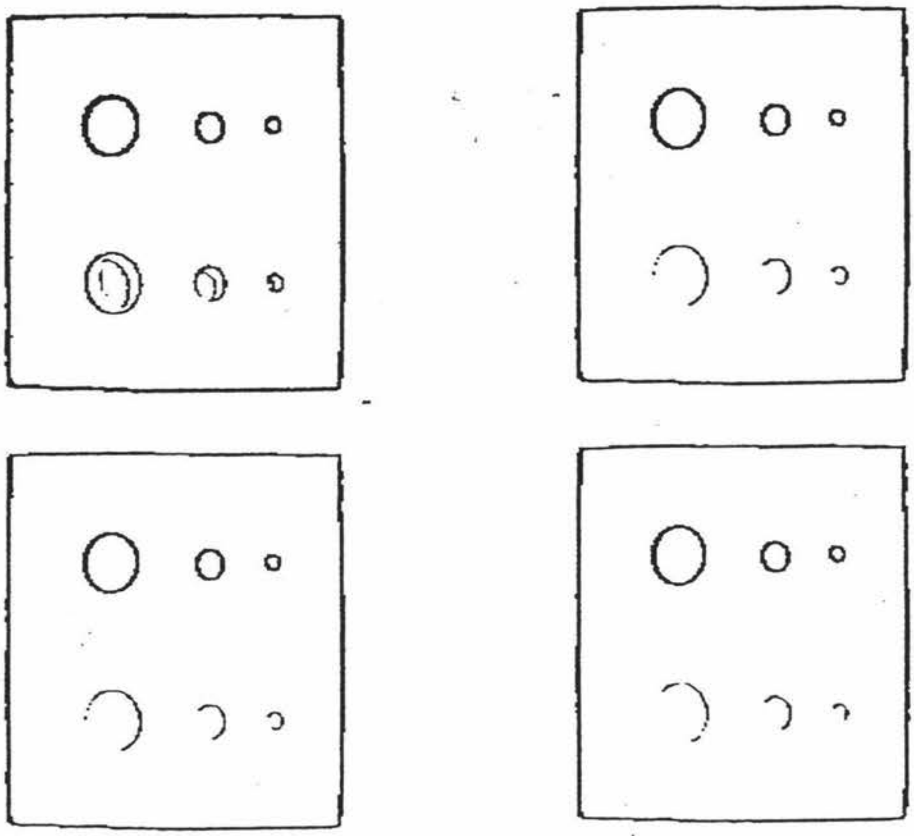


Figure 6-6(e-h) Sobel operator edge detection
(e): Thresholding at 15; (f): Thresholding at 20;
(g): Thresholding at 25; (h): Thresholding at 30.

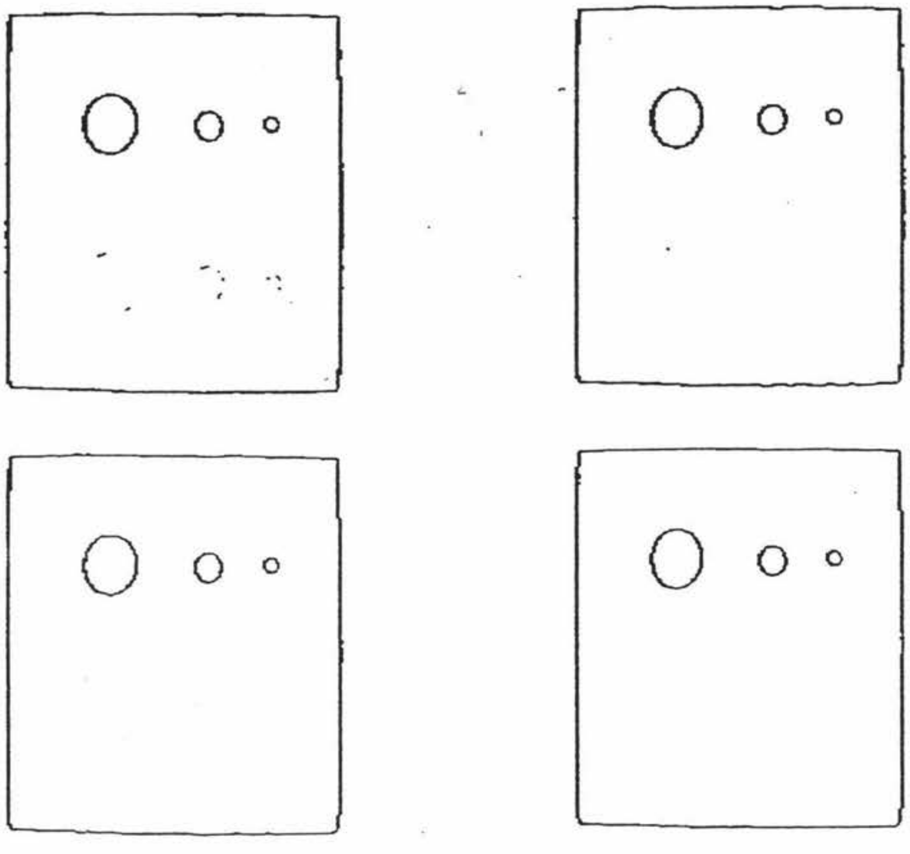


Figure 6-6(i-l) Sobel operator edge detection
(i): Thresholding at 35; (j): Thresholding at 40;
(k): Thresholding at 45; (l): Thresholding at 50.

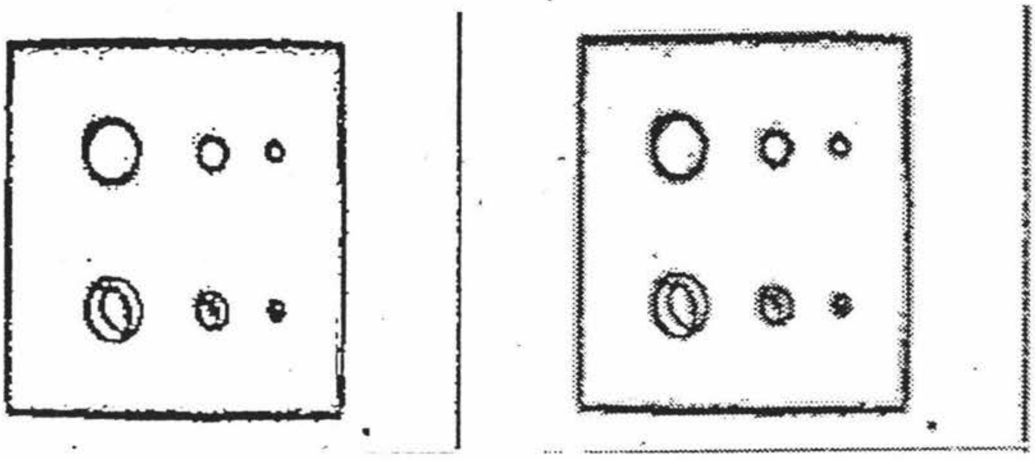


Figure 6-7(a-b) Sobel operator with linear filter
(a): Sobel filtered image; (b): Smooth Sobel filtered image;

6.2.3 DIP operator performance

The difference of inverse probabilities (DIP) operator utilises local brightness information as reviewed in chapter 2.1.3. Figure 6-8(a-d) shows the result of using the DIP operator at threshold level 1 and 2.

In Figure 6-8(c-j), different thresholds from 5 to 40 were applied. The results have shown that the DIP operator is very sensitive in background noise, but it extracts edges more thinly than the Sobel operator. Again, if the threshold is too high blind hole edges are lost.

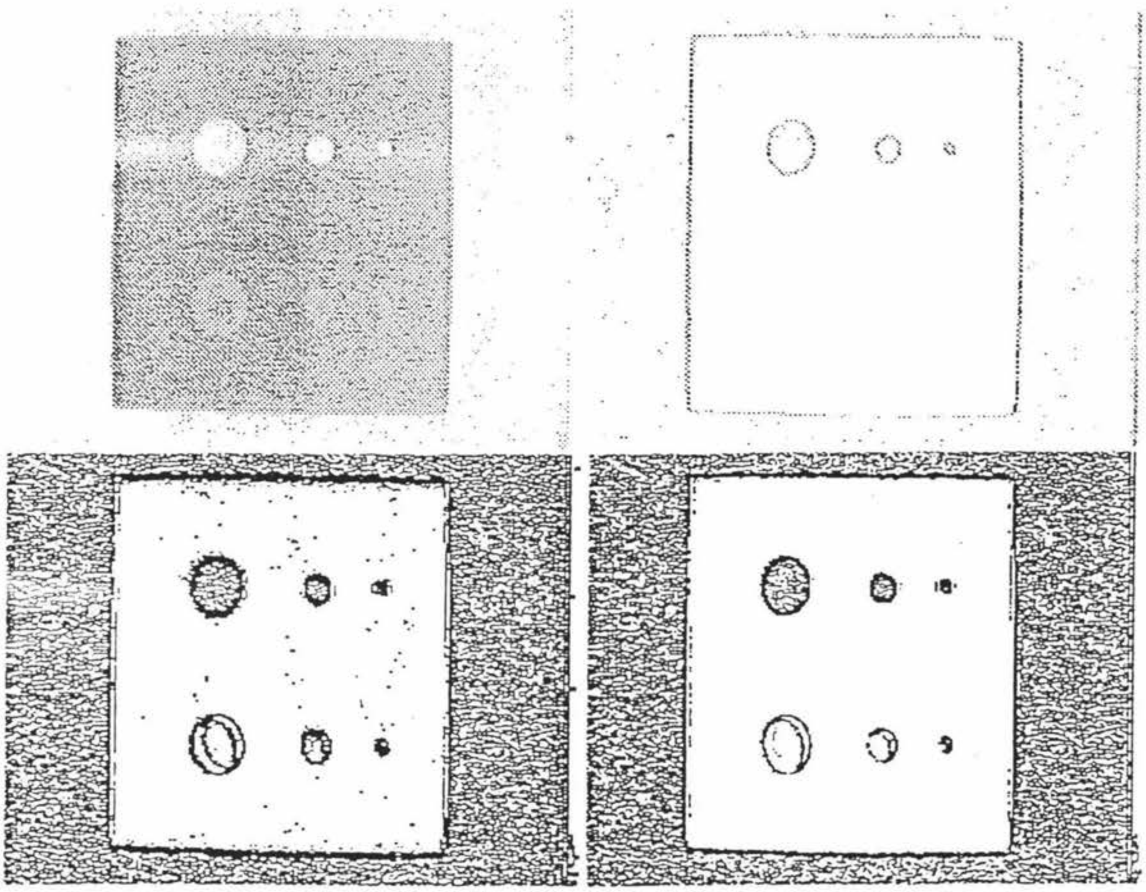


Figure 6-8(a-d) DIP operator edge detection
(a): Input image; (b): DIP filtered image;
(c): Threshold at 1; (d): Threshold at 2;

