1-1-1986

Edge enhancement via phase contrast filtering: A new technique.

Micho Srdanovic
University of Windsor

EDGE ENHANCEMENT VIA PHASE CONTRAST FILTERING: A NEW TECHNIQUE

by

Micho Srdanovic

A Thesis
submitted to the Faculty of Graduate Studies
through the Department of Electrical Engineering
in partial fulfillment of the requirements
for the degree of Master of Applied Science
of the University of Windsor

Windsor, Ontario, Canada

1986

To My Family Members

ABSTRACT

Edge enhancement is encountered in a number of applications of digital image processing. Edges characterize the boundaries of objects and are useful for segmentation, registration and object identification in images. Numerous enhancement techniques are in existence and various classes of these shall be briefly reviewed here.

An alternate technique has been proposed by Soltis [1] which he termed 'phase contrast filtering' (PCF). It is the intent of the thesis to examine this new technique on the basis of its edge enhancement capabilities and to develop a method for the design of two-dimensional (2-D) recursive digital filters to meet the specifications of the PCF method. An examination of the PCF's applicability to enhancement of various images such as medical X-rays, metal surfaces, etc., is also given.

Finally, a comparison between selected edge enhancement techniques and the PCF technique is presented.

ACKNOWLEDGEMENTS

I would sincerely like to thank Dr. M.A. Sid-Ahmed for his advice, guidance and long hours of commitment throughout the course of this research.

I wish to thank Dr. J.J. Soltis for his advice and for the idea behind the technique for which this work is devoted. The comments of other committee members are also gratefully acknowledged.

A deephearted thanks goes to my family members for their love and encouragement. Finally, a heartfelt appreciation goes to my wife Vera for her understanding and patience throughout my absenteeism from home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. A Simple Image Model

Figure 1-1 is a diagram of a scene viewed from some point in space. The word 'image' refers to some bounded region of a scene as shown below.

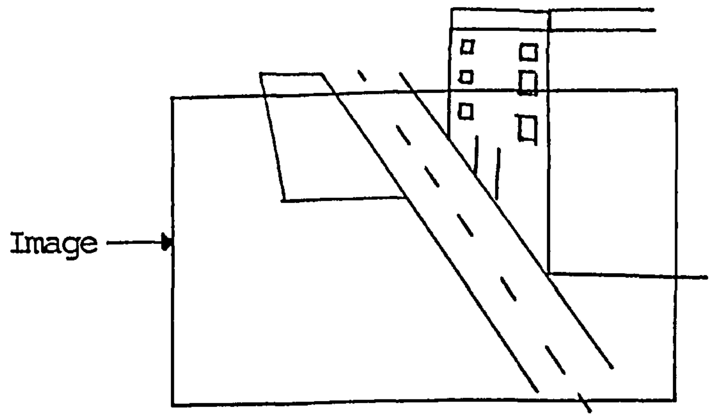


Fig. 1-1 Image Defined as Region of Some Scene

The image itself is a 2-D light intensity function denoted $f(x,y)$ [2]. The value or amplitude of $f(x,y)$ at spatial coordinates (x,y) gives the intensity of the image at that point.

The function $f(x,y)$ must be non-zero and finite, i.e.;

$$0 < f(x,y) < \infty \qquad (1.1\text{-}1)$$

As a simple model, $f(x,y)$ is represented as the product of two components, the components being illumination and reflection [3]. The illumination component is the amount of source light incident on the scene while the reflection component is the amount of light being reflected by the objects in the scene. The two components are denoted $i(x,y)$ and $r(x,y)$ respectively and are given in equation (1.1-2).

$$f(x,y) = i(x,y)\, r(x,y) \tag{1.1-2}$$

where $$0 < i(x,y) < \infty \tag{1.1-3}$$

and $$0 < r(x,y) < 1 \tag{1.1-4}$$

In equation (1.1-3) the illumination is bounded by infinity since infinite incident light is not attainable.

In (1.1-4) 0 corresponds to total absorption while 1 suggests total reflection. The component $r(x,y)$ is determined by the objects in the scene while $i(x,y)$ is irrespective of the scene.

1.2. Digital Image Processing

A digital image is obtained from the function $f(x,y)$ by digitizing $f(x,y)$ both spatially and in amplitude [4]. The spatial digitization is accomplished by image sampling, while amplitude digitization is achieved by gray-level quantization.

If the continuous image $f(x,y)$ is sampled by equally spaced samples to form an $N \times N$ array in which each element of the array is a discrete quantity, as shown in eqn.(1.2-1), then a digital image is formed.

$$f(x,y) \simeq \begin{bmatrix} f(0,0) & f(0.1) & \cdots & f(0,N) \\ f(1,0) & f(1,1) & \cdots & f(1,N) \\ \vdots & & & \\ f(N,0) & f(N,1) & \cdots & f(N,N) \end{bmatrix} \qquad (1.2\text{-}1)$$

Each element of the array is called a pixel or pel.

The digitization process requires a choice of the number of gray-levels each pixel may assume (G) as well as the number of samples of $f(x,y)$ (N,N). In digital image processing these quantities are almost always made equal to some power of two, i.e.;

$$N = 2^n \qquad (1.2\text{-}2)$$

and

$$G = 2^m \qquad (1.2\text{-}3)$$

where G is the number of gray-levels and n and m are integer numbers.

It is obvious that the larger G and N are, the closer the relationship in eqn.(1.2-1) becomes [18-20].

Figure 1-2 will be used to explain the formulation of edges in images.

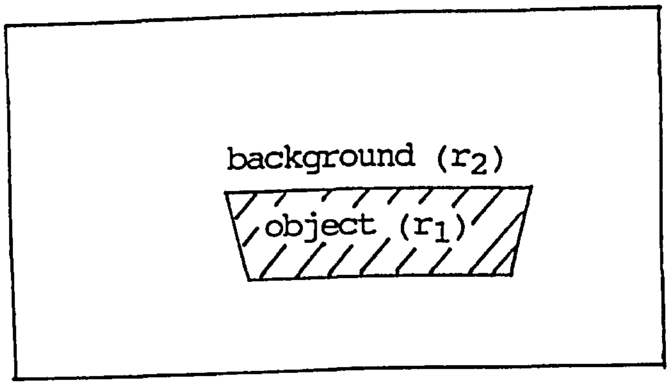


Fig. 1-2 Simple Image

In Fig.1-2 constant illumination over the image scene is assumed. The object has reflectivity r_1 while the back-ground has r_2 and $r_2 < r_1$. Therefore the image intensity at points on the object is greater than the intensity of the background of the image. At points on the edge of the object, the intensity of f(x,y) has an abrupt transition. This abrupt transition is characteristic of edges in continuous images and consequently digital images [5]. However, in digital images, edges are characterized by abrupt changes in pixel value.

1.3. Edge Enhancement Techniques

The word 'image' will be used synonymously with 'digital image' unless otherwise stated.

Since an edge point is a pixel location at which an abrupt change in its gray-level occurs, then in general an edge detection scheme would be to measure the gradient of the image. Two classes of edge enhancement (detection) operators based on the above concept are (i) Gradient Operators and (ii) Compass Operators [6-8].

For digital images, these operators, or masks, represent finite differences.

Gradient operators are expressed as a pair of masks H_1, H_2 which measure the gradient of the image f(m,n) in two orthogonal directions x and y. Therefore the gradient vector is expressed as;

$$g(m,n) = \left[g_1^2(m,n) + g_2^2(m,n) \right]^{1/2} \qquad (1.3\text{-}1)$$

$$\theta_g(m,n) = \tan^{-1} \left[\frac{g_2(m,n)}{g_1(m,n)} \right] \qquad (1.3\text{-}2)$$

where g(m,n) is the magnitude and $\theta_g(m,n)$ is the direction. The values $g_1(m,n)$ and $g_2(m,n)$ are the gradients in the x and y directions respectively. The magnitude is often expressed as in eqn.(1.3-3) for its ease of implementation on digital machines.

$$g(m,n) = | g_1(m,n) | + | g_2(m,n) | \qquad (1.3\text{-}3)$$

A list of some common gradient operators is given in Table 1-1. Note that for a uniform region on the image f(m,n) the operators yield a zero value.

A pixel location (m,n) can be declared an edge point if g(m,n) exceeds some threshold value 't'. By thresholding, an edge map e(m,n) can be created of the image f(m,n) as shown below [11-13].

$$e(m,n) = \begin{cases} 1 \; , \; g(m,n) > `t' & (1.3\text{-}4) \\ 0 \; , \; g(m,n) \leq `t' & (1.3\text{-}5) \end{cases}$$

Usually `t' is chosen such that 5 to 10 percent of pixels with largest gradients are declared edges.

	H_1	H_2
Roberts	$\begin{bmatrix} [0] & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} [1] & 0 \\ 0 & -1 \end{bmatrix}$
Smoothed (Pre-Witt)	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & [0] & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Isotropic	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & [0] & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & [0] & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

Table 1-1 - Common Gradient Operators

Compass operators measure the gradient in a selected number of directions [8]. Table 1-2 shows four compass operators for North going edges. An anticlockwise circular shift of the 8 boundary elements gives a 45° rotation of the gradient direction. As an example, for mask #1 the 8 rotations are;

$$\begin{matrix}1&1&1\\1&-2&1\\-1&-1&-1\end{matrix}\ (\mathrm{N})\qquad \begin{matrix}1&1&1\\1&-2&-1\\1&-1&-1\end{matrix}\ (\mathrm{NW})\qquad \begin{matrix}1&1&-1\\1&-2&-1\\1&1&-1\end{matrix}\ (\mathrm{W})\qquad \begin{matrix}1&-1&-1\\1&-2&-1\\1&1&1\end{matrix}\ (\mathrm{SW})$$

$$\begin{matrix}-1&-1&-1\\1&-2&1\\1&1&1\end{matrix}\ (\mathrm{S})\qquad \begin{matrix}-1&-1&1\\-1&-2&1\\1&1&1\end{matrix}\ (\mathrm{SE})\qquad \begin{matrix}-1&1&1\\-1&-2&1\\-1&1&1\end{matrix}\ (\mathrm{E})\qquad \begin{matrix}1&1&1\\-1&-2&1\\-1&-1&1\end{matrix}\ (\mathrm{NE})$$

If we let $g_k(m,n)$ denote the compass gradient in the direction $\theta_k = \pi/2 + k\,\pi/4$, $k = 0, \ldots, 7$, then the gradient at (m,n) is defined as;

$$g(m,n) = \max_k [\, | g_k(m,n) | \,] \qquad (1.3\text{-}6)$$

As in the case of the previously discussed gradient technique, an edge map may be obtained by thresholding the gradient. For higher angular resolution the size of the compass gradient mask may be increased.

$$1 \begin{bmatrix} 1 & 1 & 1 \\ 1 & [-2] & 1 \\ -1 & -1 & -1 \end{bmatrix} \qquad 2 \begin{bmatrix} 5 & 5 & 5 \\ -3 & [0] & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \text{KIRSCH}$$

$$3 \begin{bmatrix} 1 & 1 & 1 \\ 0 & [0] & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad 4 \begin{bmatrix} 1 & 2 & 1 \\ 0 & [0] & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Table 1-2 - Various (North) Compass Gradients

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & [4] & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & [8] & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & -2 & 1 \\ -2 & [4] & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Table 1-3 - Three discrete Laplacian Operators

The Laplacian [9] which is the second derivative, is also used to detect edges. The Laplacian is defined in equation (1.3-7) and three discrete approximations to this equation are given in Table 1-3.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (1.3\text{-}7)$$

Since the second derivative is involved, this operator is more sensitive to noise than the previously defined gradient techniques. Figure 1-3 demonstrates the effect of applying the Laplacian operator to an edge.

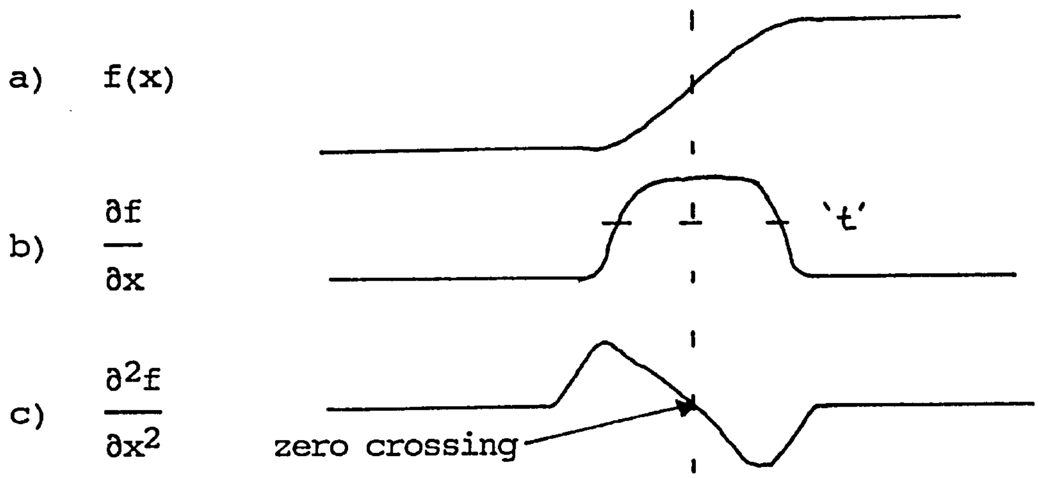


Fig. 1-3 - 1-D Example of First and Second Derivatives

A better technique for the use of the Laplacian is to use the zero crossings as an edge detection principle.

One proposed method [9] approximates the Laplacian of the Gaussian function and is defined as

$$h(m,n) = \left| 1 - \frac{k(m^2 + n^2)}{2\,\sigma^2} \right| \exp \left[- \frac{m^2 + n^2}{2\,\sigma^2} \right] \qquad (1.3\text{-}8)$$

In equation (1.3-8) σ^2 controls the mask size and k is a normalization constant so that the sum of the elements in the mask is zero.

Another useful technique involves Stochastic Gradients [11]. These are used when the image is heavily corrupted by noise. These techniques however require that a mask be designed based upon observations taken over various regions of the image.

Stochastic techniques are beyond the scope of this thesis and will not be discussed further. Details of such schemes are described in [11].

1.4. Problem Statement

The problem is to explore the PCF technique as an edge enhancement method and to examine its degree of variability and applicability. A comparison between the PCF technique and various proven methods must also be performed. A fundamental aspect is also to demonstrate 2-D recusive digital filter design to carry out PCF.

1.5. Thesis Organization

Chapter 2 is a theoretical explanation of PCF with comparison to various gradient techniques. Digital images processed by the PCF method and other techniques are presented.

Chapter 3 deals with the methodology developed for the design of 2-D digital filters required for PCF and a development for a design technique is presented.

Chapter 4 presents a comparative study, for the purpose of edge detection, between the PCF technique and various gradient techniques.

Chapter 5 provides a summary and conclusion of the research material covered in this thesis with suggestions for extensions and future work.

CHAPTER 2

PHASE CONTRAST FILTERING FOR EDGE ENHANCEMENT

2.1. Background Information

In sections 1.2 and 1.3 a background of image theory and characteristics of edges was presented. Figure 2-1 represents a typical cross section of a continuous image f(x,y) with two edges being defined.

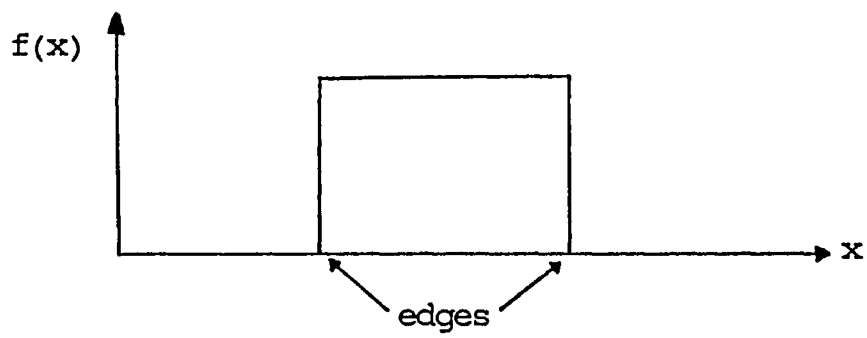


Fig. 2-1 - Cross Section of Arbitrary Continuous Image

From Fourier Series theory [17], any periodic function f(x) can be expressed as an infinite sum consisting of a fundamental wave and its harmonics.

