

1988

Focusing of images using spatial operators.

Mouloud M. Mameri
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Mameri, Mouloud M., "Focusing of images using spatial operators." (1988). *Electronic Theses and Dissertations*. Paper 2132.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

**FOCUSING OF IMAGES USING
SPATIAL OPERATORS**

By

MOULOU D M MAMERI

A thesis
submitted to the
Faculty of Graduate Studies and Research
through the Department of
Electrical Engineering in Partial Fulfillment
of the requirements for the Degree
of Master of Applied Science at
the University of Windsor

(c) Copyright MOULOU D M MAMERI, 1988.

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-43759-6

ABSTRACT.

A spatial domain procedure for restoring images blurred either by computer simulation (simulated blur), or by an image forming system (out-of-focus blur), by utilizing 2-D recursive (IIR) digital filters is presented.

Several cases are considered.

For the first case, two similar images are provided. 2-D IIR filter transfer functions are derived by the minimization of the total squared error (least-squares).

For the second case only the blurred image is available. An estimate of the spatial constant ' σ ' of the Gaussian point-spread function (PSF) is obtained.

Inverse filtering is used in the restoration process.

Restoration is achieved by using both global and local operators in the form of 2-D IIR filters.

Examples of restoration on simulated blur, as well as camera blurred images using the proposed algorithms are provided.

ACKNOWLEDGMENTS.

I would like to express my deep gratitude to Dr. M. A. Sid-Ahmed my supervisor for his guidance, support, and helpful remarks throughout the course of this work.

I am also most grateful to Dr. J. J. Soltis, Dr. W. C. Miller, and Dr. S. Bandyopadhyay for their helpful contributions and constructive suggestions.

I am very grateful to Mr. S. O. Belkasim for his great help and cooperation. I wish also to thank all my fellow graduate students and my colleagues, A. Namane, M. Tellache, and M. A. Elgabali. I would like to thank all my friends from the Electrical (Power), Civil, and Material Engineering departments. Special thanks to M. Hamdan from the Mathematics department, Nasser (Elcheikh) and Samir (Boukharou).

Last but not least, I extend my sincerest thanks to my beloved parents, grand-mother, brothers and brothers in law in Algeria.

TABLE OF CONTENTS.

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF ILLUSTRATIONS	viii
LIST OF TABLES	xii
 <u>Chapter I. INTRODUCTION.</u>	
1.1 OVERVIEW OF DIGITAL IMAGE PROCESSING	1
1.2 DIGITAL IMAGE RESTORATION	2
Degraded system	3
Different approaches	4
FOURIER TRANSFORM APPROACH	4
SPATIAL DOMAIN APPROACH	5
1.3 FOCUSING OF IMAGES USING SPATIAL OPERATORS	6
1.4 THESIS ORGANISATION	9
 <u>Chapter II DIGITAL IMAGE PROCESSING.</u>	
2.1 INTRODUCTION	10
2.2 DIGITAL IMAGE FUNDAMENTALS	11
2.3 IMAGE DIGITIZATION	12
2.4 IMAGE ENHANCEMENT AND RESTORATION	14
Enhancement approach	15

SPATIAL-DOMAIN METHODS	16
<i>Histogram modification technique</i>	16
HISTOGRAM EQUALIZATION	16
HISTOGRAM SPECIFICATION	17
- <i>Image smoothing</i>	20
NEIGHBORHOOD AVERAGING	20
SELECTIVE AVERAGING	21
MEDIAN FILTERING	22
AVERAGING OF MULTIPLE IMAGES	23
<i>Image sharpening</i>	24
SHARPENING BY DIFFERENTIATION	24
FREQUENCY DOMAIN METHODS	25
<i>Image smoothing by lowpass filtering</i>	25
IDEAL LOWPASS FILTER (ILPF)	26
<i>Image sharpening by highpass filtering</i>	27
IDEAL HIGHPASS FILTER (IHPF)	28
<i>Enhancement based on an image model</i> <i>homomorphic filtering</i>	29
CONCLUSION	32

Chapter III DEGRADATION PROCESS.