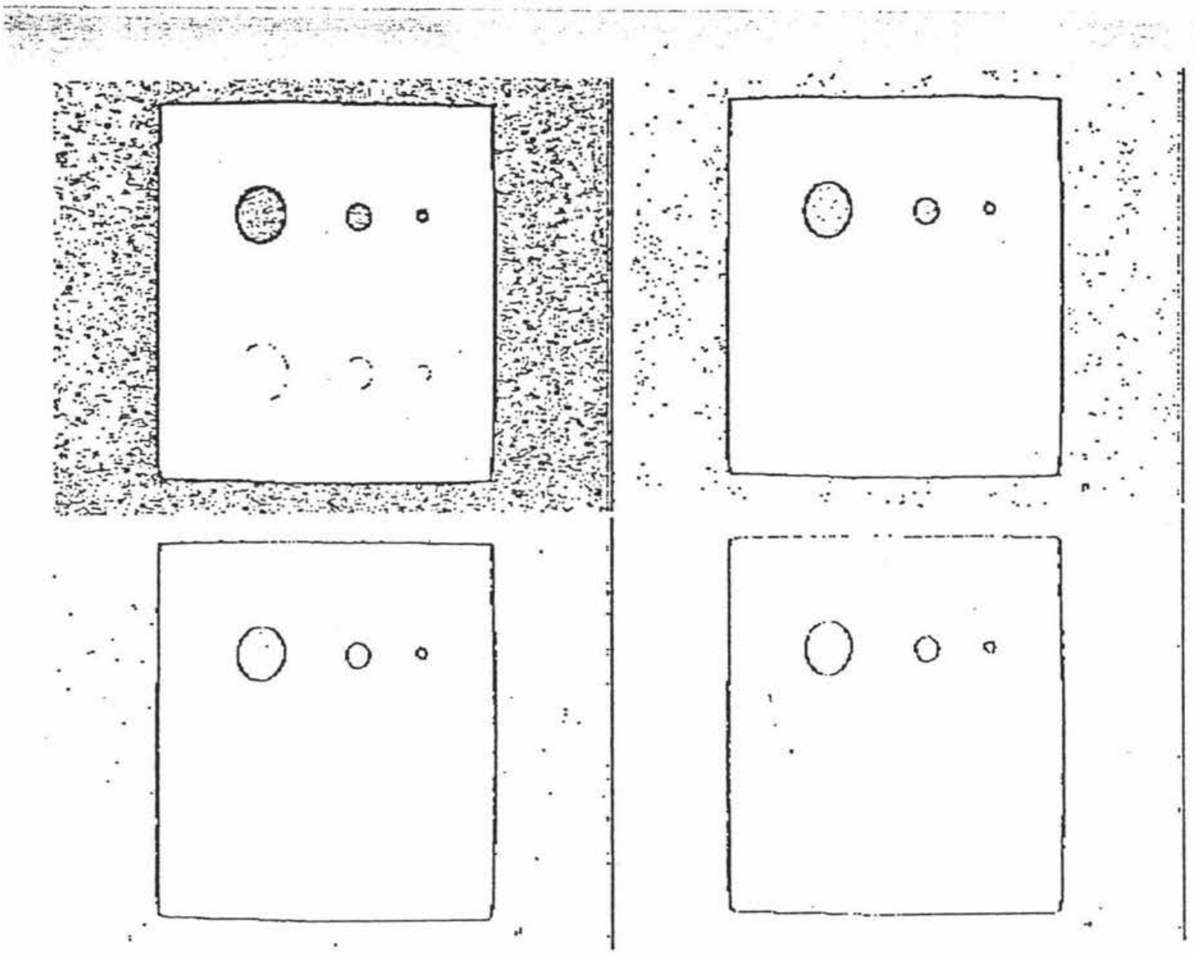


Figure 6-8(e-h) DIP operator thresholding for edge detection
(c): Th=5; (d): Th=10; (e): Th=15; (f): Th=20;

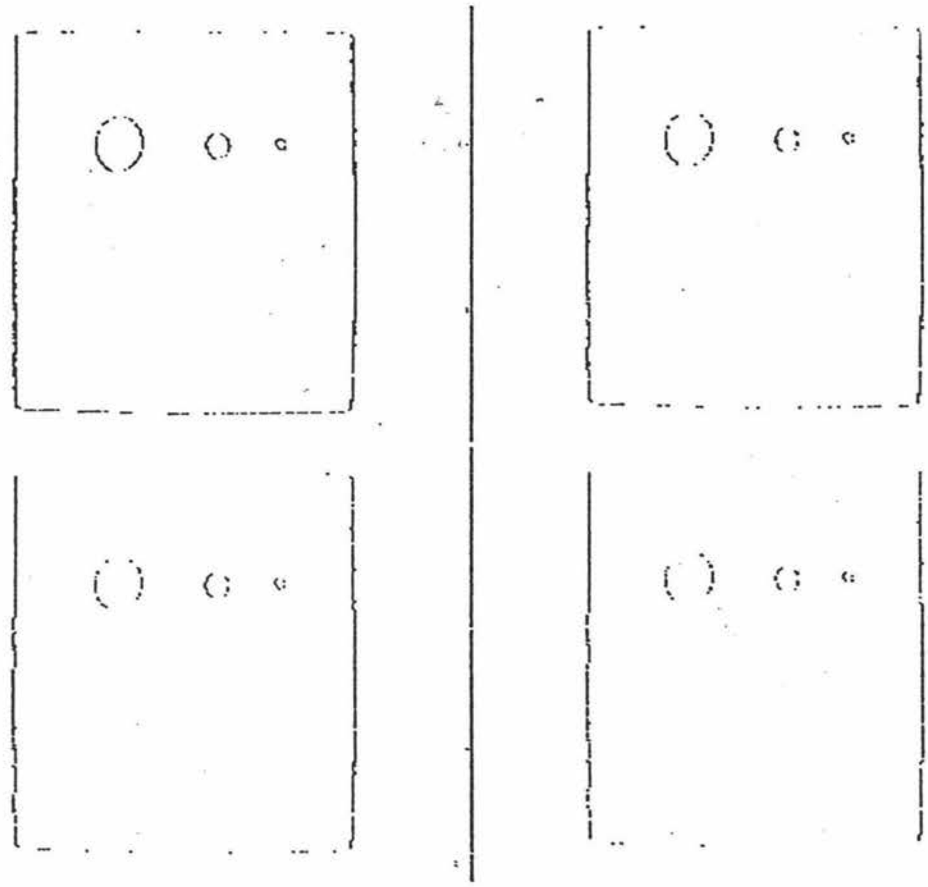


Figure 6-8(i-j) DIP operator thresholding for edge detection
(g): Th=25; (h): Th=30; (i): Th=35; (j): Th=40;

6.3 RANK & RANGE filter performance

In our experiment, both RANK and RANGE filters utilise 3 by 3 square windows. With rank filters, all the pixel values within the window are ranked according to value, regardless of physical location within the window. RANGÉ filters will detect edges by subtracting two different ranked intensity values in the window,

Figure 6-10 shows the results of RANK filtering with rank values of 3, 6, 9. More RANGE filtering results are presented in Figure 6-11 (a-x). From the results, it is clear that RANGE filters perform well with range values of RANGE 5,3 and RANGE 6,3. As shown in Figure 6-11(s,w), both full and blind holes were registered as edges, but shadow edges registered inside blind holes illustrate illumination improvement needs.