Figure 2-2 shows an ideal square wave and one consisting of components up through the fifth harmonic.

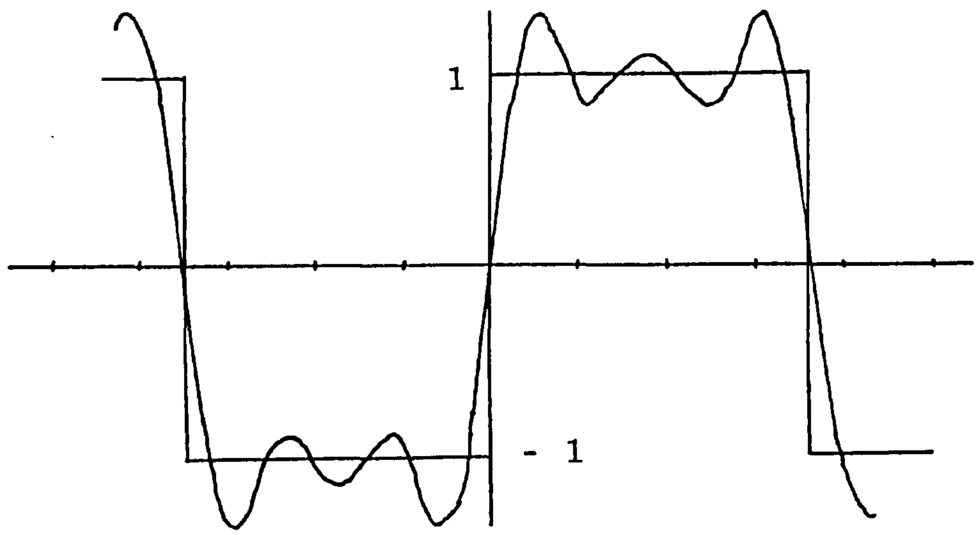


Fig. 2-2 - Ideal Square Wave and Fourier Representation Based Upon components Up To Fifth Harmonic

The absence of high frequency components has distorted the square wave to the effect shown above. In a digital image the absence of these components would likewise disfigure or blur the edges in the image. Consequently, edge information in an image is characterized by high frequency components [3,4].

2.2. Phase Contrast Filter

The Phase Contrast Filtering technique attempts to extract the high frequency components of an image via the scheme shown in Fig. 2-3.

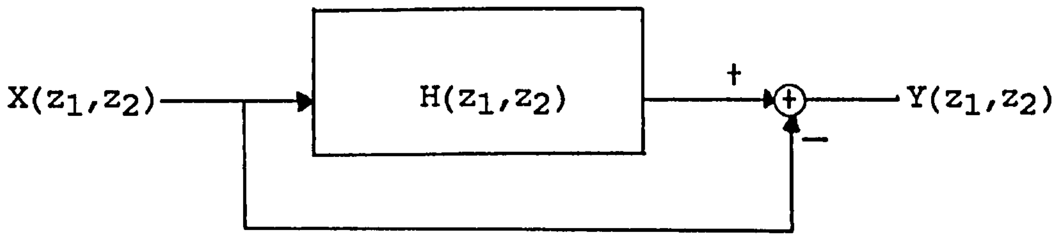


Fig. 2-3 - Phase Contrast Filtering Technique

In the block diagram of Fig.2-3, $X(z_1, z_2)$ represents a digital image, $H(z_1, z_2)$ represents a 2-D digital filter with specified magnitude and phase response and $Y(z_1, z_2)$ represents the resulting edge enhanced image.

The ideal magnitude and phase characteristics of $H(z_1, z_2)$ are given in Fig.2-4. The substitution $z = e^{j\omega T}$ [13-15] has been made and the response over the quarter plane $\omega_1(+)$ and $\omega_2(+)$ has been shown for normalized frequency, i.e.; $0 \leq \omega_i \leq \pi$, i = 1, 2.

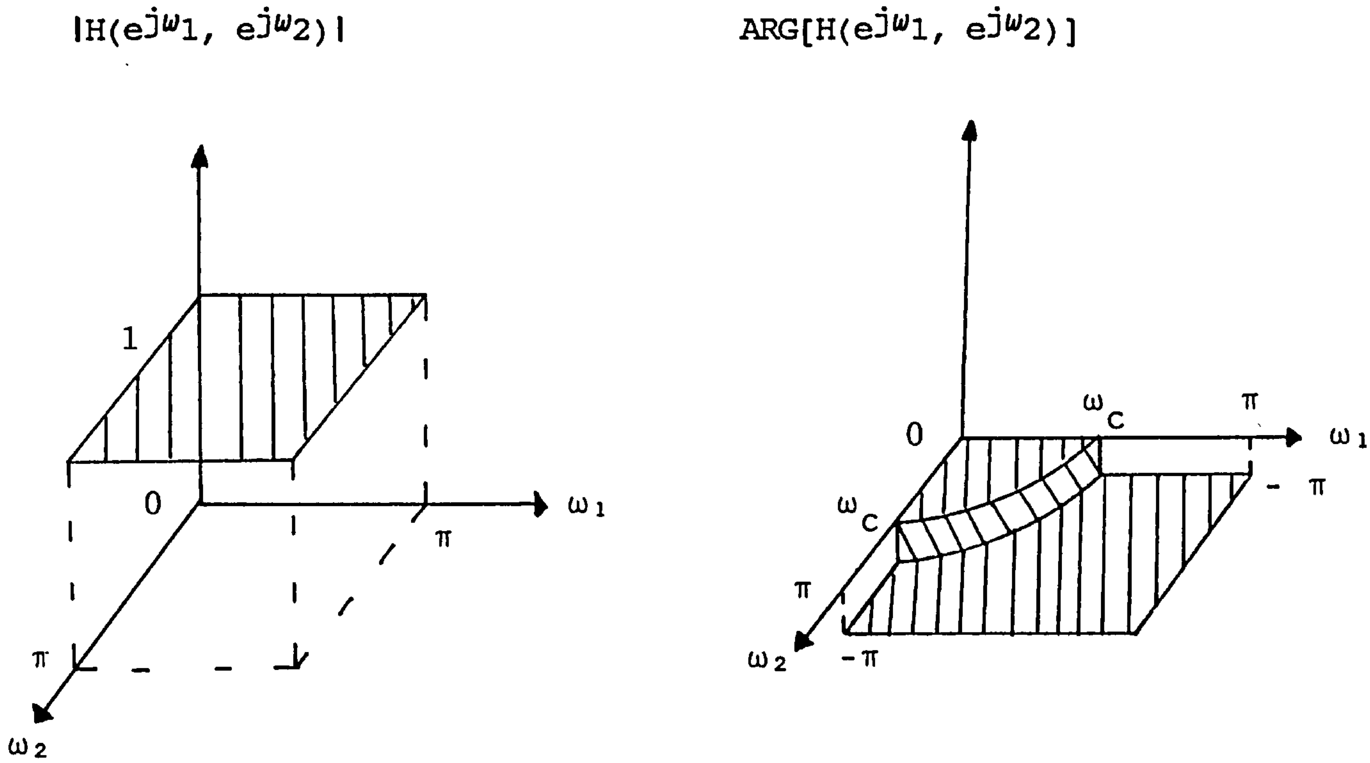


Fig. 2-4 - Ideal Magnitude and Phase Response for $H(z_1, z_2)$

The magnitude response of the filter is required to be unity over the entire frequency range in order that the magnitudes of the frequency components in the image $X(z_1, z_2)$ are unchanged after filtering. The phase response is zero up to some chosen cut-off frequency ω_c, and equal to $-\pi$ radians (-180°) for frequencies beyond ω_c in the $+\omega_1$ and $+\omega_2$ directions. Examining the block diagram of Fig. 2-3, if the image $X(z_1, z_2)$ is filtered by $H(z_1, z_2)$ with magnitude and phase responses given by Fig.2-4 the resulting filtered image will have equal magnitude as the original but frequency components above ω_c will be 180 degrees out of phase with the original. Consequently, when the subtraction operation is performed, the magnitude of frequency components below ω_c will be equal to zero while those above ω_c will be doubled. Analytically;

let $\omega_1 = e^{j\omega_1}$ and $\omega_2 = e^{j\omega_2}$

$$Y(\omega_1, \omega_2) = X(\omega_1, \omega_2)\, H(\omega_1, \omega_2) - X(\omega_1, \omega_2) \tag{2.2-1}$$

where

$$H(\omega_1, \omega_2) = |\, H(\omega_1, \omega_2)\, |\ \underline{/\mathrm{ARG}\ (H(\omega_1,\ \omega_2))} \tag{2.2-2}$$

and $$|H(\omega_1, \omega_2)| = 1\ ,\quad 0 \le \omega_i \le \pi\ ,\ i = 1,\ 2 \tag{2.2-3}$$

$$\mathrm{ARG}(H(\omega_1, \omega_2)) = \begin{cases} 0\ , & 0 \le \sqrt{\omega_1^2 + \omega_2^2} < \omega_c \\ -\pi, & \omega_c \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases} \tag{2.2-4}$$

therefore

$$X(\omega_1, \omega_2)\, H(\omega_1, \omega_2) = \begin{cases} X(\omega_1, \omega_2), & 0 \le \sqrt{\omega_1^2 + \omega_2^2} < \omega_c \\ -X(\omega_1, \omega_2), & \omega_c \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases} \tag{2.2-5}$$

$$Y(\omega_1, \omega_2) = \begin{cases} 0 & , 0 \leq \sqrt{\omega_1^2 + \omega_2^2} < \omega_c \\ -2 \bullet X(\omega_1, \omega_2), & \omega_c \leq \sqrt{\omega_1^2 + \omega_2^2} < \pi \end{cases}$$

(2.2-6)

The resulting image $Y(z_1, z_2)$ will contain only the high frequencies (edges) of the original image $X(z_1, z_2)$.

The phase response in Fig. 2-4 is shown to be circular symmetric over the quarter plane defined by $\omega_1(+)$ and $\omega_2(+)$ [21-22]. However, square symmetry as well as circular symmetry for the phase response were both examined. The results shall be presented at the end of the chapter.

2.3. Reduction of The PCF Technique

The PCF technique presented in Fig. 2-3 may be reduced to a single block as follows:

$$Y(z_1, z_2) = X(z_1, z_2)\, H(z_1, z_2) - X(z_1, z_2) \tag{2.3-1}$$

$$Y(z_1, z_2) = X(z_1, z_2)\, [H(z_1, z_2) - 1] \tag{2.3-2}$$

let $$H'(z_1, z_2) = H(z_1, z_2) - 1 \tag{2.3-3}$$

then $$H'(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} \tag{2.3-4}$$

The reduced block diagram is shown in Fig. 2-5 where $H'(z_1, z_2)$ is given by (2.3-4) and $X(z_1, z_2)$ and $Y(z_1, z_2)$ are unchanged.

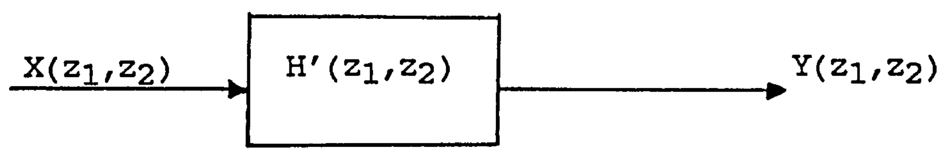


Fig. 2-5 - Reduction of the PCF Technique

Substituting $z_1 = e^{j\omega_1}$ and $z_2 = e^{j\omega_2}$, the real and imaginary parts of $H'(z_1, z_2)$ are given by:

$$R\left[H'\left(e^{j\omega_1}, e^{j\omega_2}\right)\right] = R\left[H\left(e^{j\omega_1}, e^{j\omega_2}\right)\right] - 1 \qquad (2.3\text{-}5)$$

and

$$I\left[H'\left(e^{j\omega_1}, e^{j\omega_2}\right)\right] = I\left[H\left(e^{j\omega_1}, e^{j\omega_2}\right)\right] \qquad (2.3\text{-}6)$$

The ideal magnitude and phase responses of $H(e^{j\omega_1}, e^{j\omega_2})$ are unchanged. The ideal responses for $H'(e^{j\omega_1}, e^{j\omega_2})$ are shown in Fig. 2-6.

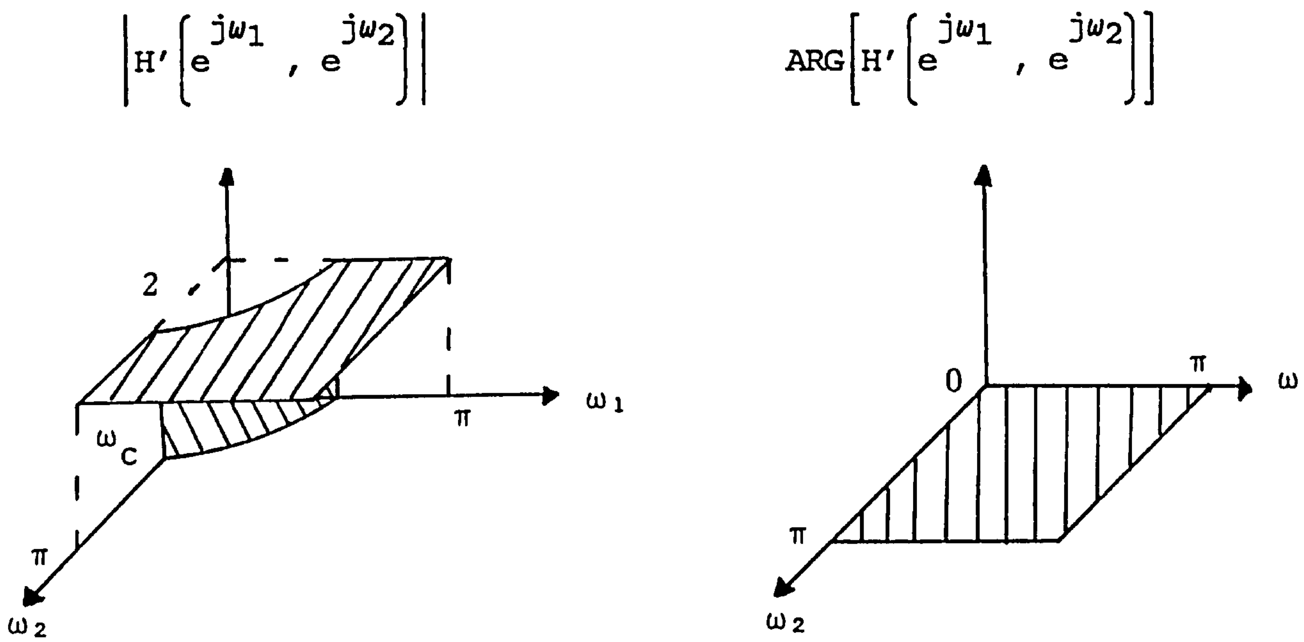


Fig. 2-6 - Ideal Magnitude and Phase Responses
For $H'(e^{j\omega_1}, e^{j\omega_2})$

The magnitude response is that of an ideal highpass filter with a gain of 2 and the phase response is zero over the frequency quarter plane $\omega_1(+)$ and $\omega_2(+)$.

Discrete 2-D digital filters were designed based on the ideal response of $H'(z_1, z_2)$ and implemented via the block diagram in Fig. 2-5. This reduced method was abandoned because of poor edge enhancement results which were due to large discrepancies between the desired phase response and the designed phase response.

2.4. Phase Contrast Filtering In The Frequency Domain

Convolution in the time domain is the equivalent to multipliction in the frequency domain and vice versa [19,20]. The filtering of images in the spatial domain is a convolution process and is equivalent to multiplying together the Discrete Fourier Transforms of the image and the filter. The 2-D convolution theorem is given in equations (2.4-1) and (2.4-2).

$$x(n,m) * h(n,m) \Longleftrightarrow X(u,v)\ H(u,v) \qquad (2.4\text{-}1)$$

$$x(n,m)\ h(n,m) \Longleftrightarrow X(u,v) * H(u,v) \qquad (2.4\text{-}2)$$

where * denotes convolution.

Due to this property, many functions which are not attainable in the time domain, the ideal filter response given in Fig. 2-4, are precisely represented in the frequency domain.