3.1 INTRODUCTION	33
3.2 DIFFERENT SOURCES OF DEGRADATION	34
Motion blur	34
Atmospheric turbulence	35
Out-of-focus	37

FOCUSING OF IMAGES USING BLURRED IMAGE ONLY	92
Algorithm	92
Application	93
Conclusion and comments	106
Chapter V SUMMARY AND CONCLUSIONS.	
5.1 SUMMARY OF THE WORK DONE	109
5.2 CONCLUSIONS	110
REFERENCES	111
APPENDIX (A)	118
Inverse filtering program using the FFT	118
APPENDIX (B)	125
Determination of the 2-D filter coefficients program	125
APPENDIX (C)	131
Filtering large images program	131
APPENDIX (D)	135
Thresholding images program	135
APPENDIX (E)	139
Calculation of 'a' program	139
APPENDIX (F)	144
Shanks method program	144
APPENDIX (G)	153
Filtering images program	153
APPENDIX (H)	157
A- IMAGE SEGMENTATION	157
1- Introduction	157
2- Point dependent techniques	158
GRAY-LEVEL THRESHOLDING	158

3.5a	Magnitude of $H(u,v)$ corresponding to Fig. 3.3	45
3.5b	Phase of $H(u,v)$ corresponding to Fig. 3.3	45
3.6a	Magnitude of $H(u,v)$ corresponding to Fig. 3.4	46
3.6b	Phase of $H(u,v)$ corresponding to Fig. 3.4	46
4.1	Inverse filtering image restoration	50
4.2a	Original image of a point source	56
4.2b	Blurred version of the point source	56
4.3a	Degraded image to be restored	57
4.3b	Restored image by inverse filtering	57
4.3c	Restored image by Wiener filtering	57
4.4	Region R over which $a_{mn} = 0$	66
4.5a	Blurred image by computer simulation	73
4.5b	Original image	73
4.5c	Restored image (global)	74
4.5d	Restored image (local)	74
4.6a	Blurred image by computer simulation	75
4.6b	Original image	75
4.6c	Restored image (global)	76
4.6d	Restored image (local)	76
4.7a	Blurred image by computer simulation	77
4.7b	Original image	77
4.7c	Restored image (global)	78
4.7d	Restored image (local)	78
4.8a	Blurred image by computer simulation	79
4.8b	Original image	79
4.8c	Restored image (global)	80
4.8d	Restored image (local)	80

4.9a	Blurred image by imaging system (camera)	81
4.9b	Original image	81
4.9c	Restored image (global)	82
4.9d	Restored image (local)	82
4.10a	Blurred image by imaging system (camera)	83
4.10b	Original image	83
4.10c	Restored image (global)	84
4.10d	Restored image (local)	84
4.11	Similar images	86
4.12a	Autocorrelation function corresponding to Fig. 4.11a	87
4.12b	Autocorrelation function corresponding to Fig. 4.11b	87
4.12c	Autocorrelation function corresponding to Fig. 4.11c	88
4.12d	Autocorrelation function corresponding to Fig. 4.11d	88
4.13a	Blurred image by camera	90
4.13b	Similar in focus image	90
4.13c	Restored image (local)	90
4.14a	Blurred image by camera	91
4.14b	Similar in focus image	91
4.14c	Restored image (local)	91
4.15a	Blurred image by computer simulation	94
4.15b	Restored image (global)	94
4.16a	Blurred image by computer simulation	95
4.16b	Restored image (global)	95
4.17a	Blurred image by computer simulation	96
4.17b	Restored image (global)	96
4.18a	Blurred image by computer simulation	97
4.18b	Restored image (global)	97

4.19a	Blurred image by camera	98
4.19b	Restored image (global)	98
4.19c	Restored image (local 0% overlapping)	99
4.19d	Restored image (local 50% overlapping)	99
4.20a	Blurred image by camera	100
4.20b	Restored image (global)	100
4.20c	Restored image (local 0% overlapping)	101
4.20d	Restored image (local 50% overlapping)	101
4.21a	Blurred image by camera	102
4.21b	Restored image (global)	102
4.21c	Restored image (local 0% overlapping)	103
4.21d	Restored image (local 50% overlapping)	103
4.22a	Blurred image by camera	104
4.22b	Restored image (global)	104
4.22c	Restored image (local 0% overlapping)	105
4.22d	Restored image (local 50% overlapping)	105
H.1	A sample histogram illustrating a bi-modal distribution	159
H.2	A sample histogram illustrating a multi-modal distribution	159

LIST OF TABLES.