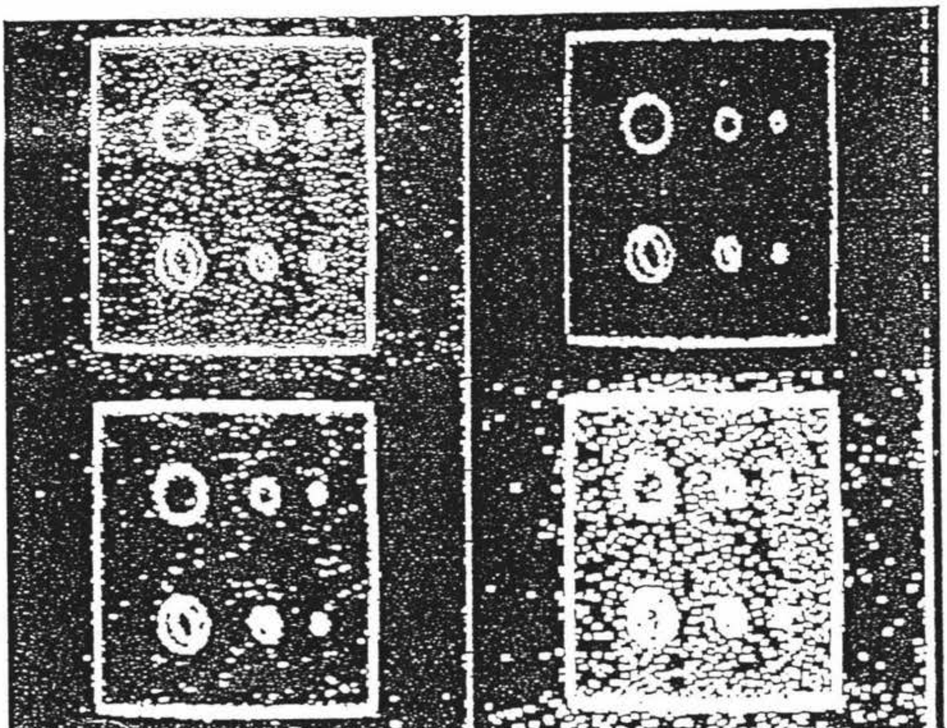


Figure 6-10(a-d) RANK filter performance
(a): RP image Th=5; (b): RANK 3; (c): RANK 6; (d): RANK 9.
Please note that registered pixel are white in this set of results

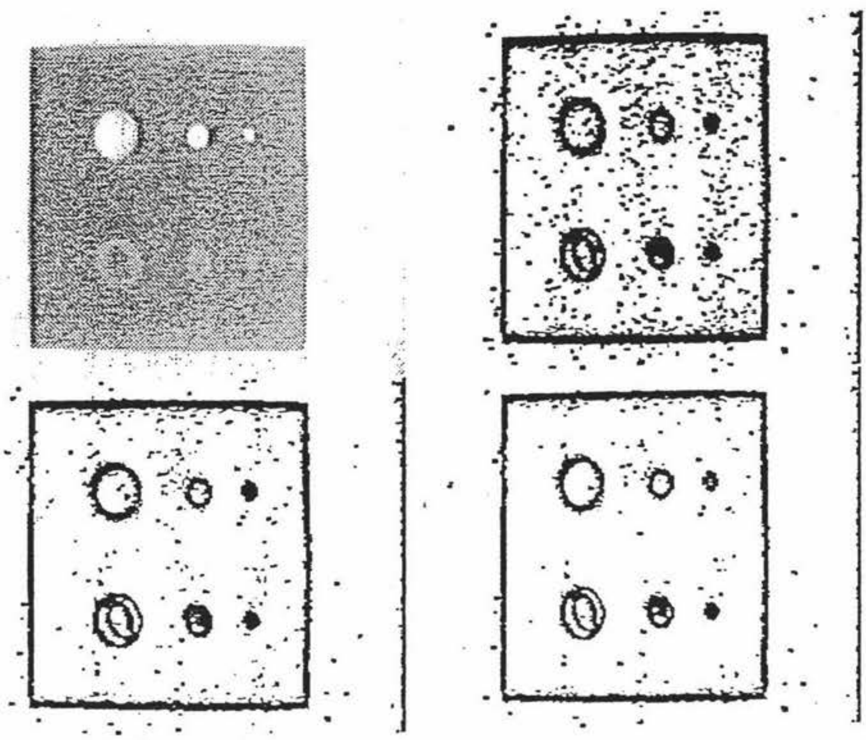


Figure 6-11(a-d) RANGE filter for edge detection
(a): Input image; (b): RANGE 9,1; (c): RANGE 9,2;
(d): RANGE 9,3. Note: all pictures thresholed at 5.

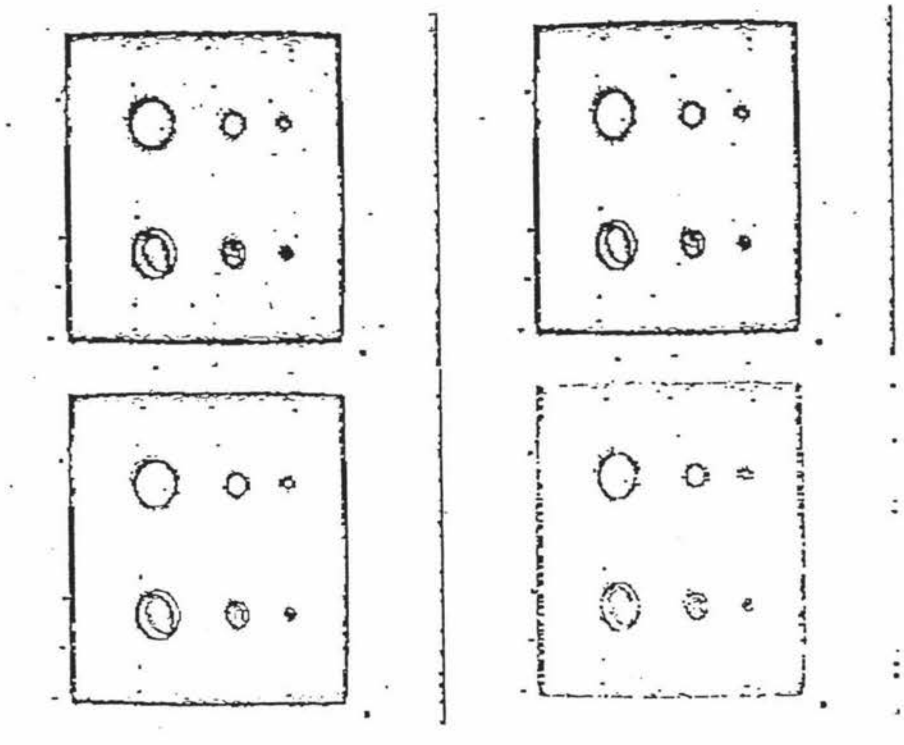


Figure 6-11(e-h) RANGE filter for edge detection
(e): RANGE 9,4; (f): RANGE 9,5;
(g): RANGE 9,6; (h): RANGE 9,7.

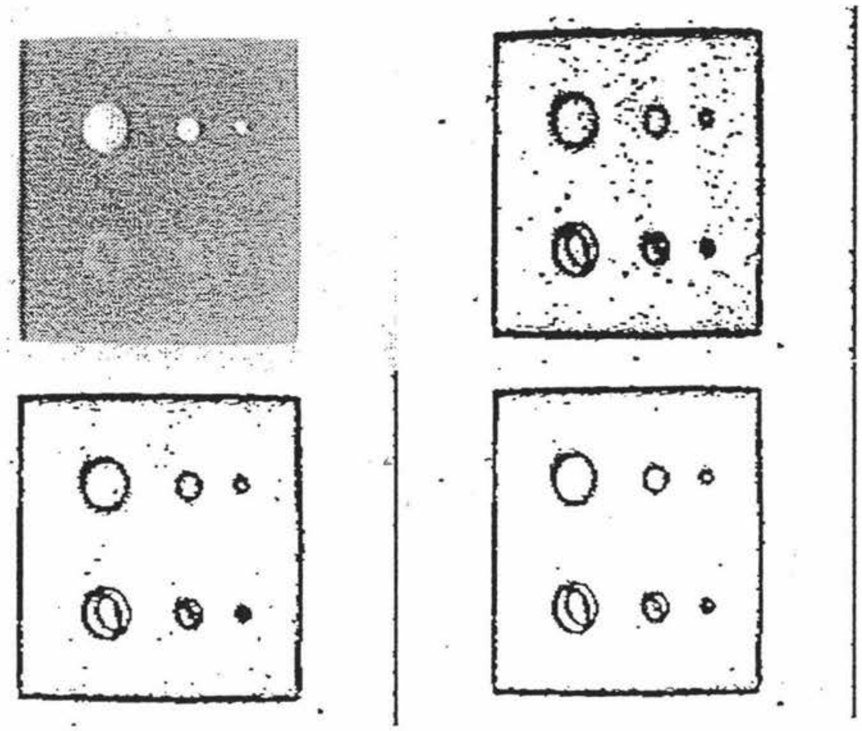


Figure 6-11(i-l) RANGE filter for edge detection
i): Input (j): RANGE 8,1;
(k): RANGE 8,2; (l): RANGE 8,3.

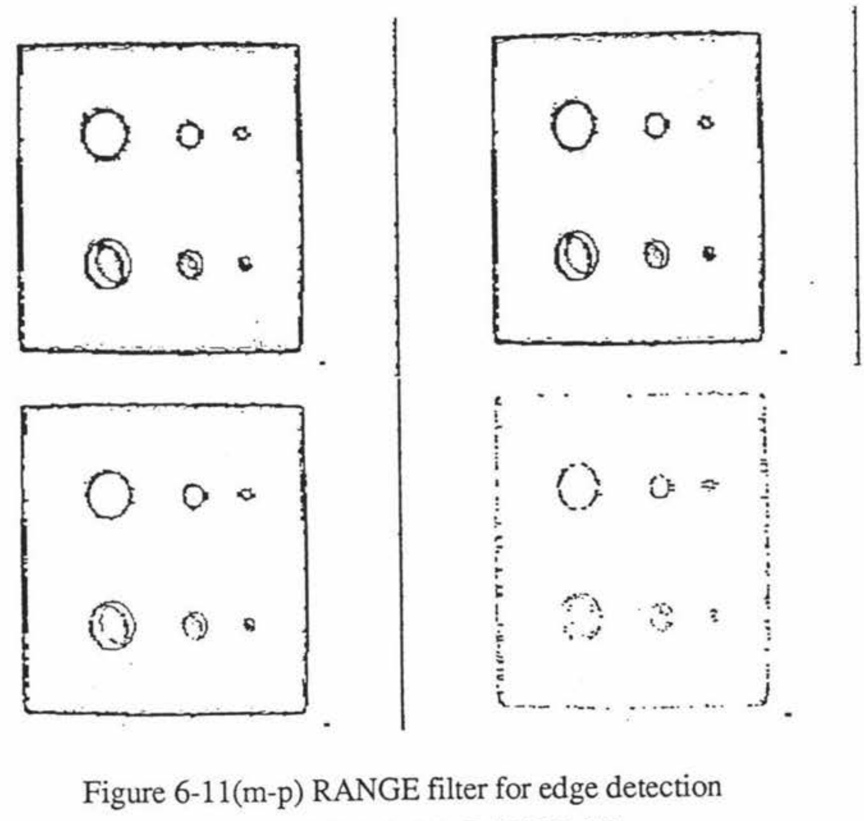


Figure 6-11(m-p) RANGE filter for edge detection
(m): RANGE 8,4; (n): RANGE 8,5;
(o): RANGE 5,6; (p): RANGE 8,7.