The PCF technique may be implemented in the frequency domain through the use of the FFT and IFFT.

The 2-D DFT of the image X(n,m) of size N x N is obtained (see section 3.3) and expressed as;

$$X(u,v) = R(u,v) + j\ I(u,v) \qquad (2.4\text{-}3)$$

for $u,v = 0, 1, 2, \ldots, N-1$

where R(u,v) and I(u,v) are the real and imaginary components respectively. The magnitude and phase are obtained by;

$$|X(u,v)| = [(R(u,v))^2 + (I(u,v))^2]^{1/2} \qquad (2.4\text{-}4)$$

$$ARG[X(u,v)] = TAN^{-1}[I(u,v) \;/\; R(u,v)] \qquad (2.4\text{-}5)$$

The phase is altered by the addition of $\theta(u,v)$ where;

$$\theta(u,v) = -\pi \exp[D_O/D(u,v)] \qquad (2.4\text{-}6)$$

and

$$D(u,v) = [u^2 + v^2]^{1/2} \qquad (2.4\text{-}7)$$

for $\quad u,v = 0, 1, 2, \ldots, N-1$

A cross section of the function $\theta(u,v)$ is given in Fig. 2-7.

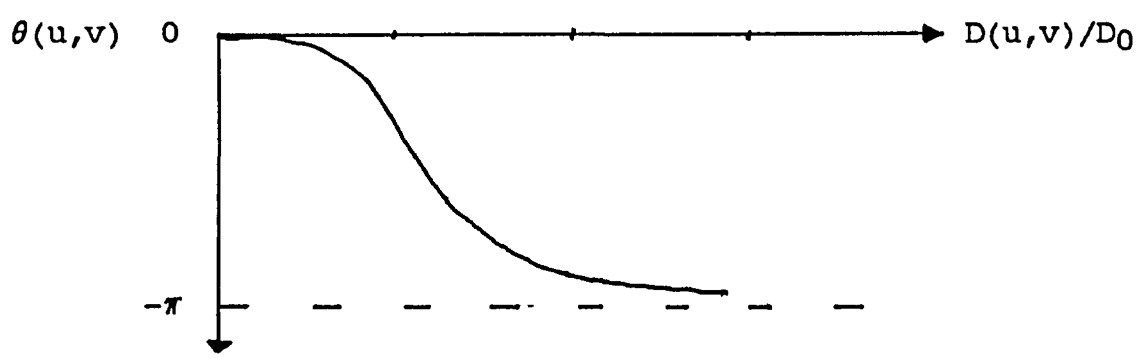


Fig. 2-7 - Cross Section of $\theta(u,v)$

The ideal phase response shown in Fig. 2-4 would not be used because of the ringing effects it produces [4,28].

The real part and imaginary part of the modified image is obtained by;

$$R'(u,v) = |X(u,v)| \bullet \cos[ARG(X(u,v)) + \theta(u,v)] \qquad (2.4\text{-}8)$$

$$I'(u,v) = |X(u,v)| \bullet \sin[ARG(X(u,v)) + \theta(u,v)] \qquad (2.4\text{-}9)$$

and

$$H'(u,v) = R'(u,v) + j\ I'(u,v) \qquad (2.4\text{-}10)$$

for $\quad u,v = 0, 1, 2, \ldots, N-1$

In order that the result of the application of the Inverse Discrete Fourier Transform (IDFT) to H'(u,v) be real, Hermitian Symmetry conditions must be maintained [25]. Therefore the array H'(u,v) must be extended to size 2N x 2N before application of the IDFT. See section 3.3 for details of the array extension and the definition of Hermitian Symmetry.

Frequency domain techniques produced results similar to that produced by the 2-D recursive digital filter where coefficients are given in Table 2-1. Their major difficulty is the large times and memory required for computing the 2-D FFT and IFFT for images of size 64 x 64 pixels this approach required about 4 minutes of CPU time on an IBM-AT, whereas the filtering approach, using order 1 2-D recursive digital filters, takes less than 15 seconds to filter images of size 256 x 256 pixels.

2.5. Results and Comments

The results of applying the Sobel operator and the PCF technique, with two different cut-off frequencies, to two images are shown in Figures 2-8 and 2-9. The images are 256 x 256 pixels with 8 bit resolution.

The Sobel operator enhances the major edges of the original image but fine details on the images are not as pronounced. The PCF technique enhances even the slight variations of the image as can be seen in Fig. 2-8 (c) and (d). The effect of increasing ω_c in the PCF technique is also presented.

Table 2-1 shows the filter coefficients used for the PCF technique of Figures 2-8 (c) and (d) and 2-9 (c) and (d).

	$\omega_c = 1.0$	$\omega_c = 1.75$
a_{00}	-3.095840	-1.634320
a_{01}	2.396338	1.328521
a_{10}	2.396338	1.328521
a_{11}	-1.553688	-0.237538
b_{00}	1.0	1.0
b_{01}	-0.616835	-0.224075
b_{10}	-0.616835	-0.224075
b_{11}	0.446490	0.252944

Table 2-1 Filter Coefficients Used For PCF Technique

Both of the above filters are circular symmetric over the quarter plane $\omega_1(+)$ and $\omega_2(+)$. The PCF technique was applied to images using a filter with a square symmetric response. This type of symmetry produces an image in which edge enhancement is lacking in various directions. As a result a circular symmetric filter response is preferred for PCF edge enhancement.

(a)

(b)

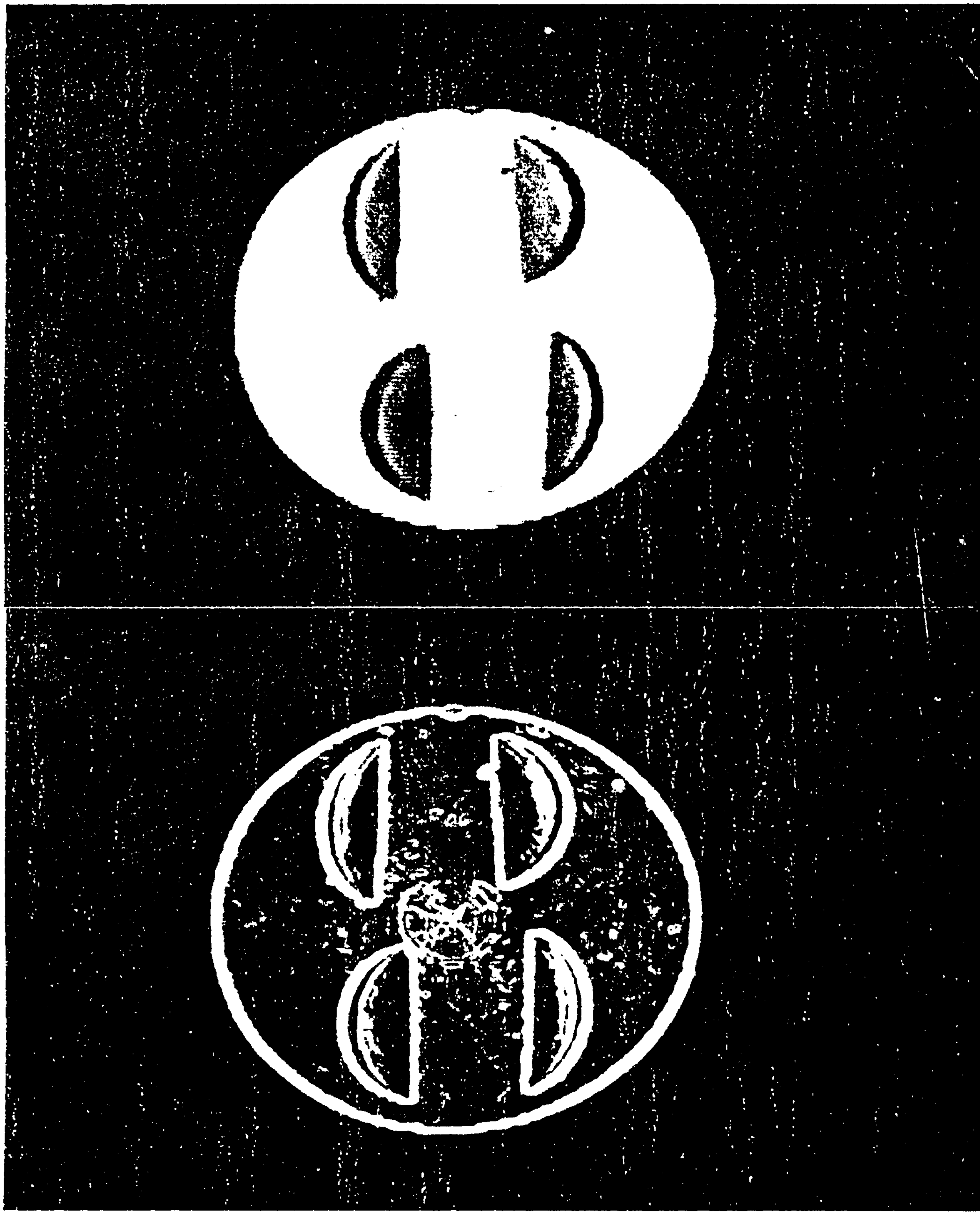
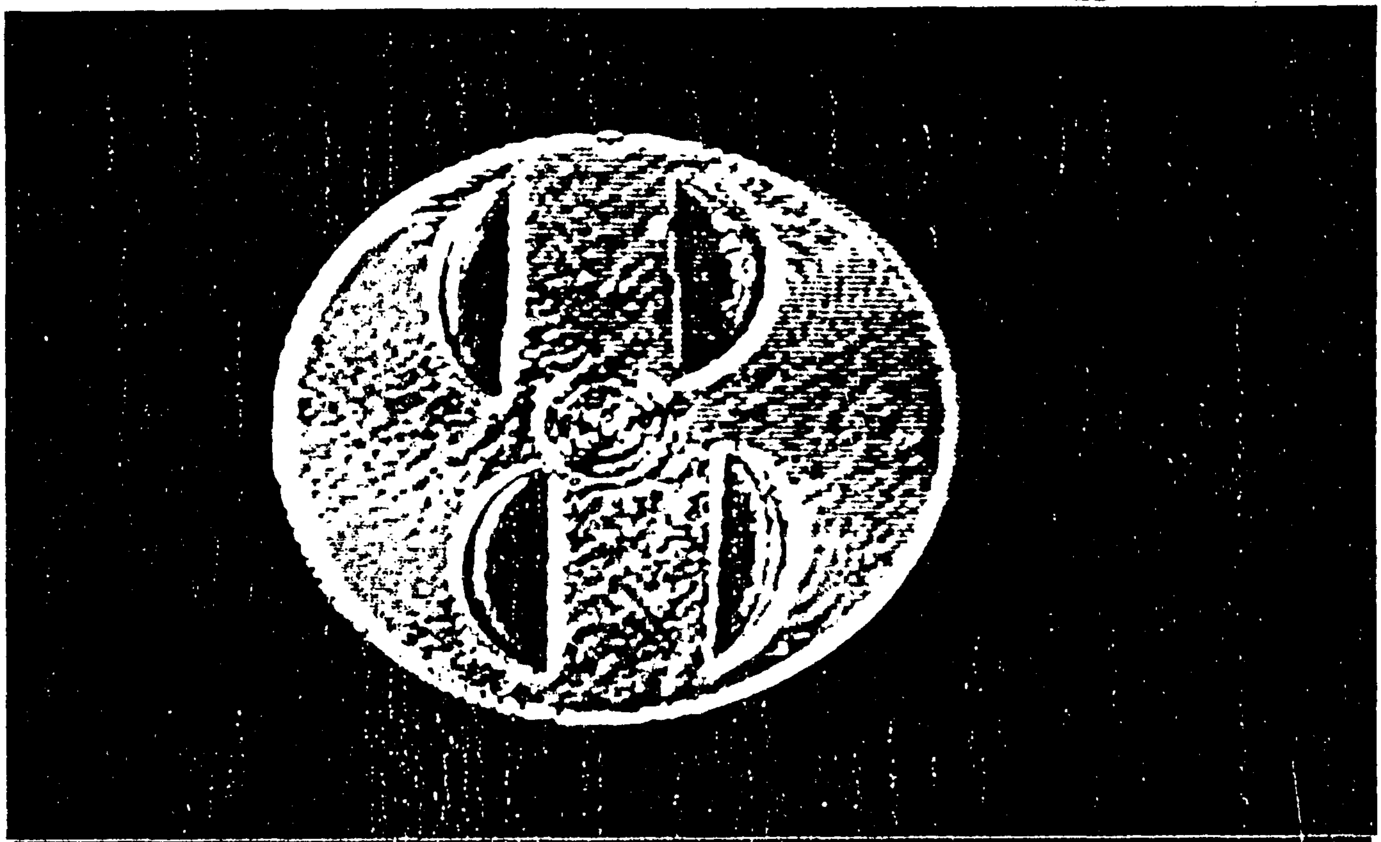


Fig. 2-8 - (a) Original Image

(b) Image Processed by Sobel Operator

(c)



(d)

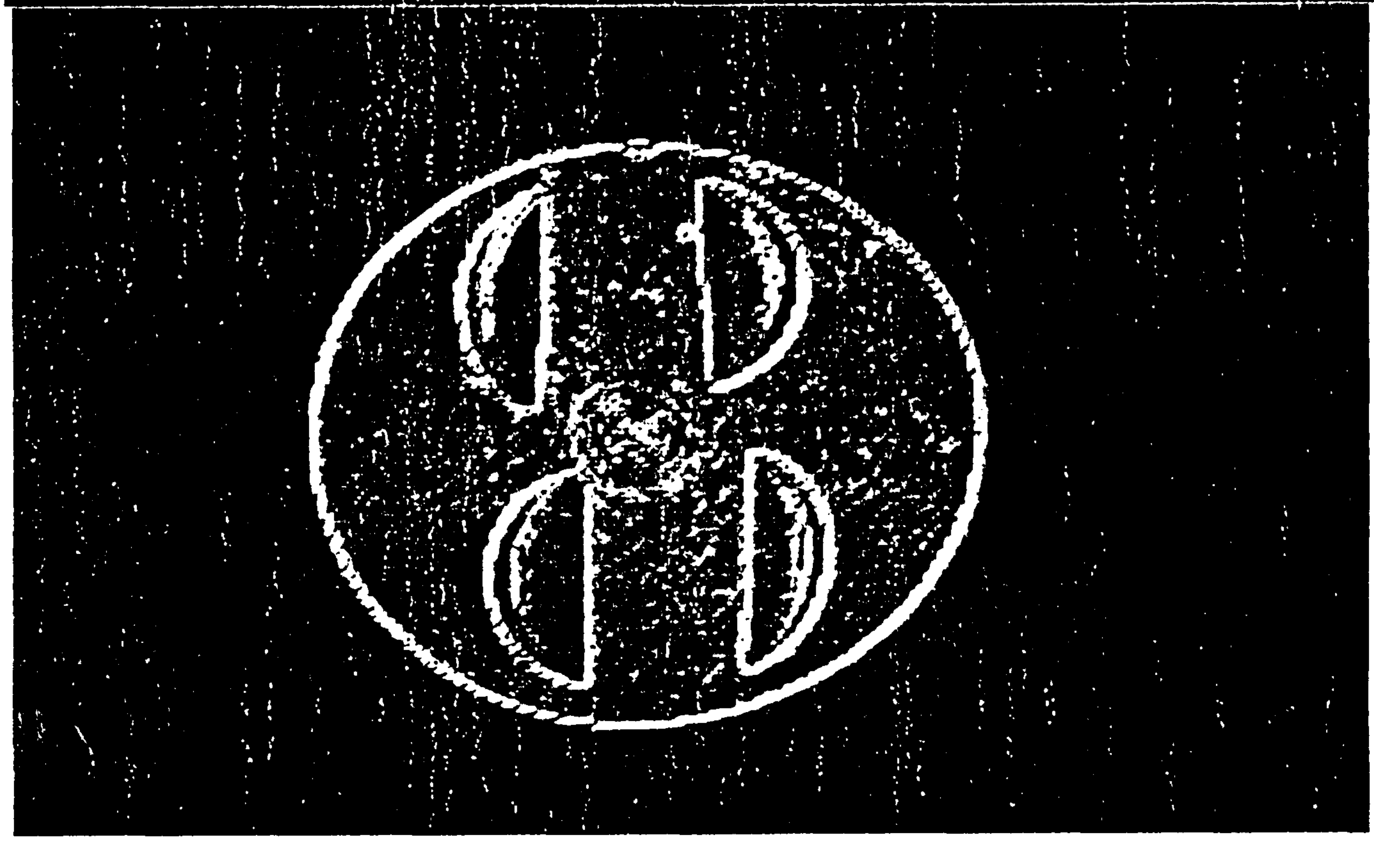


Fig. 2-8 - (c) Image Processed by PCF with ω_C = 1.0

(d) Image Processed by PCF with ω_C = 1.75

(a)