Table No.		Page
1	Different values of 'o' before and after ° processing	108

chapter I
INTRODUCTION.

1.1 OVERVIEW OF DIGITAL IMAGE PROCESSING.

The field of digital image processing can be divided into two principal categories:

- a) *Image Enhancement and Restoration*: which consist in the improvement of pictorial information for human interpretation
- b) *Pattern Recognition*: which consists in processing of scene data for autonomous machine perception.

One of the first applications of image processing techniques in the first category, in the early 1920 s, was in improving digitized newspaper pictures sent by submarine cable between London and New York [1]. Initial problems in improving the visual quality of these early digital pictures were related to the selection of printing procedures, and the distribution of brightness levels. The field of image restoration in the modern sense of the term began in the early 1950 s with the work of Marechal and his co-workers [2]. It was Marechal who first recognized its potential for restoring blurred photographs. Although improvements continued to be made, it took the combined advents of large-scale digital computers and the space program to bring into focus the possibilities of image processing concepts.

Digital computer techniques in image restoration and enhancement had their applications at the Jet Propulsion Laboratory (JPL) of the California Institute of Technology in 1964, when pictures of the Moon transmitted by Ranger 7

were processed by a computer to remove, as well as possible, the degradation from the received Moon images [3].

Today digital image restoration and enhancement is being applied in a number of areas: in medicine (diagnostic Xrays, cellbiology, anatomy, physiology), physics (high energy plasmas), astronomy, biology, defense, and industrial applications.

The second major application area of digital image processing techniques mentioned at the beginning is in problems dealing with machine perception like automatic character recognition, military recognizance, automatic processing of fingerprints, screening of Xrays and blood samples, and machine processing of aerial and satellite imagery for weather prediction.

1.2 DIGITAL IMAGE RESTORATION.

This work concerns an area of digital image processing termed image restoration.

Image restoration is the process which attempts to recover an image that has been degraded. A general block diagram of the situation is shown in Fig. (1.1).

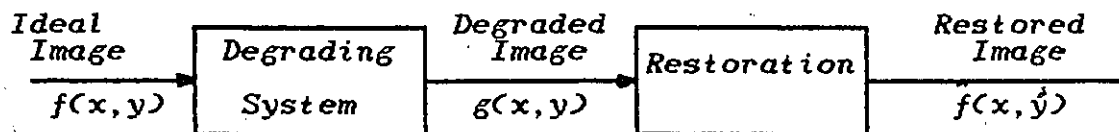


FIG. 1.1 IMAGE RESTORATION.

The purpose of image restoration is to work on the degraded image $g(x,y)$ to get an image $\hat{f}(x,y)$ which is as close to the ideal image $f(x,y)$ as possible according to, e.g., least-squares, or mean-squared error criterion.

1.2.1 Degrading system.

The ultimate goal of image restoration is to manipulate a given image in such a way as to improve its quality (i.e. remove the degradation). There are many sources of degradation including the problems of motion blur, out-of-focus, and atmospheric turbulence.

The first step in the restoration process is the identification of the kind of degradation the image has suffered. In general, the degrading system could be very complex. However, in many cases of practical importance, such as camera motion, atmospheric turbulence, and blurring due to the optical transfer functions of lenses, the degrading system can be modeled by the block diagram shown in Fig. (1.2).

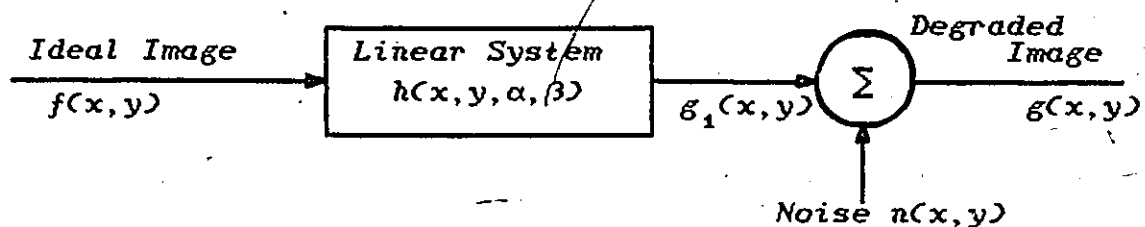


FIG. 1.2. MODEL OF THE IMAGE DEGRADATION PROCESS.