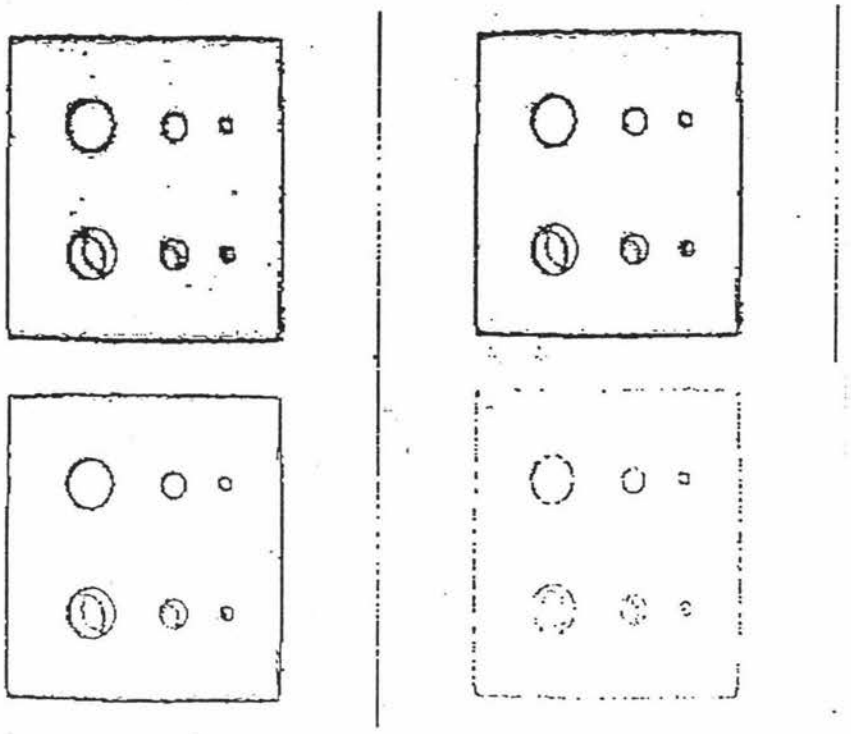


Figure 6-11(q-t) RANGE filter for edge detection
(q): RANGE 6,1; (r): RANGE 6,2;
(s): RANGE 6,3; (t): RANGE 6,4.

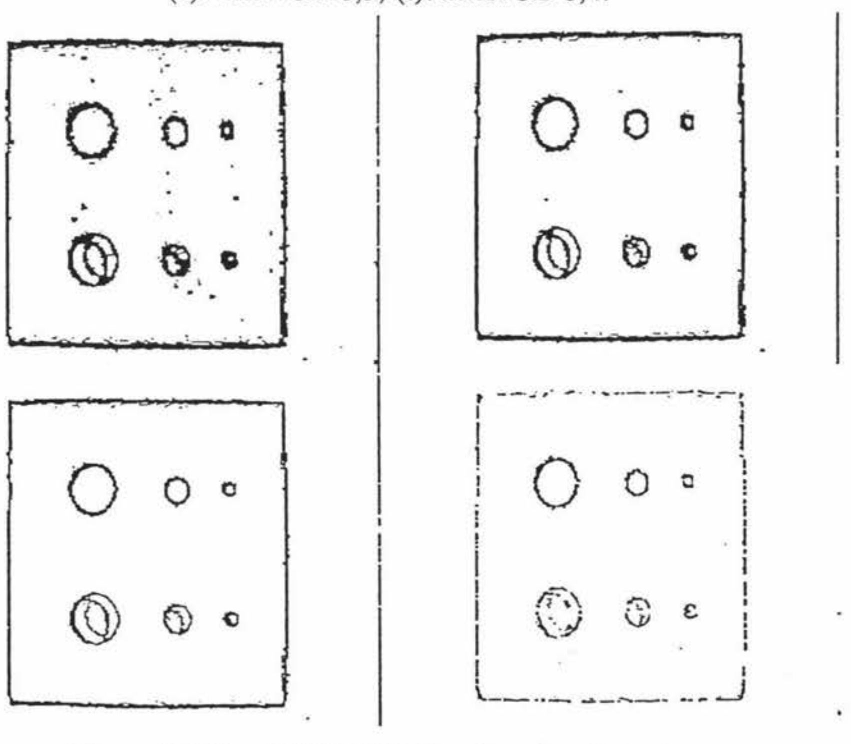


Figure 6-11(u-x) RANGE filter for edge detection
(u): RANGE 5,1; (v): RANGE 5,2;
(w): RANGE 5,3; (x): RANGE 5,4.

6.4 Discussion & conclusions

The Roberts' operator combined with flat thresholding in edge processing has the advantage of simplicity but is noise sensitive. To reduce noise, linear filtering can be used in the early processing stage or after, but will take more processing time and blur the edges.

The standard Sobel operator consists of 3 by 3 mask which performs differentiation in the horizontal and vertical directions. The Sobel operator has the advantage of noise reduction compared with Roberts' operator. One disadvantage of Sobel operator compared with Robert's is that it produces thick edges.

The DIP operator depends upon the local intensity which is similar to the natural visual principle. Therefore, DIP operator has demonstrated that it can extract features efficiently. The algorithm is also very simple. Experimental results have showed that DIP operator performance is much the same as Roberts' operator, but edge extraction is more continuous and thinner.

Experimental results have showed that 3 by 3 square window rank filtering has the ability to sharpen or enhance the edges in the blurred image. The 3 by 3 square window rank filter performs well for noise reduction with rank value of 3. For edge detection, range filters are less sensitive to noise than Roberts' and Sobel operator. However, the optimum range filter window and its rank values for detecting the blind holes' edges without shadow effects require more extensive investigation.

From the results, fixed thresholding is a simple approach, because it only involves using an established number and a simple logic comparison.

Smoothing is an effective way to reduce noise in the image pre-processing, but the cost is not only the computing time but also the blurring effects on edges and other sharp details.

Compared with some complex filtering techniques, thresholding and smoothing are computationally less expensive. With adaptive thresholding to detect desired features, there will be wide application potential in automatic visual inspection.

However, the theory and implementation of optimum edge detection is still an active research topic. Clearly, for real-time image processing fast algorithms such as Roberts' and DIP operators have advantages. Algorithm using a priori knowledge of noise or background brightness could also achieve better processing results.

Among the possible extensions of the work, the most interesting and important task is the integration of different edge operator outputs into a single description. The problem is not easy to solve because there is still a lack of well defined criteria to evaluate edge operators in performance terms.

CHAPTER 7

CHAPTER 7 Conclusions and Future Work

Automatic visual inspection techniques for quality control have been investigated. Study has showed the demand for automatic visual inspection system development for improving product quality productivity.

In automatic visual inspection, the feature extraction technique has to be used in the preprocessing stage. Clearly, model based inspection is the most suitable method for the machined parts. However, the importance of line scanning technology should not be ignored. A wide range of applications have been reported by many research organisations. Focusing on feature extraction techniques, the classical gradient operators, the non-linear filtering operators were examined.

To develop a reliable algorithm, systematic testing of noise distribution and edge strength information has been conducted within a developed high speed image processing hardware system. This noise information is a valuable reference for threshold profile setting in the system which could avoid the noise peak sensitive range and to accommodate the useful edge information.

The experimental results indicate that non-linear adaptive thresholding profile selection based on noise estimates for optimum edge detection is still a research topic. The pixel count technique is useful to characterise statistically the noise effect in the whole image processing system. However, the pixel contour number selection is one which would need more detailed examination.

In researching the principle of the non-linear thresholding, Weber's law was found to support nonlinear W U and V thresholding profiles. This hypothesis has been suggested as a principle for machine vision. Experimental results have proved the value of using the W U and V thresholding profiles. They are reliable and applicable for many situations, and work under unstructured lighting conditions. The research work suggests that inference from the natural law of vision plays a very important role in algorithm development.

VIPS system availability in the Department has provided a rich environment for algorithm comparison. The nonlinear thresholding technique compared with other edge detection operators has demonstrated that the Roberts' operator with adaptive thresholding was a simple, fast, accurate and cost effective method.

Theoretical work in the understanding of multivalley thresholding profiles remains to be done. This is the second graph of Figure 5-3 which corresponds to active vision. Investigation should focus on moving images, where automatic threshold profiles can be determined by a knowledge based system that recognizing the input image, detection requirements and knowledge of the environment characteristics.

REFERENCES

References

[Abr 81] Abramatic J F., "Note Why the Simplest 'Hueckel' Edge Detector is a Roberts Operator" Computer Graphics and Image Processing 17, pp.79-83, 1981.