(b)

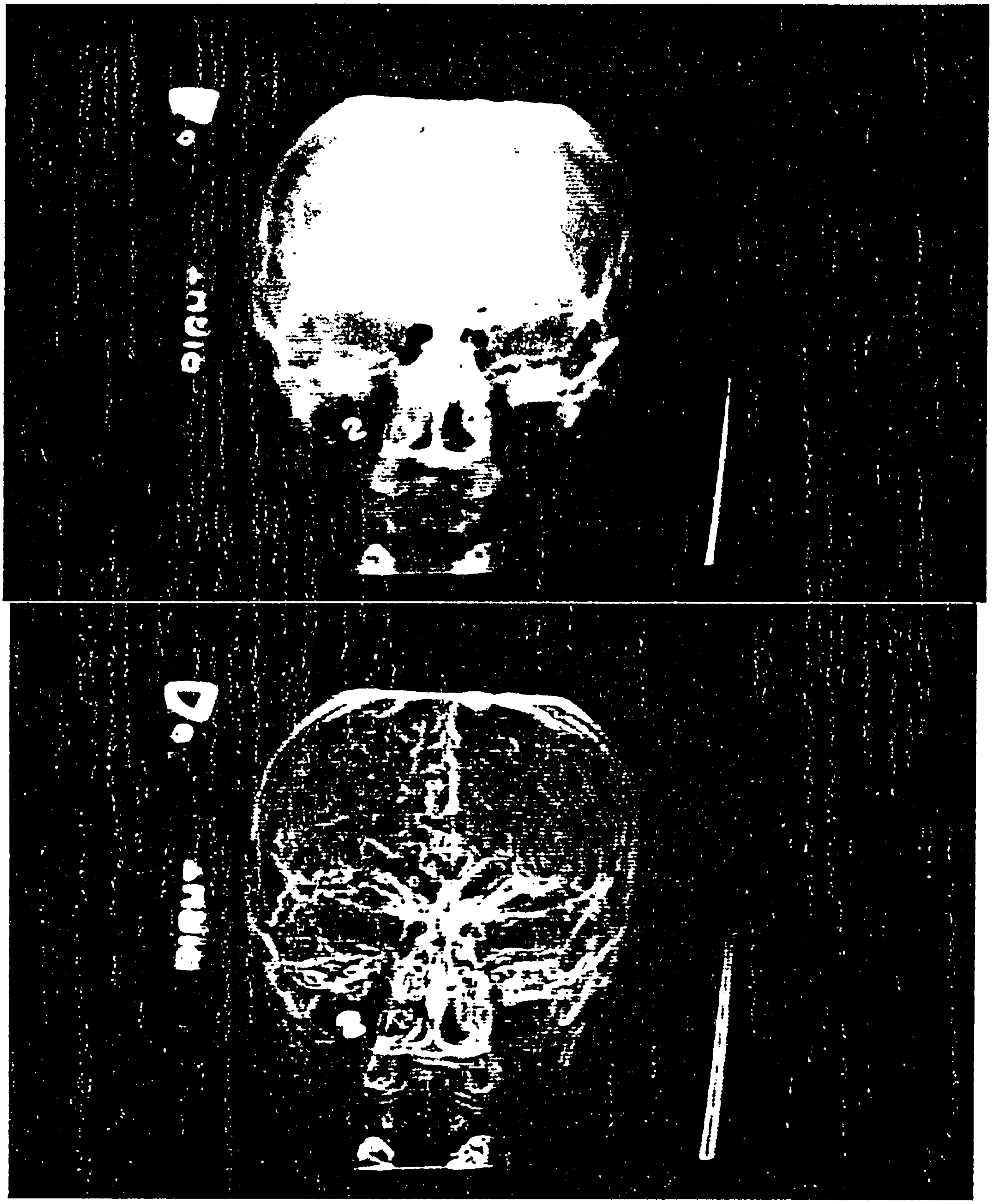


Fig. 2-9 - (a) Original Image

(b) Image Processed by Sobel Operator

(c)

(d)

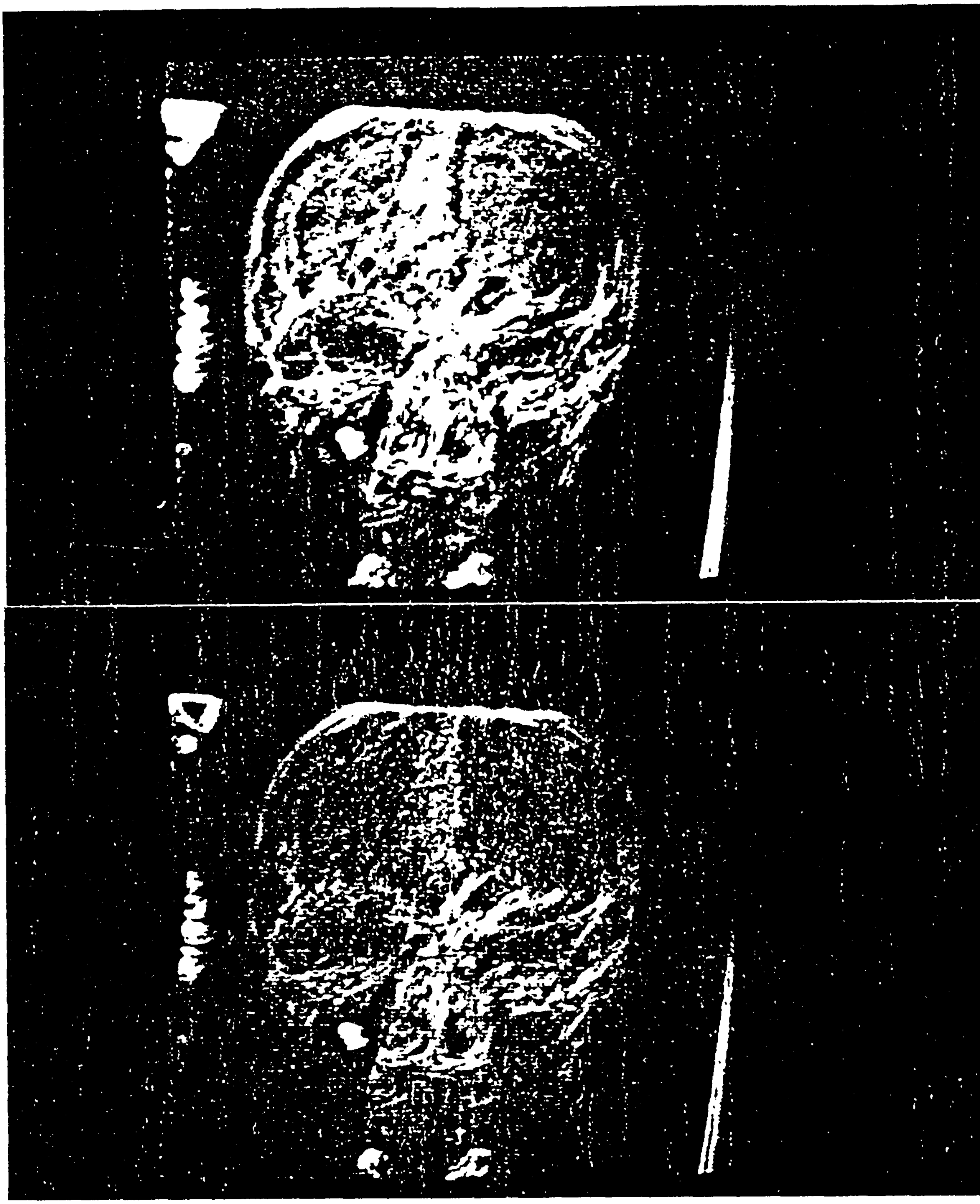


Fig. 2-9 - (c) Image Processed by PCF with $\omega_c = 1.0$

(d) Image Processed by PCF with $\omega_c = 1.75$

CHAPTER 3

FILTER DESIGN FROM MAGNITUDE AND PHASE INFORMATION

3.1. Introduction

In this chapter, a design technique will be presented for designing a 2-D digital filter to meet given phase and magnitude specifications.

3.2. Problem Statement

The desired magnitude and phase responses are specified in the frequency domain, see Fig. 2-4. The problem is to design a 2-D spatial (discrete time) filter which has an impulse response that approximates the specified impulse response of a PCF.

3.3. Design Procedure

The design technique utilized minimizes the error between the ideal impulse response, obtained from the magnitude and phase specifications, and the general case impulse response of a 2-D digital filter of order N.

With magnitude and phase specified over the entire frequency plane the impulse response is immediately attainable [27].

The filters frequency response can be separated into real and imaginary components as shown in eqn.(3.3-1) and eqn.(3.3-2) respectively.

$$R\left[H\left(e^{j\omega_1}, e^{j\omega_2}\right)\right] = \left|H\left(e^{j\omega_1}, e^{j\omega_2}\right)\right| \cos\left[\mathrm{ARG}\left(H(e^{j\omega_1}, e^{j\omega_2}\right)\right] \quad (3.3\text{-}1)$$

$$I\left[H\left(e^{j\omega_1}, e^{j\omega_2}\right)\right] = \left|H\left(e^{j\omega_1}, e^{j\omega_2}\right)\right| \sin\left[\mathrm{ARG}\left(H(e^{j\omega_1}, e^{j\omega_2}\right)\right] \quad (3.3\text{-}2)$$

The real and imaginary components can be discretized by sampling R and I over the ω_1 and ω_2 plane as shown in equation (3.3-3) [25].

$$\left.\begin{array}{l} R\left(\omega_{1_u}, \omega_{2_v}\right) \\ I\left(\omega_{1_u}, \omega_{2_v}\right) \end{array}\right\} \quad U, V = 0, 1, 2, \ldots, k-1 \quad (3.3\text{-}3)$$

and

$$H\left(\omega_{1_u}, \omega_{2_v}\right) = R\left(\omega_{1_u}, \omega_{2_v}\right) + jI\left(\omega_{1_u}, \omega_{2_v}\right) \quad (3.3\text{-}4)$$

k is the number samples taken over ω_1 and ω_2 with the sampling increment equal to π/k.

To obtain the impulse response, the Inverse Discrete Fourier Transform (IDFT) is applied to $H(\omega_{1_u}, \omega_{2_v})$.

The DFT and IDFT are given in equations (3.3-5) and (3.3-6) respectively for the 2-D case.

$$H(u, v) = \sum_{n=0}^{K-1} \sum_{m=0}^{L-1} h(n, m) \exp\left[-j2\pi \left(\frac{un}{K} + \frac{vm}{L}\right)\right] \tag{3.3-5}$$

for $u = 0, 1, 2, \ldots, K-1$ and $v = 0, 1, \ldots, L-1$

$$h(n, m) = \frac{1}{KL} \sum_{u=0}^{K-1} \sum_{v=0}^{L-1} H(u, v) \exp\left[j\, 2\pi \left(\frac{un}{K} + \frac{vm}{L}\right)\right] \tag{3.3-6}$$

for $n = 0, 1, 2, \ldots, K-1$ and $m = 0, 1, 2, \ldots, L-1$

If the number of samples in ω_1 is the same as those in ω_2 then $K = L$. For simplification, the substitution

$$u = \omega_{1_u} \tag{3.3-7}$$

and

$$v = \omega_{2_v} \tag{3.3-8}$$

is made and (3.3-4) can be written as (3.3-9).

$$H(u, v) = R(u, v) + j\, I(u, v) \tag{3.3-9}$$

To obtain the impulse response $h(n, m)$, (3.3-9) is substituted into (3.3-6).

$$h(n, m) = \frac{1}{k^2} \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} \left[R(u, v) + j\, I(u, v) \right] \exp\left[\frac{j2\pi}{k} (nu + mv) \right] \tag{3.3-10}$$

for $u, v = 0, 1, 2, \ldots, K-1$

In order to guarantee that h(n, m) be real, the condition that H(u, v) be Hermitian symmetric [25] must be met. The Hermitian Symmetry conditions are given by;

$$R(u, v) = R(k-u, k-v) \tag{3.3-11}$$

$$I(u, v) = -I(k-u, k-v) \tag{3.3-12}$$

$$R(u + k/2, v) = R(u, k-v) \tag{3.3-13}$$

$$I(u + k/2, v) = -I(u, k-v) \tag{3.3-14}$$

for $u, v = 0, 1, 2, \ldots, K/2-1$

H(u, v) is defined over an array of size k x k. In order to maintain the desired H(u, v) and the required symmetry conditions, H(u, v) must be extended to size P x P where

$$P = 2 \bullet k \tag{3.3-15}$$

The extended array is shown in Fig. 3-1.

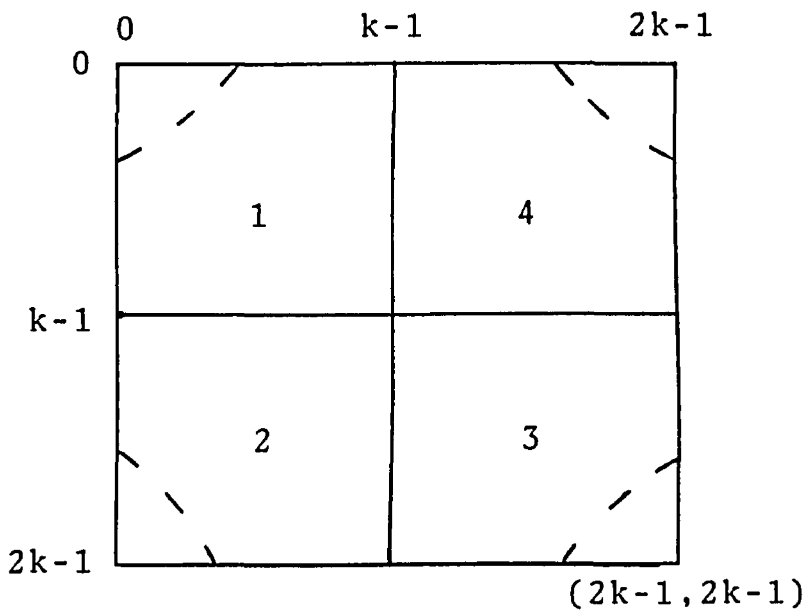


Fig. 3-1 - Extended Response For H(u, v) to Obtain Hermitian Symmetry

Quadrant 1 contains the original k x k samples of H(u, v). Quadrant 3 is obtained by applying equations (3.3-11) and (3.3-12) with P = 2k directly to H(u, v). Quadrant 2 is a shifted version of 1, and quadrant 3 is obtained by the application of (3.3-13) and (3.3-14) to the array in quadrant 2.

The result of the appliction of the IDFT to the extended array is a real impulse response of size P x P. However, the desired ideal k x k impulse response is contained in the first k x k samples of the P x P array.

The general equation for a 2-D digital filter of order N is given in equation (3.3-16) [25,26].

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{\sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij} z_1^{-i} z_2^{-j}}{1 + \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij} z_1^{-i} z_2^{-j}} \qquad (3.3\text{-}16)$$

The condition $i=j\neq 0$ states that i and j cannot be equal to 0 coincidentally.

The 2-D difference equation for $H(z_1, z_2)$ from above can be written as;

$$y(n, m) = \sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij} x(n-i, m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij} y(n-i, m-j) \qquad (3.3\text{-}17)$$

To obtain the impulse response [19-21] of the general case filter of eqn.(3.3-17) the input becomes;

$$x(n, m) \dashrightarrow \delta(n, m)$$

where $\delta(n, m)$ is the 2-D unit impulse with characteristics;

$$\delta(n, m) = \begin{cases} 1, & n = m = 0 \qquad (3.3\text{-}18) \\ 0, & \text{otherwise} \qquad (3.3\text{-}19) \end{cases}$$

With the 2-D unit impulse as the input, the output y(n,m) goes to h(n,m);

$$h(n, m) = \sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij}\, \delta(n-i, m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij}\, h(n-i, m-j) \qquad (3.3\text{-}20)$$

Equation (3.3-20) gives the general case impulse response for a 2-D digital filter of order N.