The ideal image $f(x,y)$ is first acted on by a linear system with impulse response $h(x,y,\alpha,\beta)$. The degradation process can be described by:

$$g_1(x,y) = \iint_{-\infty}^{+\infty} h(x,y,\alpha,\beta) f(\alpha,\beta) d\alpha d\beta \quad [1.1]$$

Eqn. (1.1) is called the superposition or Fredholm integral of the first kind.

A noise $n(x,y)$ is added to ' g_1 ' to yield the degraded image

$$g(x,y) = g_1(x,y) + n(x,y)$$

or

$$g(x,y) = \iint_{-\infty}^{+\infty} h(x,y,\alpha,\beta) f(\alpha,\beta) d\alpha d\beta + n(x,y) \quad [1.2]$$

In some cases, the linear system in Fig.(1.2) is shift invariant i.e. $h(x,y,\alpha,\beta) = h(x-\alpha,y-\beta)$ (We will explain this later in chapter III.).

Then Eqn.(1.2) can be written as a convolution

$$g(x,y) = \iint_{-\infty}^{+\infty} h(x-\alpha,y-\beta) f(\alpha,\beta) d\alpha d\beta + n(x,y) \quad [1.3]$$

or

$$g(x,y) = h * f + n \quad [1.4]$$

and

$$g_1(x,y) = h * f$$

1.2.2 Different approaches.

Early techniques for digital image restoration were derived mostly from the frequency domain concepts using the Fast Fourier Transform (FFT).

i/ The Fourier transform approach.

The foundation of the frequency technique is the convolution integral. The real task of image restoration is to recover the ideal image ' f ' from a degraded version ' g '. This can be done easily, if the linear system is shift invariant (LSI),

because from Eqn. (1.4) we have:

$$\hat{F}(U,V) = \frac{G_1(U,V)}{H(U,V)} + \frac{N(U,V)}{H(U,V)} \quad [1.5]$$

We can estimate $\hat{F}(U,V)$ from Eqn. (1.5). This method has been described in several papers [4-6]. The previous technique known as Inverse Filtering gives good results providing that $H(U,V)$ does not vanish at some spatial frequencies. A modified inverse filter to take care of this has been suggested in [5],[7], where the right hand side of Eqn. (1.5) has been modified in such a way to replace $1/H$ by zero in the range of (U,V) over which the noise is larger than the signal. The method is known as Pseudo-Inverse filtering.

In the presence of noise, linear least-squares error filter (Wiener-filter) are more effective [6],[8]

ii/ The Spatial domain approach.

In the linear model, the image restoration is described by the Fredholm integral equation of the first kind given by Eqn. (1.2). The discretization of this equation gives a system of linear equations of the form [9]:

$$g = [H]f + n \quad [1.6]$$

Where g : is the known or given degraded image.

f : is the unknown or undegraded image.

n : is the noise term.

$[H]$: is the real matrix resulting from the discretization of the Point-Spread Function (PSF).

The difficulty of solving Eqn.(1.6) directly is evident in the extremely large size of the linear system. For an image of size $128*128$, the system of equations in Eqn.(1.6) is of order 16384.

A classical approach for solving Eqn.(1.6) is to calculate its least-squares solution. However, Eqn.(1.6) in general is ill-conditioned, because H is near singular, i.e. small changes in 'g' may cause large changes in the solution 'f'. A successful technique for overcoming the ill-posedness of Eqn.(1.6) has been suggested in [10].

1.3 FOCUSING OF IMAGES USING SPATIAL OPERATORS.

As referred to in the title, this work is aimed at developing an image processing scheme that can be used for the purpose of the restoration of images blurred either by computer simulation (simulated blur), or by an image forming system (out-of focus blur), with a minimum priori knowledge about the point-spread function, by utilizing 2-D recursive (IIR) filters. The proposed scheme can be outlined with the help of the block diagram shown in Fig.(1.3).

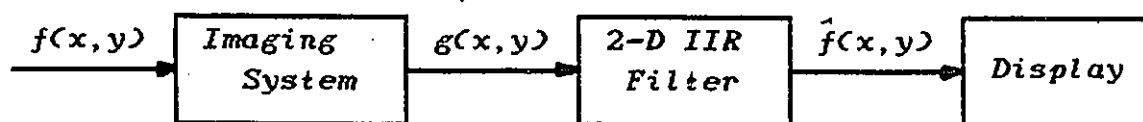


FIG. 1.3. DIGITAL IMAGE RESTORATION SYSTEM.