[Bai 85] Bailey D.G. "Hardware and software development for applied digital image processing", Ph.D Thesis. Department of Electrical and Electronic Engineering, University of Canterbury, Presented 1985.

[Bai & Hod 85] Bailey D.G. and Hodgson R.M., "Range filters: local intensity subrange and their properties", Image and Vision Computing. Vol 3, No 3, pp. 99-110, 1985.

[Bai & Hod 88] Bailey D.G. and Hodgson R.M., "VIPS- A Digital Image Processing Algorithm Development Environment" Image and Vision Computing Vol 6 No. 3 August 1988 pp.176-184.

[Bat et al 85] Batchelor B.G., Hill D.A., and Hodgson D.C.,Eds. "Automatic visual inspection". Elsevier Science Publishing, New York, 1985.

[Bov et al 87] Bovik A C., Huang T S., and Munson D C., The Effect of Median Filtering on Edge Estimation and Detection" IEEE Transaction on Pattern Analysis and Machine Intelligence, VOL. PAMI-9, No. 2, March 1987 pp.181-194.

[Can 86] Canny J., "A Computational Approach to Edge Detection" IEEE Transaction on Pattern Analysis and Machine Intelligence, VOL. PAMI-8, No. 6, November 1986 pp.679-698.

[Chi 82] Chin R T. and Harlow C A., "Automated Visual Inspection: A Survey" IEEE Transaction On Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 6, November pp. 557-573, 1982.

[Chi 88] Chin R T., "Automated Visual Inspection: 1981 to 1987" Computer Vision, Graphics, and Image Processing 41, pp.346-381, 1988.

[Cro 86] Crowley J L., "Machine Vision: Three Generation of Commercial System" MACHINE VISION CAPABILITIES FOR INDUSTRY Edited by Nello Zuech, Machine Vision Association of SME, Dearborn, Michigan 48121. 1986.

[Dav 75] Davis L S., "SURVEY A Survey of Edge Detection Techniques" Computer Graphics and Image Processing Vol. 4 pp.248-270 1975.

[Elp 87] Elphinstone A C. et al, "RAPAC: a high-speed image-processing system" pp. 39-46. IEE Proceedings, Vol. 134, Pt.E, No.1 January 1987.

[Han 86] Hanahara K., Maruyama T., and Uchiyama T., "High-speed Hough transform processor and its applications to automatic inspection and measurement", IEEE Int. Conf. Robotics and Automation, San Francisco, California, April 1986, pp. 1954-1959.

[Has & Siy 88] M. H. Hassan and P. Siy, "Thresholding Based on Learning Theory" Proceeding of the IEEE, VOL. 76, NO. 10, October 1988. pp. 1379-1381.

[Hec 24] Hecht S., "The Visual Discrimination of Intensity and the Weber-Fechner Law," J.Gen. Physiol.,7, 1924, 241.

[Hod et al 85] Hodgson R.M., Bailey D.G., Naylor M.J., Ng A.L.M., and McNeill S.J., "Properties, Implementation and Applications of RANK Filters" Image and Vision Computing, Vol 3 pp. 3-14, 1985.

[Hod et al 87] Hodgson R.M., Naylor M.J., Bailey D.G., and Edward L.N.M., "A Rank and Range Filter for Image Processing Applications", The 2nd IEE image processing and its application conference pp.130-132." 1987.

[Hod 89] Hodgson R.M., "Computer Vision - the Potential for Automatic Inspection and Quality Control". NZOQA Conf.89, Christchurch May 1989.

[Jur 79] Juran J.M., Gruna F.M., and Bingham R.S., "Quality Control Handbook" 3rd Edition, McGraw-Hill, New York 1979. pp.12- to 12- .

[Mar & Hil 79] Marr D., and Hildreth E., "Theory of edge detection" Proc. Royal Society of London Bulletin, vol. 204, pp.301-328, 1979.

[Mcl and Mon 84] Mcllroy C.D. and Monteith W, "Hardware for real-time image processing" pp. 223-229. IEE Proceedings Vol.133 Pt.E. No.6 November 1984.

[Mcl et al 84] Mcllroy C.D. Linggard R. and Monteith W, "Real-time edge detection for image processing", Proc. 7th international conference on Pattern Recognition, Montreal, Canada, pp. 445-454, July 1984.

[Mon & Xin 89] Monteith W, and Xing G.X, "The Real Time performance of a Universal Thresholding Algorithm for Machine Vision with Non Structured Light", Proceeding of The Third International Conference on Image Processing and Applications, IEE Conference Publication 307, pp 498-500, 1989.

[Mon et al 89] Monteith W, Chaplin R I, and Xing G.X, "ASIC Approach to Industrial Inspection Using Non Structured Light", Proceeding of The Third International Conference on Image Processing and Applications, IEE Conference Publication 307, 1989.

[Nag 87] Nagan K.N., Kassim A.A. and Singh H.S., "Parallel image-processing system based on the TMS32010 digital signal processor" pp. 119-124. IEE Proceedings, Vol. 134, Pt.E, No.2 March 1987.

[Pau & McC 86] Paulsen M.R. and McClure W.F., "Illumination for Computer Vision Systems" American Society of Agricultural Engineers Vol. 29(5) September-October, pp.1398-1404, 1986.

[Pra 78] Pratt W.K., "Digital Image Processing" Wiley, New York,1978.

[Rob 65] Roberts L G., "Machine Perception of Three Dimensional Solid in Optical and Electro Optical Information Processing" MIT Press, Cambridge, Mass, 1965.

[Ros 76] Rosenfeld A and Kak A.C., "Digital Image Processing" Academic Press, New York, USA, 1976.

[Ros 84] Rosenfeld A, "Digital Image Processing" Second Edition Academic Press, New York, USA, 1984.

[Ryo & Kim 88] Ryoo & Kim, "Valley Operator for Extracting Sketch Features: DIP", Electronics 14th April 1988 Vol. 24 No.8.

[Wal 88] Wallace A.M., "Industrial applications of computer vision since 1982" IEE Proceedings Vol. 135 Pt.E. No.3 May, pp.117-136, 1988.

[Wes 78] Weszka J.S., "A survey of thresholding selection techniques", Computer Graphics and Image Processing 7. 1978 pp. 259-265.

[Xin & Mon 88] Xing G.X., and Monteith W, "A High Speed Algorithm for Computed Tomographic Analysis of Bone Structures" NELCON 88, The 25th National Electronics Conference, Christchurch, New Zealand, September 1988.

[Xin 88] Xing G.X. "Optimal Edge Detection for Bone Structure Measurement from Computed Tomography Scans" Dip.Tech. Dissertation presented in 1988.