An error function E is created between the ideal impulse response, denoted $h^I(n,m)$, and the general case impulse response;

$$E = \sum_{n=0}^{K-1} \sum_{m=0}^{K-1} \left[h(n,m) - h^I(n,m) \right]^2 \qquad (3.3\text{-}21)$$

where K is the number of samples of the ideal impulse response.

Substituting (3.3-20) into (3.3-21) yields;

$$E = \sum_{n=0}^{K-1} \sum_{m=0}^{K-1} \left[\sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij}\, \delta(n-i, m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij}\, h(n-i, m-j) - h^I(n, m) \right]^2 \qquad (3.3\text{-}22)$$

Minimization of the error function (E) requires the procurement of the partial derivatives of E with respect to coefficients a_{ij} and b_{ij}. E is minimum when its derivatives with respect to coefficients a_{ij} and b_{ij} are equal to zero. This is shown in equations (3.3-23) through (3.3-25).

$$\frac{\partial E}{\partial a_{00}} = 2 \bullet \sum_{n=0}^{K-1} \sum_{m=0}^{K-1} \left[\sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij}\, \delta(n-i,m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij}\, h(n-i,m-j) - h(n,m) \right] \left[\delta(n,m) \right] = 0 \qquad (3.3\text{-}23)$$

$$\frac{\partial E}{\partial a_{01}} = 2 \bullet \sum_{n=0}^{K-1} \sum_{m=0}^{K-1} \left[\sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij}\, \delta(n-i,m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij}\, h(n-i,m-j) - h(n,m) \right] \left[\delta(n, m-1) \right] = 0 \qquad (3.3\text{-}24)$$

$$\vdots$$

$$\frac{\partial E}{\partial b_{NN}} = 2 \bullet \sum_{n=0}^{K-1} \sum_{m=0}^{K-1} \left[\sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij}\, \delta(n-i,m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij}\, h(n-i,m-j) - h(n,m) \right] \left[h(n-N, m-N) \right] = 0 \qquad (3.3\text{-}25)$$

The result of the above process is a $2 \bullet (N+1)^2 - 1$ system of linear equations which can be solved for the coefficients a_{ij} and b_{ij}.

When solving the above equations the assumption of a casual system is used, that is;

$$h(n,m) = 0 , \quad \text{for } n \text{ or } m < 0 \tag{3.3-26}$$

The solution to the system of linear equations above results in the formulation of a 2-D digital filter with magnitude and phase responses approximating the specified responses.

The implementation of the PCF technique is accomplished by applying the designed filter H(u,v) to a digital image I(n,m) of size S x S. The filtered image I'(n,m) is expressed as;

$$I'(n,m) = \sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij}\, I(n-i, m-j) - \sum_{\substack{i=0 \\ i=j\neq 0}}^{N} \sum_{j=0}^{N} b_{ij}\, I'(n-i, m-j) \tag{3.3-27}$$

for $n,m = 0, 1, 2, \ldots, S-1.$

The edge enhanced image E(n,m) is obtained by taking the absolute value of the difference between the original image and the filtered image.

$$E(n,m) = |\, I'(n,m) - I(n,m)\, | \tag{3.3-28}$$

for $n,m = 0, 1, 2, \ldots, S-1.$

3.4. Results and Comments

The filter design technique presented in the previous section was used to design filters of order 1, 2 and 3. Order 1 filters produced the results presented in the thesis. Higher-order filters designed to meet the ideal response gave a ringing effect on the edges produced in the image. This same effect was also observed when using the ideal response in the convolution approach described in section 2.4. The designed higher order filters have a magnitude response which peaks near the cut-off frequency, and a phase response which is very close to the ideal. The peaking of the magnitude response could have also contributed to the ringing (or blurring) effect. Magnitude and phase responses of a designed third order filter are shown in Figs. 3-2(a) and (b). The filter coefficients are shown on page 38(a).

(a)

(b)

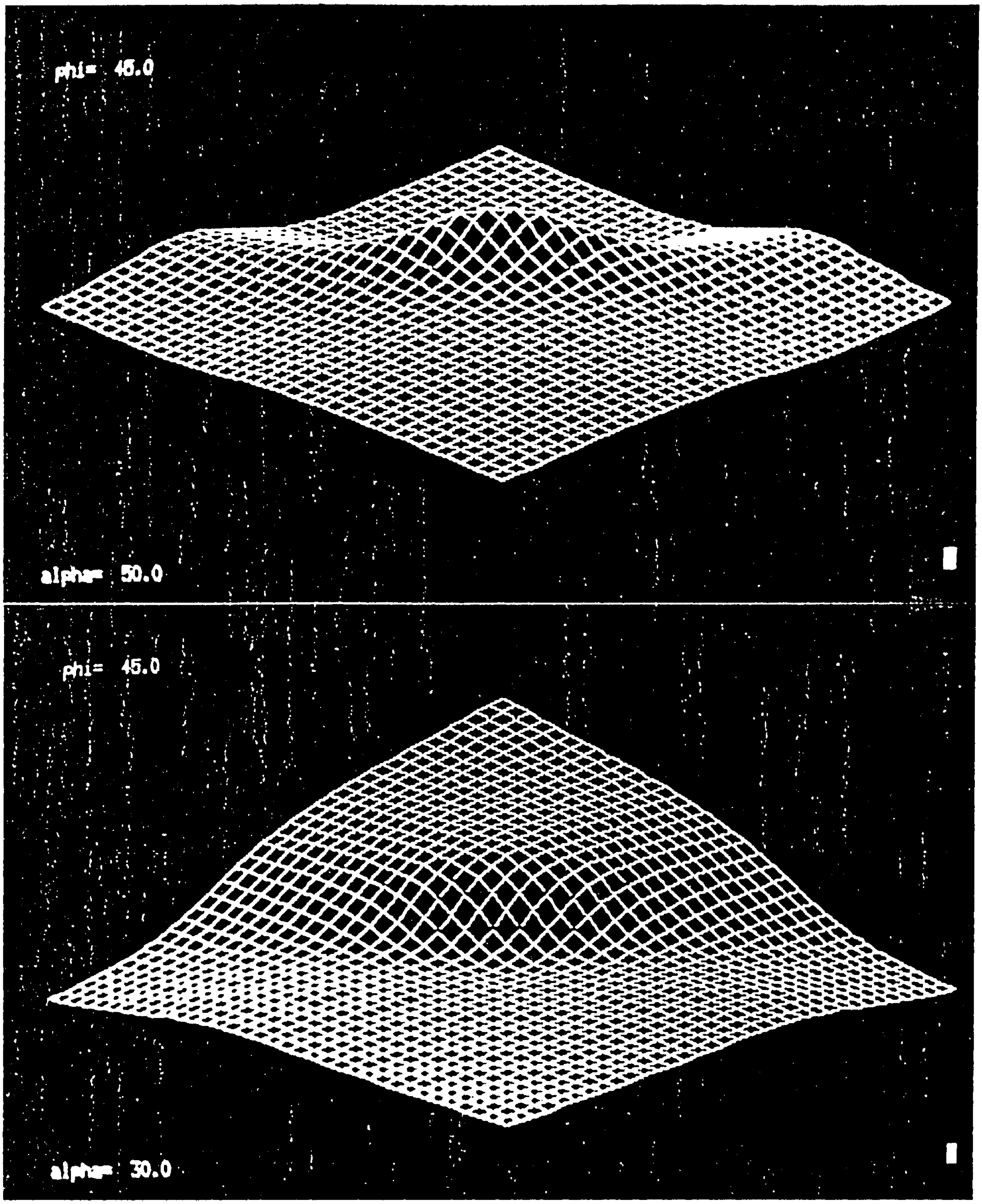


Fig. 3-2 - (a) Magnitude Response of 3rd Order Filter

(b) Phase Response of 3rd Order Filter

CHAPTER 4

A COMPARATIVE STUDY

4.1. Introduction

This chapter will be devoted to examining the phase contrast technique as an edge detector. A comparison of various gradient techniques and the phase contrast approach will be presented.

4.2. Edge Enhancement

In developing an edge detection performance criteria, one should distinguish between necessary information and additional information to be obtained from the detector. As an example, it is necessary for a detector to determine the pixel location of an edge; moreover, it is attractive if it can also provide the slope angle of the edge.

Three major errors involved with edge location detection are shown on the next page.

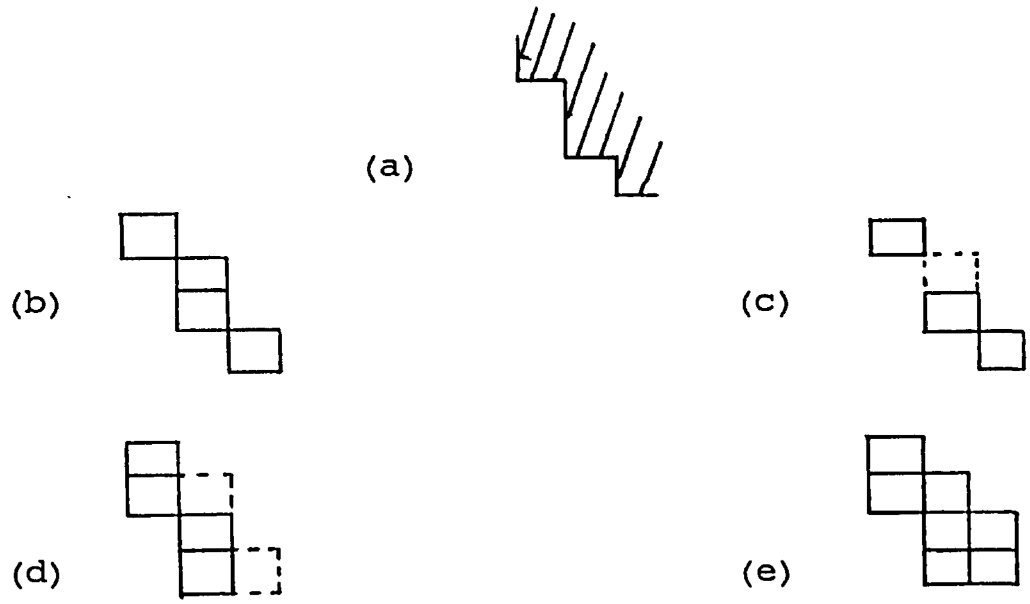


Fig. 4-1 - Edge Location (a) Image Segment; (b) Ideal Detection; (c) Fragmented Detection; (d) Offset Detection; (e) Smeared Detection.

A commonly used figure of merit for edge detection techniques [2,16] is defined by;

$$R = \frac{1}{I_N} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d} \tag{4.2-1}$$

where

$$I_N = MAX(I_I,\ I_A) \tag{4.2-2}$$

I_A represents the number of pixels declared as edge points and I_I is the number of ideal edge points. The scaling factor α is adjustable to penalize edge locations that are local but offset from their true positions. The distance between an ideal edge point and a pixel location declared as an edge point is given by d.

A comparison of various gradient techniques and the phase contrast technique was conducted using the figure of merit given above with $\alpha = 1/9$. The test image, with 8 bit resolution, consisted of a 64 x 64 pixel array with a horizontal edge running across it.

Also added to the image was independent White Gaussian noise with standard deviation σ_n.

The signal to noise ratio (SNR) is defined as;

$$SNR = \frac{h^2}{\sigma_n^2} \tag{4.2-3}$$

where h is the height of the edge.

For each result given in the comparison, the edge detection technique used was optimized to obtain the highest possible value of R. As an example, the threshold values 't' were varied in each case to maximize R.

4.3. Results and Comments

Figure 4-2 gives the results for various SNR ratios, which are summarized in Table 4-1. The edge width for this comparison was set at W = 1 pixel.

Figure 4-3 gives the results for a comparison based on edge width. The edge width (W) was varied from 1 to 4 pixels while the SNR was kept constant at 100.

Of the edge detection schemes compared, other than the PCF, the Sobel operator appears to have the best overall performance and the Roberts the poorest. This was affirmed by examination of images processed by the two techniques.

The graphs of Figures 4-2 and 4-3 seem to show little difference between the Sobel and the PCF for high SNR. Examination of Table 4-1 gives the exact values. The difference between the two techniques is apparent by examination of I_A, the number of pixels declared as edge points. Taking I_A and R both into consideration, for a high SNR the PCF outperforms the Sobel because of the Sobels tendency to give a smeared indication of edge location. This is confirmed in Fig. 4-4.

The original image in Fig. 4-4 consists of objects of different intensity with white Gaussian noise with $\sigma = 4.0$ added to it. The Sobel operator has detected all major edges but the edge map is 'thick' in comparison with the PCF technique.

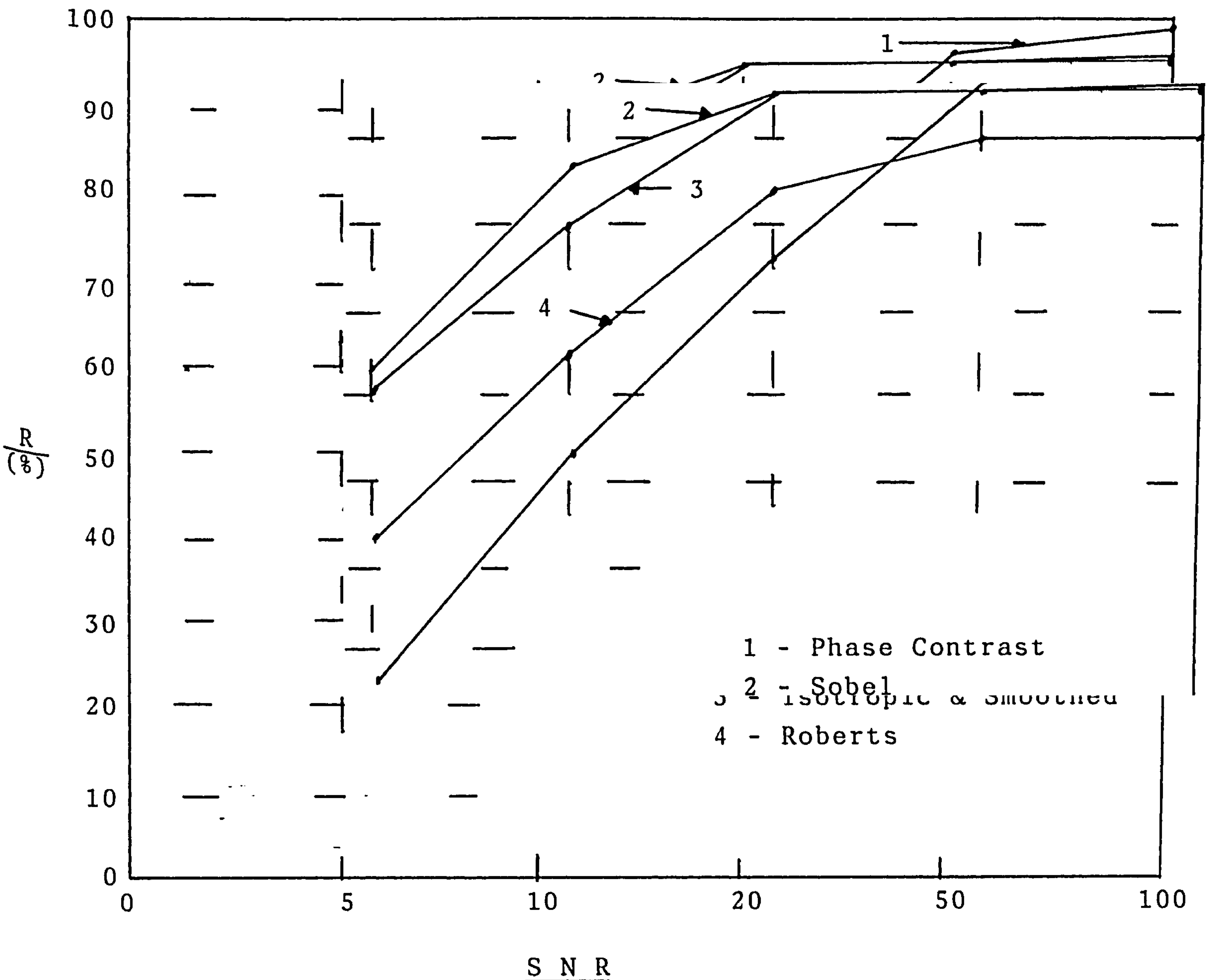


Fig. 4-2 - Edge Location Figure of Merit
As a Function of SNR. W = 1, h = 50.