In the past two decades, extensive efforts have been devoted to the problem of image deblurring under the assumption that the impulse response of the blurring system is known [11-12]. In contrast, in many practical situations one would not have knowledge of the point-spread function (PSF). Efforts at solving the image deblurring problem when the blur is random have been limited.

Early in 1967, Slepian solved the least mean-square estimation problem (Wiener filter), for the case where the degrading impulse response, $h(x,y)$, is random [8].

Stockham et al [13], and Cannon [14] have found first the nature of the blur by the use of the power cepstrum of the blurred image, then carried out the restoration process by a power spectrum equalization filter. The Optical Transfer Function (OTF) was estimated by dividing the given (blurred) image into sub-images, and then taking averages.

Heide [15] developed restoration schemes allowing for random parameters in the system's impulse response function, by assuming that all randomness obeys Gaussian statistics.

Morton and Andrews [16] attempted to remove degradation from an image using a minimum knowledge about the transfer function of the imaging system. The method was similar to that of Cannon with some modifications in the OTF phase estimation. The only necessary assumptions made were: (i) the PSF is spatially invariant, (ii) the extent of the PSF is small compared to the extent of the image, and (iii) the blurred image is noise free.

The above techniques utilized the frequency domain approach.

Ward and Saleh [17] used the well known technique of restoration by superposition of images. The assumptions made in this method were: (i) the impulse response $h(x,y)$ and the noise $n(x,y)$ are assumed to be statistically independent random functions, (ii) the random function $n(x,y)$ is stationary and has zero mean $E(n(x,y))=0$.

This thesis develops spatial domain procedures for restoring images, blurred either by computer simulation or by an image forming system (camera), by utilizing 2-D recursive filters designed to meet the PSF specifications in a least-squares sense [18]. The only blur considered is the out-of focus blur.

Several cases are considered:

- 1- Two (2) similar images are given, one in focus and the other out-of focus;
- 2- when only the blurred image is given.

For the first case, 2-D IIR filter transfer functions are determined by using the two given images.

For the second case, where only the blurred image is given, an estimate for the spatial constant (σ) of the Gaussian PSF is determined from a degraded edge [19]. Then the corresponding 2-D IIR filter coefficients are found via Shanks method [20].

The restoration is achieved either by applying the inverse filter to the total image (globally) or if the blur is more severe application of the filter to sections of the image (local operators) is utilized.

1.4 THESIS ORGANIZATION.

Chapter II gives a detailed account on digital image processing in general.. It can be considered just as a tutorial for readers with limited knowledge about digital image processing.

Chapter III contains a brief discussion on some of the degradation process that can be introduced by an image forming system. We will describe shortly the three major sources of degradation namely: motion blur, out of focus, and atmospheric turbulence. However we will emphasize on the out-of focus blur since we want to restore image degraded by this kind of blur. We will consider our images as relatively noise free so we will not discuss the restoration of noisy images, which is, by itself, a large different field in image restoration, and it utilizes statistical approaches.

Chapter IV gives us a detailed view on the different approaches used in our work. Also the results of applying the proposed algorithms on simulated blur (by computer), as well as, on camera blurred images are provided.

Chapter V finally summarizes the work done and discusses the derived conclusions.

Chapter II

DIGITAL IMAGE PROCESSING

2.1 INTRODUCTION.

The notion of processing an image, involves the transformation of that image from one form into another. Generally, two distinct kinds of processing are possible. One kind involves a form of transformation for which the result appears as a new image which is different from the original image in some desirable way. The input and output are both images, with the output an improved version of the input. The other kind involves a result which is not an image, but may take the form of a decision, an abstraction or a parameterization. The input is an image, but the output is a description of the content of that image.

Techniques for digital image processing may be divided into four principal categories:

- 1- *Image Digitization.*
- 2- *Image Enhancement and Restoration.*
- 3- *Image Encoding.*
- 4- *Image Segmentation and Description.*

Since the first two topics relate to the thesis work, they will be described in the following sections. For the remaining topics (viz., Image Encoding and Image Segmentation and Description) refer to the bibliography cited at the end of this work.