Appendix A

Software for feature extraction algorithm development

MACRO-11 ASSEMABLY SOURCE PROGRAM LIST

```
        .TITLE PIXEL MAC DYNAMIC RP/FLAB WITH EDGE PICTURE PIXEL
COUNT
        .SBTTL PIXEL MAC DYNAMIC RP/FLAB WITH EDGE PICTURE PIXEL
COUNT
;
;       title:    GXPXCT.MAC
;       description:    This program was written to test the noise inherent
;                       in the video camera and Edge Processor of the Real
;                       time Image Processing System.
;
;                       This is accomplished by converting the number of
;                       pixels in a single edge picture frame of size 256 by
;                       292 pixels.
;
;                       The program prints the notch width (W), f(LAB) (FLAB),
;                       the local average brightness (LAB), the actual frame
;                       and theedge picture pixel count (PIXEL1 & PIXEL2).
;
;       history:
;       22-Nov-89       G.X. XING created from early pixcon.mac version.
;       10-Jan-90       Added comments.
        .NLIST
        .INCLUDE /SYSDEF.INC/
        .INCLUDE /CONFIG.INC/
        .LIST
        .NLIST BEX
        .MCALL          .EXIT,          .PRINT
                                        ;
                                        ;Define constants
                                        ;

        SKIPLN          = 3                     ;"EXTRA" VIDEO LINES TO IGNORE
        SKIPBY          = SKIPLN * VROWBY       ;NUMBER OF BYTES TO
IGNORE
        VEND            = EDVRAM - SKIPBY       ;VIDEO END FOR THIS PROG

        TEN     = 10.
        HUNDRD          = 100.
        THOU            = 1000.
        TNTHOU          = 10000.
.ASECT
.=BASE2

        .PSECT          CODE
                                        ;
                                        ;Initialize program. Set priority
                                        ;and mode, print a nice message.
                                        ;Then map to the video RAM and fill
                                        ;it with a know value.
                                        ;
```

```
PIXEL::
        MOV  #PRI7,R0
        BIS  #PSMODE,R0
        MOV  R0,@#PSW              ;SET PRRIORITY 7, USER MODE
        .PRINT      #STRMSG
        CALL INMVID                        ;INITIALIZE VIDEO FRAME
MAPPING
        CALL MAPVID                        ;MAP TO IT
        MOVB        #'Z,R0                         ;FILL VIDEO WITH
SOMETHING KNOWN
        CALL FILVID

                                   ;
                                   ;INITIALIZE I/O PORTS.
                                   ;SET NOTCH WIDTH AND # OF TEST
COUNTER
                                   ;

INIT:   CLR  PTXCSR                        ;DISABLE PRINTER XMIT
INTERRUPT
        CLR  PTRCSR                        ;DISABLE PRINTER RC INTERRUPT
        MOV  #400,PRFCSR                   ;SETUP PROFILE PORT FOR OUTPUT
        MOV  #400,CTLCSR                   ;SETUP CONTROL PORT FOR
OUTPUT
        MOV  #EPOFF,CTLDBR                 ;DISABLE EDGE PROCESSOR
        MOV  #1,@#W                        ;SET W NOTCH WIDTH
        CLR  C                     ;CLEAR COUNTER

                                   ;
                                   ;INITIALISE PARAMETER
                                   ;

LOOP0:       MOV  #0,R0                    ;CLEAR RP COUNTER
LOOP1:       MOV  #0,R2
LOOP2:       MOV  #1,DELAY
        MOV  R2,@#LAB
        MOV  #0,R4
        MOV  @#W,R1
                                   ;
                                   ;GENERATE THE PROFILE TO BE SENT TO
                                   ;EDGE PROCESSOR. THE PROFILE IS
                                   ;CREATED ONE BYTE AT A TIME IN R3
                                   ;
LOOP3:       CMP  R2,R4
        BGE  LOOP4
        CMP  W,R1
        BLT  LOOP4
        INC  R1
        MOV  R0,R3
        JMP  LOOP6
;
```

```
LOOP4:      MOV  #377,R3
                                     ;SEND PROFILE BYTE TO EDGE
PROCESSOR
                                     ;CARD, IF DONE THIS PROFILE THEN
                                     ;OUTPUT FRAME AND DATA
LOOP6:      CALL  SNDPRF                       ;SEND PROFILE
      INC   R4
      CMP   #377,R4
      BNE   LOOP3
      INC   R2
;
      SAVEREGS                       ;SAVE REGISTERS
      CLR   PTXCSR
      CLR   PTRCSR
      MOV   #LF,PTXBUF                    ;SEND EXTRA LF TO PRINTER
      CALL  WAIT
                                     ;PRINT NOTCH WIDTH
      CLR   B1
      CLR   B2
      MOVB        W,B1
      CALL  CONV
      CALL  PRASC
                                     ;PRINT FLAB
      MOVB        FLAB,B1
      CALL  CONV
      CALL  PRASC
                                     ;PRINT LAB
      MOVB        LAB,B1
      CALL  CONV
      CALL  PRASC
      SAVEREGS                       ;SAVE ALL REGISTERS
      .PRINT      #STRFRM                      ;PRINT NICE MESSAGE
      CALL  FRMGRB                        ;FRAME GRAB
      CALL  MAPVID                        ;MAP VIDEO RAM
;     CALL  EPSNPV                        ;EPSON PRINT VIDEO FRAME
      RESTORE                             ;RESTORE REGISTERS
      CALL  PIXCT                         ;COUNT THE PIXELS
                                     ;PRINT PIXEL COUNT
      MOV   PIXEL1,B1
      MOV   PIXEL2,B2
      CALL  CONV
      CALL  PRASC
;     MOV   #LF,PTXBUF                    ;ADVANCE PAPER
      CALL  WAIT
      MOV   #CR,PTXBUF                    ;CARRIAGE RETURN
      CALL  WAIT
      RESTORE                             ;RESTORE ALL REGISTERS
      CMP   #226,R2                       ;#226,R2 STOP AT LAB=149,
#400=256.
      BEQ   LOOP8
```

```
LOOP7:          DEC    DELAY                    ;DELAY FOR MONITOR TO
                                                DISPLAY FRAME
        CMP     #0,DELAY
        BNE     LOOP7
        JMP     LOOP2
LOOP8:          INC    R0                       ;INCREAMENT RP COUNTER
        CMP     #15,R0                          ;STOP AT (#15 RP=12)
;       CMP     #12,R0                          ;STOP AT (#12 RP=9)
        BEQ     10$                     ;IF NOT DONE THEN
        JMP     LOOP1
10$:    INC     C                       ;DO NEXT PROFILE
        CMP     #1,C                    ;STOP AT THE 3RD TEST
        BEQ     1$                      ;IF NOT DONE THEN REPEAT TEST
        JMP     LOOP0
1$:     CALL    UMPVID                          ;UNMAP VIDEO RAM/DUAL PORT
MEMORY
        .PRINT          #ENDMSG
        .EXIT                           ;END: PIXEL
;
;       SUBROUTINE:         CONV
;       DESCRIPTION:        CONVERT 2 16 BIT WORDS TO DECIMAL ASCII
;       INPUT:              B1 - LOW WORD; B2 - HIGH WORD
;       OUTPUT:             ASCII  - TEN THOUSANDS DIGIT
;                       ASCII1 - THOUSANDS DIGIT
;                       ASCII2 - HUNDREDS DIGIT
;                       ASCII3 - TENS DIGIT
;                       ASCII4 - ONES DIGIT
;       HISTORY:
;       22-NOV-89   G.X. XING MODIFIED FORM PIXCON.MAC
;
CONV:           MOVB            #ASCII0,ASCII
        MOVB            #ASCII0,ASCII1
        MOVB            #ASCII0,ASCII2
        MOVB            #ASCII0,ASCII3
        MOVB            #ASCII0,ASCII4
1$:     SUB     #TNTHOU,B1
        SBC     B2
        BCC     6$
        ADD     #TNTHOU,B1
        ADC     B2
2$:     CMP     B1,#THOU
        BGE     7$
        CMP     B2,#0
        BGT     7$
3$:     CMP     B1,#HUNDRD
        BGE     9$
        CMP     B2,#0
        BGT     9$
```

```
4$:     CMP   B1,#TEN
        BGE   11$
        CMP   B2,#0
        BGT   11$
5$:     CMP   B1,#0
        BGT   13$
        CMP   B2,#0
        BGT   13$
        BR    99$
6$:     INCB  ASCII
        JMP   1$
7$:     SUB   #THOU,B1
        BCC   8$
        DEC   B2
8$:     INCB  ASCII1
        JMP   2$
9$:     SUB   #HUNDRD,B1
        BCC   10$
        DEC   B2
10$:    INCB  ASCII2
        JMP   3$
11$:    SUB   #TEN,B1
        BCC   12$
        DEC   B2
12$:    INCB  ASCII3
        JMP   4$
13$:    DEC   B1
        BCC   14$
        DEC   B2
14$:    INCB  ASCII4
        JMP   5$
99$:    RETURN
;
;       SUBROUTINE:       WAIT
;       DESCRIPION:       WAIT FOR CHARACTER TO BE SENT OUT TO THE
PRINTER
;       HISTORY:
;       22-NOV-89   G.X. XING MODIFIED FROM PIXCON.MAC
;
WAIT:           CMP   #XREADY,@#PTXCSR      ;WAIT FOR XMIT READY
        BNE   WAIT
W1:     CMP   #RDONE,@#PTRCSR               ;SEE IF CHAR HAS BEEN
SENT
        BNE   W3                ;RTS IF NO
        CMP   #CTRLS,@#PTRBUF               ;SEE IF CTRL-S
        BNE   W3                ;RTS IF NO
W2:     CMP   #RDONE,@#PTRCSR               ;SEE IF ANOTHER CHAR
SENT
        BNE   W2                ;LOOP IF NO
W3:     RETURN                          ;RETURN WHEN NEXT CHAR SENT
;
```

```
;       SUBROUTINE:      PRASC
;       DESCRIPTION:     PRINT THE FIVE DECIMAL DIGITS FOLLOWED BY
A
;                   BLACK CHARACTER.
;       HISTORY:
;       22-NOV-89   G.X. XING MODIFIED FROM PIXCON.MAC
;
PRASC:       MOVB         ASCII,@#PTXBUF
        CALL  WAIT
        MOVB        ASCII1,@#PTXBUF
        CALL  WAIT
        MOVB        ASCII2,@#PTXBUF
        CALL  WAIT
        MOVB        ASCII3,@#PTXBUF
        CALL  WAIT
        MOVB        ASCII4,@#PTXBUF
        CALL  WAIT
        MOVB        #40,@#PTXBUF
        CALL  WAIT
        RETURN
;
;       SUBROUNTINE:     PIXCT
;       DESCRIPTION:     COUNT THE PIXELS IN A COMPLETED FRAME
EDGE PICTURE
;       HISTORY:
;       22-NOV-89   G.X. XING MODIFIED FROM PIXCON.MAC
;
PIXCT:       PUSH         R0                   ;STACK R0, R1 & R2
        PUSH        R1
        PUSH        R2
        CLR   PIXEL1                  ;CLEAR PIXEL COUNTER
        CLR   PIXEL2                  ;
        MOV   #STVRAM,R0              ;INIT LOCATION POINTER
1$:     CLR   R2                 ;CLEAR BIT COUNT
        MOV   (R0)+,R1           ;GET PIXEL WORD
2$:     ROL   R1                 ;EXTRACT PIXEL
        BCC   3$                 ;PIXEL=0?
        ADD   #1,PIXEL1          ;UPDATE DOUBLE WORD PIXEL COUNT
        ADC   PIXEL2
3$:     INC   R2                 ;INCREMENT BIT COUNT
        CMP   R2,#20                  ;WORD COMPLETE?
        BLT   2$                 ;NO
        CMP   R0,#VEND           ;FINISHED PICTURE?
        BLE   1$                 ;NO
        POP   R2                 ;RESTORE REGISTERS R0, R1 & R2
        POP   R1
        POP   R0
        RETURN
```

```
        .PSECT DATA
;
LBAD:           .BLKW       1           ;LINE BASE ADDRS
CBAD:           .BLKW       1           ;CURRENT BASE ADDRS
CWAD:           .BLKW       1           ;CURRENT WORD ADDRS
DWCNT:          .BLKW       1           ;DATE WORD COUNT
OWCNT:          .BLKW       1           ;O/P WORD COUNT
LCNT:           .BLKW       1           ;LINE COUNT
BCNT:           .BLKW       1           ;BIT COUNT
OPBUF:          .BLKW       1           ;O/P BUFFER
DELAY:          .BLKW       1           ;DELAY
B1:     .BLKW       1           ;BINARY NUMBER BUFFER
B2:     .BLKW       1           ;
C:      .BLKW       1           ;REPEAT COUNT NUMBER
W:      .BLKW       1           ;W - NOTCH WIDTH
FLAB:           .BLKW       1           ;CURRENT FLAB VALUE
LAB:    .BLKW       1           ;CURRENT LAB VALUE
ASCII:          .BLKW       1           ;ASCII
ASCII1:         .BLKW       1           ;ASCII1
ASCII2:         .BLKW       1           ;ASCII2
ASCII3:         .BLKW       1           ;ASCII3
ASCII4:         .BLKW       1           ;ASCII4
PIXEL1:         .BLKW       1           ;PIXELS COUNTER
PIXEL2:         .BLKW       1
STRMSG:         .ASCIZ      /START PIXEL COUNT PROGRAM/
ENDMSG:         .ASCIZ      /END PIXEL COUNT PROGRAM/
STRFRM:         .ASCIZ      /CALL FRAME GRAB/
        .END    PIXEL
```