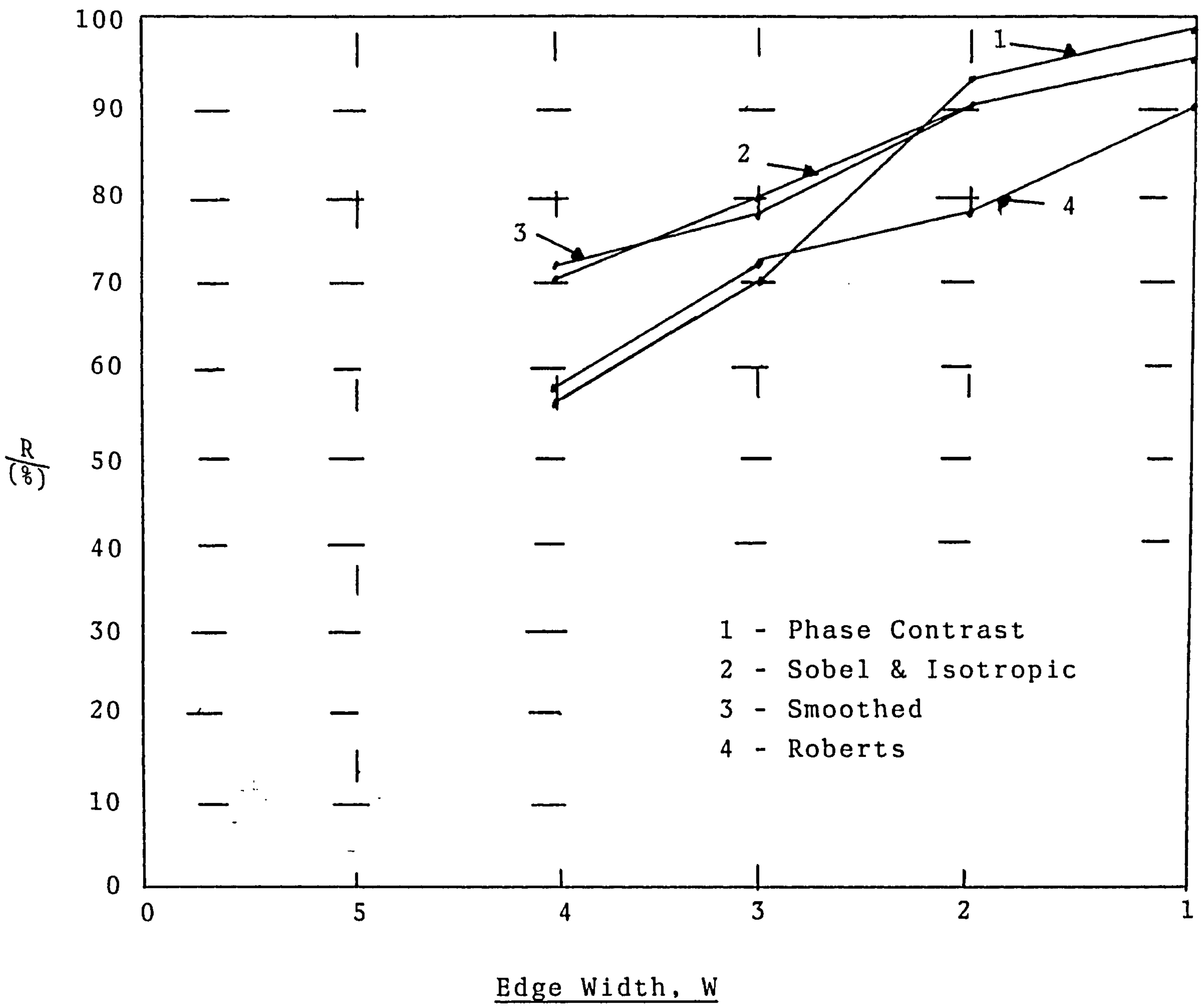


Fig. 4-3 - Edge Location Figure of Merit
As a Function of Edge Width. SNR = 100, h = 50

SNR		Phase Contrast	Sobel	Isotropic	Smoothed	Roberts
100	R	98.4	95.27	95.0	95.0	90.0
	I_A	62	74	124	124	63
50	R	95.48	95.12	95.0	94.9	90.0
	I_A	63	78	124	120	63
20	R	75.48	95.0	94.88	95.00	84.17
	I_A	66	88	90	80	62
10	R	52.98	85.7	80.29	80.53	65.18
	I_A	66	63	81	64	68
5	R	25.76	62.74	60.65	60.24	42.29
	I_A	70	61	65	63	66

Table 4-1 - Edge Detection Comparison Data
$h = 50$, $W = 1$, $I_I = 62$.

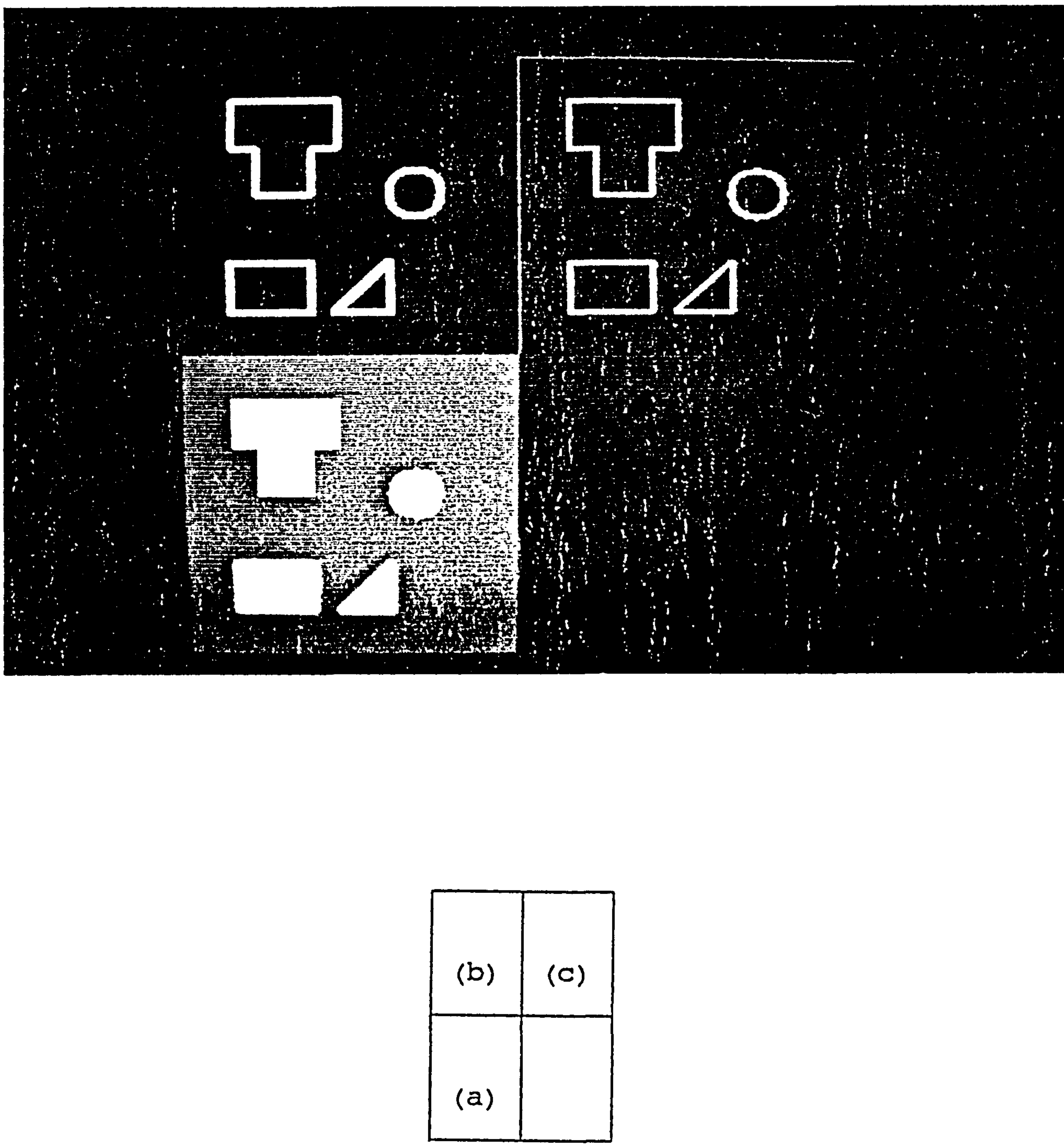


Fig. 4-4 - (a) Original Image
(b) Image Processed by Sobel Operator
(c) Image Processed by PCF with $\omega_c = 1.4$

CHAPTER 5

DISCUSSIONS, EXTENSIONS AND CONCLUSIONS

5.1. Introduction

The previous chapters have presented the PCF technique, a design method for obtaining the required filters for this technique and a comparison between the PCF and various gradient techniques. Conclusions and extensions of the PCF are presented.

5.2. Extensions

Section 3.4 presented the filter design technique's inability to obtain a unity magnitude response for filters of order of 2 or larger. An additional approach to this problem which may be explored is to use the function in equation (5.2-1) as the starting point for the minimization procedure.

It can easily be proven that this function provides a unity magnitude response over the entire frequency plane.

$$H(z_1, z_2) = \frac{\sum_{i=0}^{N} \sum_{j=0}^{N} a_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=0}^{N} \sum_{j=0}^{N} a_{N-i,N-j} z_1^{-i} z_2^{-j}} \qquad (5.2\text{-}1)$$

Future work could also explore the possibilities of obtaining the phase response, of which the cross-section is shown, in Fig. 5-1 for normalized frequency.

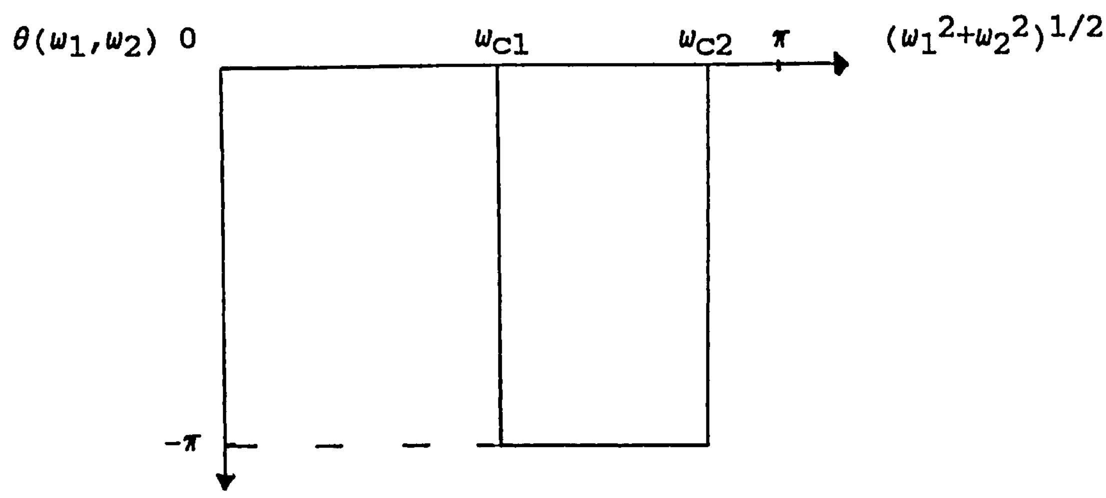


Fig. 5-1 Cross-section of Alternate Ideal Phase Response

With a phase response as shown above, unwanted high frequency noise would be eliminated when incorporated in the PCF technique.

5.3. Conclusions

A new method has been developed for the edge enhancement of digital images. This method offers a flexibility through its choice of ω_c and filter order which is unequaled by gradient techniques (beyond the selection of mask size).

Filters with a square symmetric phase response, when used in the PCF technique, instead of emphasing actually de-emphasize edges which have a slope of -1 relative to the x and y axes of the image. This type of response, although not suited for edge enhancement, might find a useful application in an area such as pattern recognition. This is deemed worthy of further study.

A fundamental point is the question of the ideal phase response necessary for the PCF technique. This ideal phase response although not attainable in the discrete time domain can easily be implemented in the frequency domain. Results have shown that when this ideal response is applied to an image through the PCF technique, the resulting image suffers from ringing or "Gibbs phenomena". Therefore a filter with a smoother response might be more useful in practice than that depicted by the ideal phase response in this thesis.

The data obtained in the comparison of the various techniques for figure of merit versus signal to noise ratio and edge width agrees with results obtained when applying the various techniques to images. Of the gradient techniques the Sobel operator has the best performance and the Roberts the poorest.

The PCF technique is capable of enhancing small changes in pixel values whereas the gradient techniques examined have an averaging effect so small variations in pixel values are not as pronounced. This sensitivity however, creates a noisy image when the PCF is applied to images with a low SNR. Practically, images with a low SNR would require the stochastic gradient methods mentioned in section 1.3.

The PCF technique has been shown to be effective for edge enhancement. The implementation of the PCF technique is straightforward and execution time, for filter order 1, is roughly equal that of a 3 x 3 gradient technique.

For conciseness all the data obtained during the course of the study has not been presented. However, the data is available on floppy disk through the author. Results are available of the following: (a) the PCF technique applied in the frequency domain through the use of the FFT and IFFT, images showing the performance of the PCF technique encorporating a square symmetric phase response, results of the use of higher order filters and results showing the application of the various gradient techniques compared in this thesis.