2.2 DIGITAL IMAGE FUNDAMENTAL : THE STRUCTURE OF IMAGES

The images we perceive in our every day visual activities normally consist of light reflected from objects. As an energy, signal light, $f(x,y)$ must be positive and nonzero. This situation is expressed in Eqn.(2.1)

$$0 < f(x,y) < \infty \quad [2.1]$$

where $f(x,y)$ refers to a two-dimensional light intensity function, and x and y represent the spatial domain of the image. Furthermore, since images are commonly formed from light reflected from objects, the structure of images divides physically into two basic parts.

One part is the amount of light available for illuminating the objects; the other is the ability of those objects to reflect light. These image parts are called, respectively, the illumination and reflectance components, and are denoted by $i(x,y)$ and $r(x,y)$. Like the image itself, these basic parts must be positive and nonzero as indicated in Eqns.(2.2)and (2.3).

$$0 < i(x,y) < \infty \quad [2.2]$$

$$0 < r(x,y) < 1 \quad [2.3]$$

The functions $i(x,y)$ and $r(x,y)$ combine, according to the law of reflection, to form the image $f(x,y)$. Since that law is a product law, Eqns.(2.2) and (2.3) combine as Eqn.(2.4)

$$0 < f(x,y) = i(x,y)r(x,y) < \infty \quad [2.4]$$

which is in agreement with Eqn.(2.1).

It follows from Eqn.(2.4) that two basic kinds of information are stored by an image. The first is carried by $i(x,y)$, and has to do primarily with the lighting of the scene. The second is carried by $r(x,y)$ and is concerned with the nature of the objects in the scene ($r(x,y)=0$, means total absorption like black objects, and $r(x,y)=1$, means total reflectance like white objects).

2.3 IMAGE DIGITIZATION.

To process any given image by a digital computer, it is necessary to sample and discretize the given image.

The image function $f(x,y)$ must be digitized both spatially and in amplitude. Digitization of the spatial coordinates (x,y) will be referred to as image sampling, while amplitude digitization will be called gray-levels quantization. The image is then represented as a two-dimensional matrix, whose elements represent the gray-levels and their relative position in the image. Each element of the array is referred to as an image element, picture element, pixel or pel.

The fundamental components of an image processing system are shown in Fig.(2.1).

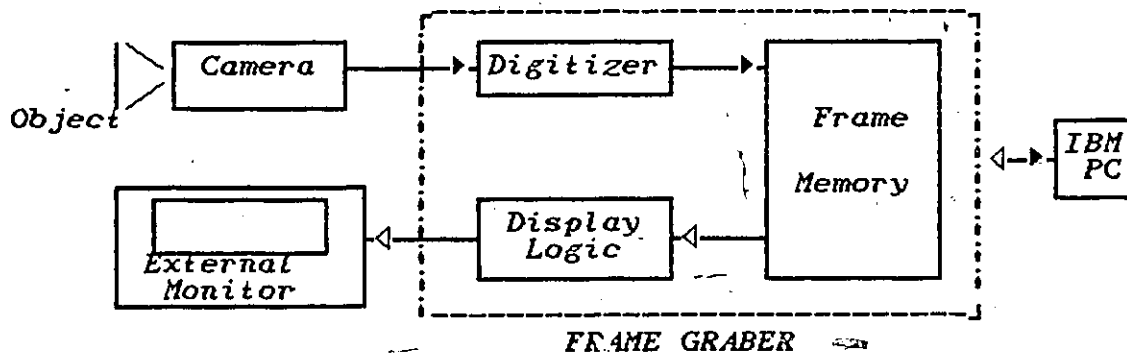


FIG. 2.1 COMPONENTS OF A DIGITAL IMAGE PROCESSING SYSTEM.

The image originates at a video source, by means of a vidicon camera. The analog signal produced by the camera is then transformed to a digital format through a process called digitization (analog to digital converter). The digitization process involves taking samples of the analog signal at discrete time intervals, and converting each individual sample, or pixel, to a digital value. The digital data is then stored in the frame memory. Display logic transforms the pixels stored in the frame memory back into an analog format (digital to analog converter), so that the image may be viewed on a monitor.