MACRO-11 ASSEMABLY PROGRAM FOR WRITE A W SHAPE THRESHOLDING PROFILE

```
        .TITLE GXW0 THRESHOLD FUNCTION WRITE
;
;       THRESHOLDING FUNCTION
;
        .MCALL  .EXIT,.PRINT
        .INCLUDE /SYSDEF.INC/
        .INCLUDE /STRUCT.INC/

        .PSECT  CODE
GXW0::
INIT:   MOV     #400,CSRC
        MOV     #400,CSRD       ;SET UP CS REGS FOR O/P
;
        CLR     R4              ;CLEAR INDEX REG
        MOV     #177777,DBRD
LOOP:   MOV     TF(R4),DBRC     ;SEND DATA/ADD WORD
        MOV     #177774,DBRD
        MOV     #177777,DBRD    ;PULSE LUW AND WE LOW
        INC     R4
        INC     R4              ;NEXT WORD
        CMP     #1000,R4        ;256 WORDS SENT?
        BNE     LOOP            ;LOOP IF NOT FINISHED
        .EXIT
        HALT
;
;           F(LAB) DATA/ADDRESS MATRIX
;
TF:     .WORD   177400,177401,177402,177403,005004,005005,006006,007007

        .WORD   001010,001111,001212,001313,001414,001415,001316,001217

        .WORD   001120,001021,000722,000623,000524,000525,177426,177427

        .WORD   177430,177431,177432,177433,177434,177435,177436,177437

        .WORD   177440,177441,177442,177443,177444,177445,177446,177447

        .WORD   177450,177451,177452,177453,177454,177455,177456,177457

        .WORD   177460,177461,177462,177463,177464,177465,177466,177467

        .WORD   177470,177471,177472,177473,177474,177475,177476,177477

        .WORD   177500,177501,177502,177503,177504,177505,177506,177507
```

```
.WORD   177510,177511,177512,177513,177514,177515,177516,177517
.WORD   177520,177521,177522,177523,177524,177525,177526,177527
.WORD   177530,177531,177532,177533,177534,177535,177536,177537
.WORD   177540,177541,177542,177543,177544,177545,177546,177547
.WORD   177550,177551,177552,177553,177554,177555,177556,177557
.WORD   177560,177561,177562,177563,177564,177565,177566,177567
.WORD   177570,177571,177572,177573,177574,177575,177576,177577
.WORD   177600,177601,177602,177603,177604,177605,177606,177607
.WORD   177610,177611,177612,177613,177614,177615,177616,177617
.WORD   177620,177621,177622,177623,177624,177625,177626,177627
.WORD   177630,177631,177632,177633,177634,177635,177636,177637
.WORD   177640,177641,177642,177643,177644,177645,177646,177647
.WORD   177650,177651,177652,177653,177654,177655,177656,177657
.WORD   177660,177661,177662,177663,177664,177665,177666,177667
.WORD   177670,177671,177672,177673,177674,177675,177676,177677
.WORD   177700,177701,177702,177703,177704,177705,177706,177707
.WORD   177710,177711,177712,177713,177714,177715,177716,177717
.WORD   177720,177721,177722,177723,177724,177725,177726,177727
.WORD   177730,177731,177732,177733,177734,177735,177736,177737
.WORD   177740,177741,177742,177743,177744,177745,177746,177747
.WORD   177750,177751,177752,177753,177754,177755,177756,177757
.WORD   177760,177761,177762,177763,177764,177765,177766,177767
.WORD   177770,177771,177772,177773,177774,177775,177776,177777
;
.END GXW0
```

MACRO-11 ASSEMABLY PROGRAM FOR WRITE A U SHAPE THRESHOLDING PROFILE

```
        .TITLE GXU0 THRESHOLD FUNCTION WRITE
;
;       THRESHOLDING FUNCTION
;
        .MCALL  .EXIT,.PRINT
        .INCLUDE /SYSDEF.INC/
        .INCLUDE /STRUCT.INC/

        .PSECT  CODE
GXU0::
INIT:   MOV     #400,CSRC
        MOV     #400,CSRD       ;SET UP CS REGS FOR O/P
;
        CLR     R4              ;CLEAR INDEX REG
        MOV     #177777,DBRD
LOOP:   MOV     TF(R4),DBRC     ;SEND DATA/ADD WORD
        MOV     #177774,DBRD
        MOV     #177777,DBRD    ;PULSE LUW AND WE LOW
        INC     R4
        INC     R4              ;NEXT WORD
        CMP     #1000,R4        ;256 WORDS SENT?
        BNE     LOOP            ;LOOP IF NOT FINISHED
        .EXIT
        HALT
;
;           F(LAB) DATA/ADDRESS MATRIX
;
TF:     .WORD   177400,177401,177402,177403,177404,177405,177406,177407

        .WORD   177410,177411,177412,177413,177414,177415,177416,177417

        .WORD   177420,177421,177422,177423,177424,177425,177426,177427

        .WORD   000530,000531,000532,000533,000534,000535,000536,000537

        .WORD   000640,000641,000642,000643,000644,000645,000646,000647

        .WORD   000750,000751,000752,000753,000754,000755,000756,007457

        .WORD   001060,001061,001062,001063,001064,001065,001066,001067

        .WORD   001170,001171,001172,001173,001174,001175,001176,001177

        .WORD   001300,001301,001302,001303,001304,001305,001306,001307
```

```
.WORD   001410,001411,001412,001413,001414,001415,001416,001417
.WORD   001520,001521,001522,001523,001524,001525,001526,001527
.WORD   001630,001631,001632,001633,001634,001635,001636,001637
.WORD   001740,001741,001742,001743,001744,001745,001746,001747
.WORD   002050,002051,002052,002053,002054,002055,002056,002057
.WORD   002160,002161,002162,002163,002164,002165,002166,002167
.WORD   002270,002271,002272,002273,002274,002275,002276,002277
.WORD   002400,002401,002402,002403,002404,002405,002406,002407
.WORD   002510,002511,002512,002513,002514,002515,002516,002517
.WORD   002620,002621,002622,002623,002624,002625,002626,002627
.WORD   002630,002631,002632,002633,002634,002635,002636,002637
.WORD   003640,003641,003642,003643,003644,003645,003646,003647
.WORD   004650,004651,004652,004653,004654,004655,004656,004657
.WORD   004660,004661,004662,004663,004664,004665,004666,004667
.WORD   177670,177671,177672,177673,177674,177675,177676,177677
.WORD   177700,177701,177702,177703,177704,177705,177706,177707
.WORD   177710,177711,177712,177713,177714,177715,177716,177717
.WORD   177720,177721,177722,177723,177724,177725,177726,177727
.WORD   177730,177731,177732,177733,177734,177735,177736,177737
.WORD   177740,177741,177742,177743,177744,177745,177746,177747
.WORD   177750,177751,177752,177753,177754,177755,177756,177757
.WORD   177760,177761,177762,177763,177764,177765,177766,177767
.WORD   177770,177771,177772,177773,177774,177775,177776,177777
;
.END GXU0
```