APPENDIX

```
c
c
c       this program determines the filter coefficients by solving a
c       matrix determined by the minimization of the error between the
        ideal and general impulse responses for the 1 dimensional case.
        real a(20,20),h(129),x(20),aa(20),bb(20),y(129)
c
c
c       the 1-d impulse response is read from file for003.dat

        print*, ' enter the order of the filter'
        read*,n
        print*, ' enter the gain'
        read*,gain
        numsamp=129
c
c
c       read the impulse response

        open(4,file='for003.dat',status='old')
        do 20 i=1,numsamp
        read(4,*)h(i)
20      continue
        close(4)
        do 67 i=1,20
        aa(i)=0.0
67      bb(i)=0.0
c
c
c       set the matrix coeff. equal to zero

        do 100 i=1,2*n+1
        do 100 j=1,2*n+2
100     a(i,j)=0
c
c
c       set upper left matrix diagonal = to 1

        do 1 i=1,n+1
        do 1 j=1,n+1
        if(i.eq.j)a(i,j)=1
1       continue
c
c
c       determine lower left coefficients.

        k=2
        do 2 i=n+2,2*n+1
        do 3 j=k,n+1
        a(i,j)=-h(j-k+1)
3       continue
        k=k+1
2       continue
c
c
c       determine upper right matrix coeff.

        k=1
        do 4 i=2,n+1
        do 5 j=1,k
        a(i,j+n+1)=-h(i-j)
5       continue
        k=k+1
4       continue
c
c
c       determine far upper right column
```

```
      do 6 i=1,n+1
6     a(i,2*n+2)=h(i)
c
c     determine lower right matrix
c
      k=0
      do 7 i=n+2,2*n+1
      k=k+1
      l=0
      do 8 j=n+2,2*n+1
      l=l+1
      sum=0
      do 9 nn=1,numsamp
      ll=nn-l
      kk=nn-k
      if(ll.lt.1.or.kk.lt.1)goto 9
      sum=sum+h(nn-l)*h(nn-k)
9     continue
      a(i,j)=sum
8     continue
7     continue
c
c     determine lower far right column.
c
      k=2
      j=2*n+2
      do 10 i=n+2,2*n+1
      sum=0
      do 11 nn=k,numsamp
      sum=sum+h(nn)*h(nn-k+1)
11    continue
      k=k+1
      a(i,j)=-sum
10    continue
      m=2*n+2
      n1=n*2+1
c
c     the matrix is solved using gauss gorden with partial pivoting
c
      call gauss(a,n1,m,x)
      do 62 i=1,n+1
      aa(i)=x(i)*gain
      write(9,*)aa(i)
      j=i-1
      print 33,j,aa(i)
33    format(' ','a(',i2,')',5x,'equal',f10.6)
62    continue
      do 63 i=n+2,2*n+1
      bb(i-n-1)=x(i)
      write(9,*)x(i)
      j=i-n-1
      print 34,j,x(i)
34    format(' ','b(',i2,')',5x,'equal',f10.6)
63    continue
c
c
c
c     generate impulse response of the designed filter and store it in
c     file for004.dat;* .  note this impulse response is designed for
c     up to order 5 filter only.
```

```
c
      y(1)=aa(1)
      k=1
      do 75 i=2,n+1
      l=i-1
      sum=0.0
      do 76 j=1,k
      sum=sum+bb(j)*y(l)
      l=l-1
76    continue
      y(i)=aa(i)-sum
      k=k+1
75    continue
      do 71 i=n+2,NUMSAMP
      y(i)=-bb(1)*y(i-1)-bb(2)*y(i-2)-bb(3)*y(i-3)-bb(4)*y(i-4)-bb(5)*y(i-5)
71    continue
      do 72 i=1,NUMSAMP
72    write(4,*)y(i),h(i)
99    continue
      stop
      end
c
c
c     8888888888888888888888888888888888888888888888888888888
c
      subroutine gauss(a,n1,m,x)
      real a(20,20),x(20)
c
c
c     m=n+1 and a is the augmented matrix
c     solution is given in x
c
      n=n1
      do 10 j=1,n
      big=abs(a(j,j))
      l=j
      do 20 k=j+1,n
      if(big.lt.abs(a(k,j)))then
      big=abs(a(k,j))
      l=k
      endif
20    continue
      if(big.lt.1e-7)then
      print*, ' no unique solution'
      return
      endif
      do 30 k=1,n+1
      temp=a(j,k)
30    a(j,k)=a(l,k)
      a(l,k)=temp
40    do 40 k=j+1,n+1
      a(j,k)=a(j,k)/a(j,j)
      a(j,j)=1.0
      do 10 i=1,n
      if(i.eq.j)goto 10
50    do 50 k=j+1,n+1
10    a(i,k)=a(i,k)-a(i,j)*a(j,k)
      a(i,j)=0.0
60    do 60 j=1,n
      x(j)=a(j,n+1)
      return
      end
```

```
c
c       this program determines the filter coefficients by solving a
c       matrix determined from the minimization of the error between
c       the ideal and general case impulse responses.
        real a(60,60),h(129,129),x(60),aa(60),bb(60),y(129,129)
c
c       the ideal impulse response is read from a file called for021.dat;*
c       this impulse response must be supplied.
c
        print*, ' enter the order of the filter'
        read*,n
        print*,' enter the desired gain'
        read*,gain
        n1=n+1
        n1s=n1*n1
c
c       read the impulse response
c
        open(4,file='for021.dat',status='old')
        read(4,*)((h(i,j),j=1,129),i=1,129)
        close(4)
c
c       set the matrix coeff. equal to zero
c
        do 100 i=1,60
        do 100 j=1,60
100     a(i,j)=0
c
c       set upper left matrix diagonal = to 1
c
        do 1 i=1,n1s
        do 1 j=1,n1s
1       if(i.eq.j)a(i,j)=1
        continue
c
c       determine middle upper right coefficients.
c
        k=1
        l=1
        do 2 i=1,n1s
        if(l.eq.1)goto 3
        do 4 j=1,l-1
4       a(i,j+n1s)=-h(k,l-j)
3       continue
        l=l+1
        if(l.gt.n1)then
        l=1
        k=k+1
        endif
2       continue
c
c       determine upper right matrix coeff.
c
        do 5 num=1,n
        l=1
        k=1
        do 6 i=num*n1+1,n1s
        do 7 j=1,l
        l1=n1s+n+(num-1)*n1+j
        a(i,l1)=-h(k,l-j+1)
```

```
7       continue
        l=l+1
        if(l.gt.n1)then
        l=1
        k=k+1
        endif
6       continue
5       continue
c
c       determine lower left matrix coeff.
c
        do 8 i=n1s+1,2*n1s-1
        do 8 j=1,n1s
8       a(i,j)=a(j,i)
c
c       determine middle lower right matrix coeff.
c
        do 11 num=1,n1
        if(num.eq.1)nn1=n
        if(num.gt.1)nn1=n+1
        do 12 i=1,nn1
        do 13 j=1,n
c
c       sum is determined
c
        sum=0
        do 14 nn=1,129
        do 15 mm=1,129
        jj1=mm-j
        jj2=nn-num+1
        jj3=mm-i
        if(num.gt.1)jj3=jj3+1
        if(jj1.lt.1.or.jj2.lt.1)goto 15
        if(jj3.lt.1)goto 15
        sum=sum+h(nn,jj1)*h(jj2,jj3)
15      continue
14      continue
        if(num.eq.1)ii=n1s+i
        if(num.gt.1)ii=n1s+n+(num-2)*(n+1)+i
        a(ii,j+n1s)=sum
13      continue
12      continue
11      continue
c
c       determine lower right matrix coeff.
c
        do 22 num1=1,n
        do 22 num=1,n1
        if(num.eq.1)nn1=n
        if(num.gt.1)nn1=n1
        do 17 i=1,nn1
        do 18 j=1,n1
c
c       sum is determined
c
        sum=0
        do 19 nn=1,129
        do 19 mm=1,129
        jj1=nn-num1
        jj2=mm-j+1
```

```
      jj3=nn-num+1
      jj4=mm-i
      if(num.gt.1)jj4=jj4+1
      if(jj1.lt.1.or.jj2.lt.1)goto 19
      if(jj3.lt.1.or.jj4.lt.1)goto 19
      sum =sum +h(jj1,jj2)*h(jj3,jj4)
19    continue
      if(num.eq.1)ii=n1s+i
      if(num.gt.1)ii=n1s+n+(num-2)*(n+1)+i
      jj=n1s+n+(num1-1)*(n+1)+j
      a(ii,jj)=sum
18    continue
17    continue
22    continue
c
c     determine far upper right column
c
      jj=2*n1s
      do 30 i=1,n1
      do 30 j=1,n1
      ii=(i-1)*n1+j
      a(ii,jj)=h(i,j)
30    continue
c
c     determine far lower right column
c
      do 31 i=1,n1
      if(i.eq.1)then
      nan=0
      nn1=n
      else
      nan=1
      nn1=n1
      endif
      do 31 j=1,nn1
      if(i.eq.1)ii=n1s+j
      if(i.gt.1)ii=n1s+n+(i-2)*n1+j
      sum=0
      do 32 nn=1,129
      do 32 mm=1,129
      jj1=nn-i+1
      jj2=mm-j+nan
      if(jj1.lt.1.or.jj2.lt.1)goto 32
      sum=sum+h(nn,mm)*h(jj1,jj2)
32    continue
      a(ii,jj)=-sum
31    continue
c
c

      nn1=2*n1s-1
      m=nn1+1
c
c
c     the matrix which is to be solved is written in file for016.dat;*
c
c     write(16,*)((a(i,j),j=1,2*n1s),i=1,2*n1s-1)
c
c     the matrix is solved using gauss gorden with partial pivoting

      call gauss(a,nn1,m,x)
      k=0
```

```
      do 62 i=1,n1
      do 62 j=1,n1
      k=k+1
      ii=i-1
      jj=j-1
      aa(k)=x(k)*gain
      write(19,*)aa(k)
83    print 83,ii,jj,aa(k)
62    format(' ','a(',i2,i2,')',5x,'equal',f10.6)
      continue
      do 63 i=1,n1
      if(i.eq.1)then
      nn1=n
      else
      nn1=n1
      endif
      do 63 j=1,nn1
      k=k+1
      ii=i-1
      jj=j-1
      if(i.eq.1)jj=j
      write(19,*)x(k)
84    print 84,ii,jj,x(k)
63    format(' ','b(',i2,i2,')',5x,'aqual',f10.6)
      continue
99    goto 99
      stop
c     end
c
c     888888888888888888888888888888888888888888888888888888888

      subroutine gauss(a,nn1,m,x)
c     real a(60,60),x(60)
c
c     m=n+1 and a is the augmented matrix
c     solution is given in x

      n=nn1
      do 10 j=1,n
      big=abs(a(j,j))
      l=j
      do 20 k=j+1,n
      if(big.lt.abs(a(k,j)))then
      big=abs(a(k,j))
      l=k
20    endif
      continue
      if(big.lt.1e-7)then
      print*, ' no unique solution'
      return
      endif
      do 30 k=1,n+1
      temp=a(j,k)
30    a(j,k)=a(l,k)
      a(l,k)=temp
40    do 40 k=j+1,n+1
      a(j,k)=a(j,k)/a(j,j)
      a(j,j)=1.0
      do 10 i=1,n
      if(i.eq.j)goto 10
```

```
      do 50 k=j+1,n+1
50    a(i,k)=a(i,k)-a(i,j)*a(j,k)
10    a(i,j)=0.0
      do 60 j=1,n
60    x(j)=a(j,n+1)
      return
      end
```

```
c
c     this program generates ideal phase and magnitude responses in the
c       frequency domain and generates the ideal impulse response for the
c       1 dimensional case.
c
c
      REAL PI,W(256),THETA(256),MAG(256),X(256),Y(256)
      INTEGER N,N1,N2
      print*, ' enter the cut off frequency in radians'
      read*,wc
      N=128
      N2=256
      PI=4.0*ATAN(1.0)
      N1=(WC/PI)*(N+1)
      print*,n1
      DELTA=PI/(n+1)
c
c
c     the phase response is 0 form w=0 to wc and it is equal to -pi
C     form w = wc to pi
C

      W(1)=delta
      DO 1 I=1,N1
      THETA(I)=0.0
1     W(I)=W(I-1)+DELTA
C
      DO 2 I=N1+1,N+1
      THETA(I)=-PI
2     W(I)=W(I-1)+DELTA
C
c
C     determine real and imaginary part of frequency response

      DO 3 I=1,N+1
      X(I)=COS(THETA(I))
3     Y(I)=SIN(THETA(I))
c
c
c     generate odd and evan symetry of function

      KOUNT=N
      DO 56 I=N+2,N2
      X(I)=X(KOUNT)
      Y(I)=-Y(KOUNT)
      KOUNT=KOUNT-1
56    CONTINUE
C
C
C

      DO 57 J=N+1,N2
57    W(J)=W(J-1)+DELTA
c
c
c     determine idft of the frequency response
c     the real part is stored in a file called for003.dat;*

      M=8
      ISIGN=-1
      N=n2
      CALL FFT(X,Y,N,M,ISIGN)
      WRITE(9,*)'  IDFT IE h(n) '
      DO 94 I=1,N2
      write(9,*)x(i),y(i),i
```

```
94      continue
        n=128
c       do 77 i=1,n
c       x(i)=x(i)*(-1)**i
c77     l=l+2
        DO 51 I=1,n2
51      WRITE(3,*)X(I)
        STOP
        END
C
c       if isign =-1 the idft is determined
c       if isign =+1 the dft is determined
C
        SUBROUTINE FFT(X,Y,N,M,ISIGN)
        REAL X(256),Y(256)
        N2=N
        DO 10 K=1,M
        N1=N2
        N2=N2/2
        E=6.283185307/N1
        A=0.0
        DO 20 J=1,N2
        C=COS(A)
        S=SIN(A)
        IF(ISIGN.EQ.-1)S=-S
        A=J*E
        DO 30 I=J,N,N1
        L=I+N2
        XT=X(I)-X(L)
        X(I)=X(I)+X(L)
        YT=Y(I)-Y(L)
        Y(I)=Y(I)+Y(L)
        X(L)=C*XT+S*YT
        Y(L)=C*YT-S*XT
30      CONTINUE
20      CONTINUE
10      CONTINUE
C
C
C       THIS IS THE BIT REVERSAL PROGRAM
C
C
100     J=1
        N1=N-1
        DO 104 I=1,N1
        IF(I.GE.J)GOTO 101
        XT=X(J)
        X(J)=X(I)
        X(I)=XT
        XT=Y(J)
        Y(J)=Y(I)
        Y(I)=XT
101     K=N/2
102     IF(K.GE.J)GOTO 103
        J=J-K
        K=K/2
        GOTO 102
103     J=J+K
104     CONTINUE
        IF(ISIGN.EQ.-1)THEN
```

```
      DO 33 I=1,N
      X(I)=X(I)/N
33    Y(I)=Y(I)/N
      ENDIF
      RETURN
      END
```

```
c     this is the program that generates an ideal phase and magnitude
c       response in the frequency domain and determines the ideal imulse
c       response.
c       The phase response is 0 from w equal 0 rad. to w equal wc
c       and -pi from w equal wc to pi.
c
c

        REAL PI,THETA(256,256),TEMR(256,256),TEMI(256,256),XXI(256,256)
        real temp(35,35)
        INTEGER N,N1,N2
        print*, ' enter the lower cut off frequency in radians'
        read*,wc
        N=128
        N2=256
        PI=4.0*ATAN(1.0)
        xinc=pi/129
        N1=(WC/PI)*(N+1)
C       print*,n1
c
C       generate zero phase from  w =0 to wc

        DO 1 I=1,N+1
        DO 1 J=1,N+1
        theta(i,j)=0
        PP1=I
        PP2=J
        XX1=SQRT(PP1**2+PP2**2)
1       IF(XX1.GT.N1)THETA(I,J)=-PI
c       CONTINUE
c
c       This small section writes in file for008.dat;* the ideal
c       phase response you have chosen so you may display it using
c       the hid.for program imediately after executing this prorgam

        nsamp=35
        xinc=pi/35
        n21=(wc/pi)*nsamp
        do 96 i=1,nsamp
        do 96 j=1,nsamp
        temp(i,j)=0
        p1=i
        p2=j
        xx1=sqrt(p1**2+p2**2)
96      if(xx1.ge.n21)temp(i,j)=-pi
        continue
        write(8,*)nsamp,nsamp,xinc,xinc
        do 97 i=1,nsamp
        do 97 j=1,nsamp
97      write(8,*)temp(i,j)
C       continue
C
C       determine real and imaginary parts of filters frequency response

        DO 3 I=1,N+1
        DO 3 J=1,N+1
        TEMR(I,J)=COS(THETA(I,J))
3       TEMI(I,J)=SIN(THETA(I,J))
c       continue
c
        generate odd and evan symetry of function
```

```
C
C
C     GENERATE QUADRANT 4

      k=n+1
      do 4 i=n+1,n2
      l=n+1
      do 5 j=n+1,n2
      TEMR(i,j)=TEMR(k,l)
      TEMI(I,J)=-TEMI(K,L)
5     L=L-1
4     K=K-1
C
C
C     GENERATE QUADRANT 1

      DO 6 I=1,N
      L=N
      DO 6 J=N+2,N2
      TEMR(I,J)=TEMR(I,L)
      TEMI(I,J)=TEMI(I,L)
6     L=L-1
C
C
C     GENERATE QUADRANT 3

      K=N
      DO 7 I=N+2,N2
      L=N2
      DO 8 J=2,N
      TEMR(I,J)=TEMR(K,L)
      TEMI(I,J)=-TEMI(K,L)
8     L=L-1
7     K=K-1
      L=N
      DO 9 I=N+2,N2
      TEMR(I,1)=TEMR(L,1)
      TEMI(I,1)=-TEMI(L,1)
9     L=L-1
C
C
C    determine IDFT of filter response

      M=8
      ISIGN=-1
      N=n2
      CALL DDFFT(XXI,N,M,ISIGN,TEMR,TEMI)
c     n=128
c
c     filters impulse response is in array temr(i,j)
c
c
c     store the impulse response in file called "FOR021.DAT;*"

99    WRITE(21,*)((TEMR(I,J),J=1,N+1),I=1,N+1)
      STOP
C     END
C
C     THIS ROUTINE DETERMINES THE 2-D DFT OF A REAL ARRAY XX1
C     IF ISIGN=1 WHERE THE REAL PART AND IMAGINARY PART ARE
C     STORED IN ARRAYS TEMR AND TEMI RESPECTIVELY.
C
C     IF ISIGN=-1 THE IDFT IS DETERMINED AND THE REAL AND IMAGINARY
      PARTS ARE STORED BACK INTO THE ARRAYS TEMR AND TEMI WHICH
```

```
C
      CONTAIN THE ORIGINAL REAL AND IMAGINARY PARTS RESPECTIVELY.
C
      SUBROUTINE DDFFT(XXI,N,M,ISIGN,TEMR,TEMI)
      REAL XXI(256,256),TEMR(256,256),TEMI(256,256),X(256),Y(256)
      INTEGER N,M,ISIGN
      IF(ISIGN.EQ.-1)GOTO 47
C
C     THE DO 10 LOOP DETERMINES THE ROW DFT
C
      DO 10 I=1,N
      DO 20 J=1,N
      X(J)=XXI(I,J)
20    Y(J)=0.0
      CALL FFT(X,Y,N,M,ISIGN)
      DO 30 J=1,N
      TEMR(I,J)=X(J)
30    TEMI(I,J)=Y(J)
10    CONTINUE
C
C     THE DO 40 LOOP DETERMINES THE COLUMN DFT
C
      DO 40 J=1,N
      DO 50 I=1,N
      X(I)=TEMR(I,J)
50    Y(I)=TEMI(I,J)
      CALL FFT(X,Y,N,M,ISIGN)
      DO 60 I=1,N
      TEMR(I,J)=X(I)
60    TEMI(I,J)=Y(I)
40    CONTINUE
      GOTO 48
C
C     THE DO 41 LOOP DETERMINES THE ROW DFT FOR THE IDFT
C
47    DO 41 I=1,N
      DO 42 J=1,N
      X(J)=TEMR(I,J)
42    Y(J)=TEMI(I,J)
      CALL FFT(X,Y,N,M,ISIGN)
      DO 43 J=1,N
      TEMR(I,J)=X(J)
43    TEMI(I,J)=Y(J)
41    CONTINUE
C
C     THE DO 44 LOOP DETERMINES THE COLUMN DFT FOR THE IDFT.
C
      DO 44 J=1,N
      DO 45 I=1,N
      X(I)=TEMR(I,J)
45    Y(I)=TEMI(I,J)
      CALL FFT(X,Y,N,M,ISIGN)
      DO 46 I=1,N
      TEMR(I,J)=X(I)
46    TEMI(I,J)=Y(I)
44    CONTINUE
48    RETURN
      END
C
C     THIS ROUTINE DETERMINES THE DFT AND IDFT OF A COMPLEX ARRAY
C     THE REAL PART IS STORED IN X AND THE IMAGINARY PART IN Y
```

```
C
C     FOR ISIGN =1 AND -1 THE DFT AND IDFT IS DETERMINED RESP.
C     M IS LOG(BASE 2) OF N.- EX. FOR N=128, M=7
      SUBROUTINE FFT(X,Y,N,M,ISIGN)
      REAL X(256),Y(256)
      N2=N
      DO 10 K=1,M
      N1=N2
      N2=N2/2
      E=6.283185307/N1
      A=0.0
      DO 20 J=1,N2
      C=COS(A)
      S=SIN(A)
      IF(ISIGN.EQ.-1)S=-S
      A=J*E
      DO 30 I=J,N,N1
      L=I+N2
      XT=X(I)-X(L)
      X(I)=X(I)+X(L)
      YT=Y(I)-Y(L)
      Y(I)=Y(I)+Y(L)
      X(L)=C*XT+S*YT
      Y(L)=C*YT-S*XT
30    CONTINUE
20    CONTINUE
10    CONTINUE
C
C
C
C     THIS IS THE BIT REVERSAL PROGRAM
C
100
      J=1
      N1=N-1
      DO 104 I=1,N1
      IF(I.GE.J)GOTO 101
      XT=X(J)
      X(J)=X(I)
      X(I)=XT
      XT=Y(J)
      Y(J)=Y(I)
      Y(I)=XT
101   K=N/2
102   IF(K.GE.J)GOTO 103
      J=J-K
      K=K/2
      GOTO 102
103   J=J+K
104   CONTINUE
      IF(ISIGN.EQ.-1)THEN
      DO 33 I=1,N
33    X(I)=X(I)/N
      Y(I)=Y(I)/N
      ENDIF
      RETURN
      END
```

```
c
c      this program determines the discrete impulse response for a filter
c      of order n. the filter coefficients are read from a file called
c      'for019.dat;*'.
c
       real h(129,129),a(3,3),b(3,3)
       print*,' enter filter order'
       read*,n
       n1=n+1
       open(2,file='for019.dat',status='old')
       do 30 i=1,n1
       do 30j=1,n1
       read(2,*)a(i,j)
       print*,a(i,j)
30     continue
       do 31 i=1,n1
       do 31 j=1,n1
       if(i.eq.1.and.j.eq.1)goto 31
       read(2,*)b(i,j)
       print*,b(i,j)
31     continue
       do 10 nn=1,129
       do 10 mm=1,129
       sum=0.0
       do 12 i=1,n1
       do 12 j=1,n1
       jj1=nn-i+1
       jj2=mm-j+1
       if(jj1.lt.1.or.jj2.lt.1)goto 12
       sum=sum+b(i,j)*h(jj1,jj2)
12     continue
       h(nn,mm)=-sum
       if(nn.le.n1.and.mm.le.n1)h(nn,mm)=h(nn,mm)+a(nn,mm)
10     continue
       write(21,*)((h(i,j),j=1,129),i=1,129)
       stop
       end
```

```
C
C      THIS PROGRAM ADDS WHITE GAUSSIAN NOISE WITH STANDARD DEVIATION 'sd
C      TO AN IMAGE 'img '.
C
       real sd,r1,r2,x(16384),twopi
       integer img(128,128)
       double precision xix
C
       write(*,*)' enter the desired standard deviation'
       read(*,*)sd
C
       twopi=8.0*atan(1.0)
       n1=128*128
C
C      the seed for the generator is 5.0d0
C
       xix=5.0d0/2147483647.0d0
       do 10 i=1,8192
       call rand(xix)
       r1=sngl(xix)
       call rand(xix)
       r2=sngl(xix)
       x(i)=sd*sqrt(-2.0*alog(r1))*cos(twopi*r2)
       x(i+8192)=sd*sqrt(-2.0*alog(r1))*sin(twopi*r2)
10     continue
       kount=0
       do 20 i=1,128
       do 20 j=1,128
       kount=kount+1
20     img(i,j)=x(kount)
C
C      store the noisy image.
C
       open(3,file='temp.img',form='unformatted',status='new')
       write(3)((img(i,j),j=1,128),i=1,128)
       close(3)
       stop
       end
c
c*****************************************************
c
       subroutine rand(xix)
       double precision xix
c
c
c      m-modulus is 2**31-1 = 2,147,483,647
c      a-multiplicator- is 7**5 = 16,807
c      ix-seed- equals 5 for the first call and xix for subsequent calls.
c
c
c      the form of the generator is z(i)=z(i-1)*a
c
       xix=xix*16807.0d0
       xix=dmod(xix,1.0d0)
       return
       end
```

```
C
C     THIS PROGRAM PERFORMS THE 2-D FILTERING OF LARGE SIZE IMAGES GIVEN
C     COEFFICIENTS OF THE 2-D DIGITAL RECURSIVE FILTER.
C
C     WRITTEN BY DR. M. SID-AHMED, ELECTRICAL ENGINEERING DEPARTMENT,
C     UNIVERSITY OF WINDSOR,WINDSOR ONT.
C
      REAL*4 a(10,10),b(10,10)
      REAL*4 IX(10,256),IY(10,256),IYT,MIN,MAX
      CHARACTER IARRAY(256)
      CHARACTER*13 FILN
      MIN=0
      MAX=255
      WRITE(*,51)
51    FORMAT(' ENTER SIZE OF IMAGE. IE 64 X64,128 X128,256 X256 ETC.'\)
      READ(*,*)NSIZ
      WRITE(*,3)
3     FORMAT(' INPUT FILE NAME---> '\)
      READ(*,'(A13)')FILN
      OPEN(1,FILE=FILN,FORM='BINARY',STATUS='OLD')
      WRITE(*,4)
4     FORMAT(' OUTPUT FILE NAME---> '\)
      READ(*,'(A13)')FILN
      OPEN(2,FILE=FILN,FORM='BINARY',STATUS='NEW')
      WRITE(*,6)
6     FORMAT(' FILTER COEFFICIENTS FILE NAME--->',\)
      READ(*,'(A13)')FILN
      OPEN(7,FILE=FILN,FORM='UNFORMATTED',STATUS='OLD')
      WRITE(*,*) 'INPUT ORDER OF FILTER'
      READ(*,*) N
      WRITE(*,*)N
      DO 70 I=1,N+1
      DO 70 J=1,N+1
      READ(7)A(I,J)
70    CONTINUE
      DO 71 I=1,N+1
      DO 71 J=1,N+1
      READ(7)B(I,J)
71    CONTINUE
      CLOSE(7)
C
C
      WRITE(*,*)N
      DO 17 I=1,N+1
17    WRITE(*,*)(A(I,J),J=1,N+1)
      DO 18 I=1,N+1
18    WRITE(*,*)(B(I,J),J=1,N+1)
C
C
      DO 8 I=1,N+1
      DO 8 J=1,NSIZ
      IX(I,J)=0
8     IY(I,J)=0
      K=0
C
C
      DO 9 L=1,NSIZ
      READ(1)(IARRAY(J),J=1,NSIZ)
      DO 10 J=1,NSIZ
10    IX(1,J)=ICHAR(IARRAY(J))
C
C
```

```
C
      DO 23 M=1,NSIZ
      SUM=0.0
      DO 11 I=1,N+1
      DO 11 J=1,N+1
      IF((M+1-J).GT.0)THEN
      SUM=SUM+A(I,J)*IX(I,M+1-J)
      IF((I.EQ.1).AND.(J.EQ.1))GOTO 11
      SUM=SUM-B(I,J)*IY(I,M+1-J)
      ENDIF
11    CONTINUE
23    IY(1,M)=SUM
      K=K+1
      IF(K.EQ.(N+1))THEN
      DO 21 I=N+1,1,-1
      DO 14 J=1,NSIZ
      IYT=ABS(IY(I,J)-IX(I,J))
C      IYT=IY(I,J)
      IF(IYT.LT.MIN)IYT=0
      IF(IYT.GT.MAX)then
      maxxx=iyt
      IYT=MAX
      endif
      NN=IYT
14    IARRAY(J)=CHAR(NN)
21    WRITE(2)(IARRAY(J),J=1,NSIZ)
      K=0
      ENDIF
      IF((L.EQ.NSIZ).AND.(K.NE.0))THEN
      DO 22 I=K,1,-1
      DO 15 J=1,NSIZ
      IYT=ABS(IY(I,J)-IX(I,J))
C      IYT=IY(I,J)
      IF(IYT.LT.MIN)IYT=0
      IF(IYT.GT.MAX)then
      maxxx=iyt
      IYT=MAX
      endif
      NN=IYT
15    IARRAY(J)=CHAR(NN)
22    WRITE(2)(IARRAY(J),J=1,NSIZ)
      ENDIF
      DO 12 I=1,N
      DO 12 J=1,NSIZ
      IY(N+2-I,J)=IY(N+1-I,J)
12    IX(N+2-I,J)=IX(N+1-I,J)
9     CONTINUE
      CLOSE(1)
      CLOSE(2)
      write(*,*)maxxx
      STOP
      END
```

REFERENCES

[1] J.J. Soltis, private communication.

[2] W.K. Pratt, "Digital Image Processing", Wiley Interscience, New York, 1978.

[3] A. Rosenfeld and A.C. Kak, "Digital Picture Processing", Vol.1, Academic Press, Orlando, FL., 1982.

[4] R.C. Gonzalez and P. Wintz, "Digital Image Processing", Addison-Wesley, Reading, Mass., 1983.

[5] J.K. Aggarwal, R.O. Duda and A. Rosenfeld (Eds), "Computer Methods in Image Analysis", IEEE Press, 1977.

[6] E. Argyle, "Techniques For Edge Detection", Proc. of the IEEE, Vol.59, No.2, pp.285-287, Feb. 1971.

[7] A. Rosenfeld and M. Thurston, "Edge and Curve Detection For Visual Scene Analysis", in [5].

[8] G.S. Robinson, "Edge Detection by Compass Gradient Masks", Comp. Graphics and Image Proc., Vol.6, pp.492-501, 1977.

[9] D. Mavv and E.C. Hildreth, "Theory of Edge Detection", Proc. R. Society London, B, 270, pp.187-217, 1980.

[10] M. Heuckle, "An Operator Which Locates Edges in Digitized Pictures", J. ACM, Vol.18, No.1, pp.113-125, Jan. 1976.

[11] A.K. Jain and S. Ranganath, "Image Restoration and Edge Extraction Based on 2-D Stochastic Models", Proc. ICASSP-82, Paris, May 1982.

[12] J.M.S. Prewitt, "Object Enhancement and Extraction", Picture Processing and Psychopictoris, B.S. Lipkin and A. Rosenfeld (Eds.), Academic Press, New York, 1970.

[13] I.D.G. MacLeod, "Comments on `Techniques for Edge Detection'", Proc. IEEE, 60, 3, March 1972.

[14] A. Rosenfeld, M. Thurston and Y. Lee, "Edge and Curve Detection: Further Experiments", IEEE Trans. Computers, C-21, 7, pp.677-715, July 1972.

[15] A. Rosenfeld, "A Nonlinear Edge Detection Technique", Proc. IEEE Letters, 58, 5, pp.814-816, May 1970.

[16] J.R. Fram and E.S. Deutsch, "On the Evaluation of Edge Detection Schemes and Their Comparison to Human Performance", IEEE Trans. Computers, C-24, 6, pp.616-628, June 1975.

[17] W.D. Stanley, "Electronic Communications Systems", Prentice-Hall, Reston, Virginia, 1982.

[18] G.R. Dougherty, R. Dougherty and W.D. Stanley, "Digital Signal Processing", Prentice-Hall, Reston, Virginia, 1984.

[19] A. Peled and B. Liu, "Digital Signal Processing Theory, Design and Implementation", John Wiley and Sons, New York, 1970.

[20] A.V. Oppenheim and R.W. Schafer, "Digital Signal Processing", Prentice-Hall, Englewood Cliffs, N.J., 1985.

[21] L.R. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing", Prentice-Hall, Englewood Cliffs, N.J., 1975.

[22] C.M. Rader and B.Gold, "Digital Processing of Signals", McGraw-Hill Book Co., New York, 1969.

[23] R.M. Mersereau and T.C. Speake, "A Unified Treatment of Cooley-Tukey Algorithms for the Evaluation of the Multidimensional DFT", IEEE Trans. ASSP-29, No.5, pp.1011-1018, Oct. 1981.

[24] R.M. Mersereau and D.E. Dudgeon, "The Representation of Two-Dimensional Sequences as One-Dimensional Sequences", IEEE Trans. ASSP, 22, No.5, pp.320-325, Oct. 1974.

[25] D.E. Dudgeon and R.M. Mersereau, "Multi Dimensional Digital Signal Processing", Prentice-Hall, Englewood Cliffs, N.J., 1984.

[26] B.T. O'Connor and T.S. Huang, "Stability of General Two-Dimensional Recursive Filters", IEEE Trans. ASSP, 26, No.6, pp.550-560, Dec. 1978.

[27] C.M. Rader and B. Gold, "Digital Filter Design Techniques in the Frequency Domain", Proc. IEE, 55, No.2, pp.149-171, Feb. 1967.

[28] J.F. Kaiser, "Digital Filters", Chapter 7 in System Analysis by Digital Computer, F.F. Kuo and J.F. Kaiser (Eds.), John Wiley and Sons, New York, 1966.

VITA AUCTORIS

March 20, 1960	Born in Sudbury, Ontario, Canada

Dates Attended

Sept. 1974 - May 1978	Sudbury Secondary School. Received high school diploma while enrolled in the advanced high school program.
Sept. 1978 - May 1981	Cambrian College of Applied Arts and Technology. Received diploma in Electronics Engineering Technology.
June 1982 - May 1984	Lakehead University. Received Bachelor of Applied Science degree in Electrical Engineering.
Sept. 1985 - present	University of Windsor. Enrolled in Master of Applied Science in the Electrical Engineering Department.