MACRO-11 ASSEMABLY PROGRAM FOR WRITE A V SHAPE THRESHOLDING PROFILE

```
    .TITLE GXV0 THRESHOLD FUNCTION WRITE
;
;   THRESHOLDING FUNCTION
;
    .MCALL  .EXIT,.PRINT
    .INCLUDE /SYSDEF.INC/
    .INCLUDE /STRUCT.INC/

    .PSECT  CODE
GXV0::
INIT: MOV   #400,CSRC
    MOV   #400,CSRD      ;SET UP CS REGS FOR O/P
;
    CLR   R4             ;CLEAR INDEX REG
    MOV   #177777,DBRD
LOOP: MOV   TF(R4),DBRC   ;SEND DATA/ADD WORD
    MOV   #177774,DBRD
    MOV   #177777,DBRD   ;PULSE LUW AND WE LOW
    INC   R4
    INC   R4             ;NEXT WORD
    CMP   #1000,R4       ;256 WORDS SENT?
    BNE   LOOP           ;LOOP IF NOT FINISHED
    .EXIT
    HALT
;
;       F(LAB) DATA/ADDRESS MATRIX
;
TF:   .WORD  177400,177401,177402,177403,005004,005005,006006,007007

    .WORD  001010,001111,001212,001313,001414,001415,001316,001217

    .WORD  001120,001021,000722,000623,000524,000525,177426,177427

    .WORD  177430,177431,177432,177433,177434,177435,177436,177437

    .WORD  177440,177441,177442,177443,177444,177445,177446,177447

    .WORD  177450,177451,177452,177453,177454,177455,177456,177457

    .WORD  177460,177461,177462,177463,177464,177465,177466,177467

    .WORD  177470,177471,177472,177473,177474,177475,177476,177477

    .WORD  177500,177501,177502,177503,177504,177505,177506,177507
```

```
        .WORD   177510,177511,177512,177513,177514,177515,177516,177517
        .WORD   177520,177521,177522,177523,177524,177525,177526,177527
        .WORD   177530,177531,177532,177533,177534,177535,177536,177537
        .WORD   177540,177541,177542,177543,177544,177545,177546,177547
        .WORD   177550,177551,177552,177553,177554,177555,177556,177557
        .WORD   177560,177561,177562,177563,177564,177565,177566,177567
        .WORD   177570,177571,177572,177573,177574,177575,177576,177577
        .WORD   177600,177601,177602,177603,177604,177605,177606,177607
        .WORD   177610,177611,177612,177613,177614,177615,177616,177617
        .WORD   177620,177621,177622,177623,177624,177625,177626,177627
        .WORD   177630,177631,177632,177633,177634,177635,177636,177637
        .WORD   177640,177641,177642,177643,177644,177645,177646,177647
        .WORD   177650,177651,177652,177653,177654,177655,177656,177657
        .WORD   177660,177661,177662,177663,177664,177665,177666,177667
        .WORD   003270,003471,003672,004073,004274,004475,004676,005077
        .WORD   005300,005501,005702,006103,006304,006505,006706,007107
        .WORD   177710,177711,177712,177713,177714,177715,177716,177717
        .WORD   177720,177721,177722,177723,177724,177725,177726,177727
        .WORD   177730,177731,177732,177733,177734,177735,177736,177737
        .WORD   177740,177741,177742,177743,177744,177745,177746,177747
        .WORD   177750,177751,177752,177753,177754,177755,177756,177757
        .WORD   177760,177761,177762,177763,177764,177765,177766,177767
        .WORD   177770,177771,177772,177773,177774,177775,177776,177777
;
        .END GXV0
```

Appendix B

Algorithm development in VIPS VAX Image Processing System

Roberts' operator PASCAL source program

```
[inherit ('VIPINH', 'SYSINH')] program Robert (input, output);
{ G.X. Xing, 15 January 1990

COMMAND:        Robert

VIPS Format:        Robert operator image_in img_out img_edge [output]

img_in      (IMAGE) <INPUT> ONE GREY LEVEL IMAGE TO BE ROCESSED.
IMG_OUT   (IMAGE) <OUTPUT> THE PROCESSED IMAGE USING ROBERT
OPERATOR. }

PROCEDURE ROBERT_PROCESS (

                        VAR Image_in : ARRAY [lra..ura : integer; lca..uca :
integer] OF v_pixel;
                        VAR Image_out : ARRAY [lrb..urb : integer; lcb..ucb :
integer] OF v_pixel;
                  VAR RPImage : ARRAY [lrc..urc : integer; lcc..ucc : integer] OF
v_pixel);

 VAR
  Profile : ARRAY [0..255] OF integer;
  D     : ARRAY [1..9] OF integer;
  Lower_LAB, Upper_LAB, Lower_FLAB, Upper_FLAB, RP, LAB, WinSize :
integer;
  x, y , i, FB, MX, MY, MIN , LABType, CCompare : integer;
  Compare : Boolean;
  Slope : real;

 BEGIN
  Min := 0;
  writeln(' Modified Robert Product processing');
   writeln(' ');
   writeln(' LAB Window size, 2 = 2 by 2, 3 = 3 by 3');
  readln( WinSize);
  Writeln('LAB Type, 1 = Average, 2 = Median');
  readln( LABType );
  writeln(' Enter lower and upper LAB limits, 0 - 255 ');
  readln( Lower_LAB, Upper_LAB );
  writeln(' Enter Lower and Upper F(LAB) limits, 0 - 255 ');
  readln( Lower_FLAB, Upper_FLAB);
  writeln(' Enter F(LAB) BASE FB, 0 - 250 ');
  readln( FB );
  Slope := (Upper_FLAB - Lower_FLAB)/(Upper_LAB - Lower_LAB);
  FOR i := 0 TO 255 DO
   BEGIN
   IF (i > Upper_LAB) or (i< Lower_LAB)
    THEN Profile[i] := 255
    ELSE Profile[i] := Round(FB + Slope * (i - Lower_LAB));
  END;
```

```
    FOR y := lca TO (uca-2) DO
     FOR x := lra TO (ura-2) DO
      Begin
       RP := ABS(Image_in[x,y] - Image_in[x+1,y+1]) + ABS(Image_in[x+1,y] -
Image_in[x,y+1]);
       IF WinSize = 2
        THEN LAB := (Image_in[x,y] + Image_in[x+1,y]+ Image_in[x,y+1] +
Image_in[x+1,y+1]) Div 4

        ELSE  LAB := (Image_in[x,y] + Image_in[x+1,y]+ Image_in[x+2,y] +
Image_in[x,y+1] + Image_in[x+1,y+1] + Image_in[x+2,y+1] + Image_in[x,y+2] +
Image_in[x+1,y+2] + Image_in[x+2,y+2]) Div 9 ;
        IF LABType = 1
          THEN LAB := LAB
          ELSE BEGIN
               FOR MX := 1 TO 3 DO
                FOR MY := 1 TO 3 DO
                  BEGIN
                   D[MX*MY] := Image_in[x+MX-1,y+MY-1];
                  END;
               REPEAT
                Compare := False;
                FOR MX := 1 TO 8 DO
                  BEGIN
                   IF D[MX+1] > D[MX]
                     THEN BEGIN
                          MIN:=D[MX]; D[MX]:=D[MX+1]; D[MX+1]:=MIN;
                          Compare := True; CCompare := CCompare + 1;
                         END;
                END;
               IF CCompare > 56
                THEN BEGIN
                     Compare := False ;
                      writeln(' ERROR IN SORT');
                    END;
                UNTIL Not Compare;
              LAB := D[5] ;
              END;
       RPImage[x+1,y+1] := RP;
       IF RP >= Profile[LAB]
       THEN Image_out[x+1,y+1] := 255
       ELSE Image_out[x+1,y+1] := 0;
      END;
     END;

BEGIN
  vip_load_command (
                    { invocation: }     'RP', %ref Robert_Process,
                    { parameter types } vt_b_img, vt_b_img,vt_b_img,,,,,,
                    { parameter source}   vp_io,    vp_io,   vp_io,,,,,,
                    { default values  }        ,          ,          ,,,,,);

 vip_main_loop;
 END.
```

DIP operator PASCAL source program

```
[inherit ('VIPINH', 'SYSINH')] program Dip (input, output);
{ G.X. Xing, 1990
COMMAND:  DIP

VIPS Format:  img_a img_b img_edge

img_a        (IMAGE) <INPUT> One grey level image to be processed.
img_b        (IMAGE) <OUTPUT> The processed image using DIP operator.
img_edge     (IMAGE) <RESULT> The binary image after thresholding.  }

 PROCEDURE DIP_PROCESS   (
   VAR image_a : ARRAY [lra..ura : integer; lca..uca : integer] OF v_pixel;
   VAR image_b : ARRAY [lrb..urb : integer; lcb..ucb : integer] OF v_pixel;
   VAR DIP_e : ARRAY [lre..ure : integer; lce..uce : integer] OF v_pixel);

CONST
   WHITE = 255.0;
   BLACK = 0.0;

 VAR
   MIN, MAX, MIN1, MAX1, ratio : real;
   Th, row, col, x, y, i, j, S, m, n, FSUM : integer;
   TEMP : ARRAY [1..256, 1..256] of real;
```

```
BEGIN
  writeln(' DIP Valley operator processing');
   writeln(' ');
   writeln(' ');
  writeln(' Input M by N Mask, for example 3 by 3');
   writeln(' ');
  READ(M);
   writeln(' ');
  READ(N);
  writeln(' Input scaling "S" values, for example 1');
   writeln(' ');
  READ(S);
  writeln(' Input thresholding "Th" value, for example 20');
  READ(Th);
   FOR row := lra TO (ura - M) DO
    FOR col := lca TO (uca - N) DO
      BEGIN
      FSUM :=0;
      MIN := WHITE;
      MAX := BLACK;
       FOR x := row TO (row + M - 1) DO
        FOR y := col TO (col + N - 1) DO
          BEGIN
           FSUM := FSUM + Image_a[x,y];
           IF Image_a[x,y] >= MAX
             THEN MAX := Image_a[x,y];
             IF image_a[x,y] < MIN
             THEN MIN := Image_a[x,y]
          END;
```

```
        TEMP[row,col] := FSUM * ((1 / (image_a[row+1,col+1] + 1)) - 1 / (MAX + 1))
* S;
        IF TEMP[row,col] >= MAX1
         THEN MAX1 := Image_a[row,col];
        IF image_a[row,col] < MIN1
         THEN MIN1 := Image_a[row,col];
       END;
        ratio := 255 / (ABS(MAX1) + ABS(MIN1) + 1);
        FOR i :=lra To (ura - M) DO
         FOR j := lca TO (uca - N) DO
          BEGIN
           Image_b[i,j] := TRUNC((TEMP[i,j] - MIN1) * ratio);
           IF Image_b[i,j] >= Th
           THEN DIP_e[i,j] := 255
           ELSE DIP_e[i,j] := 0;
          END;
END;
BEGIN
  Vip_load_command ('DIP', %REF Dip_process,
                    vt_b_img,  vt_b_img,       vt_b_img,  vt_re, ,,,,
                    vp_in,     vp_in,      vp_out,        ,         ,,,,
                    ,          ,           ,              ,         ,,,);
  vip_private_help( 'USERDISK:[BLOGGS]PVTHELP' );
Vip_main_loop;
END.
```