The next step in the digitization process requires that a value for N , the number of samples (dimension of the matrix), as well as, on G , the number of discrete gray-levels for each pixel, has to be selected. In practice in digital image processing these quantities are chosen to be integer of two; that is.

$$N = 2^n \quad [2.5]$$

and

$$G = 2^m \quad [2.6]$$

The digitization of an image is usually arranged in the form of a square array of dimension $N \times N$.

The resolution (i.e., the degree of discernable details) of an image is strongly dependent on both N and m . When these parameters are increased the resolution increases and vice-versa. As a basis for comparison, the requirements to obtain a quality comparable to that of monochrome TV pictures, N should be equal to 512 and m to 7 [21] (i.e., 512×512 pixels with 128 gray-levels).

2.4 IMAGE ENHANCEMENT AND RESTORATION.

Since chapter IV will be devoted to image restoration, we are going, in the following, to describe; first the difference between image enhancement and restoration, and then we will give outlines for the different techniques for digital image enhancement.

Both enhancement and restoration techniques deal with the improvement of a given image for human or machine perception. Enhancement techniques are designed to manipulate an image to take advantage of the psychological aspects of the human visual system, while restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image.

2.4.1 Enhancement approach.

Digital image enhancement techniques may be divided into two principal categories [21]:

- 1- *Spatial-domain methods*
- 2- *Frequency-domain methods.*

Spatial-domain enhancement techniques consist of procedures that operate directly on the pixels of the image in question. Examples of spatial-domain enhancement techniques include smoothing by neighborhood averaging, sharpening by using gradient type operators, and global enhancement by means of histogram modification techniques [22-26].

Approaches based on the second category consist basically of computing a two-dimensional transform (generally Fourier transform) of the image to be enhanced, modifying the transform, and computing the inverse yield an image that has been enhanced in some manner. Examples of frequency-domain techniques include low and high-pass filtering for image smoothing and sharpening, respectively, and homomorphic filtering for manipulating the effects of illumination and reflectance in an image [22-23].

2.4.1.1. SPATIAL-DOMAIN METHODS.

A- HISTOGRAM MODIFICATION TECHNIQUES.

The methods discussed in this sub-section achieve enhancement by modifying the histogram of a given image in a specified manner. Before proceeding with a discussion of the different techniques, we have to define what is meant by histogram. The histogram is the probability density function of the gray-levels, or in other words, it is the plot of $P_x(x_k)$ versus x_k for $k = 0, 1, \dots, L-1$; where L is the number of levels, $P_x(x_k)$ is the probability of the k^{th} gray level.

The histogram is of fundamental importance in describing the visual characteristics of an image.

1- Histogram equalization.

Histogram equalization is the technique which consist on transforming the histogram $P_x(x)$ of the original image in such a way for obtaining a uniform histogram $P_y(y)$. The problem can be stated as to determine the transformation function $T(x)$ which will yield the desired $P_y(y)$ from $P_x(x)$. The transformation T is determined from the desired $P_y(y)$; the transformation function $y = T(x)$ and its inverse $x = T^{-1}(y)$, are guaranteed to be strictly monotonically increasing in the interval $[0,1]$ if the function $T(x)$ is strictly monotonic in the same interval and $0 < T(x) < 1$ for $x \in [0,1]$. Under these conditions, $P_y(y)$ can be written in terms of $P_x(x)$ and $T(x)$ as follows [21]:

$$P_y(y) = \left[P_x(x) \frac{dx}{dy} \right]_{x = T^{-1}(y)} \quad [2.71]$$

A density equalization technique [23-26] is obtained from Eqn.(2.7) by using the transformation function

$$y = T(x) = \int_0^x P_x(s) ds \quad 0 < x < 1 \quad [2.8]$$

The most right-hand side of Eqn.(2.8) is known as the cumulative distribution function (CDF) of x.

From Eqn.(2.8) we have that $\frac{dx}{dy} = \frac{1}{P_x(x)}$, and Eqn.(2.7)

reduces to:

$$P_y(y) = 1 \quad 0 < y < 1 \quad [2.9]$$

which is a uniform density.

2- Histogram specification.

Although histogram equalization can be quite useful in some applications, but this method is not appropriate for interactive image enhancement since all it can do is to produce a density function $P_y(y)$ that is uniform. As it will be seen below, however, this approach can be used as an intermediate step in a transformation which will yield a specified $P_y(y)$. This transformation procedure is referred to as histogram specification [21], [26].

For the previous technique (histogram equalization), the gray-levels x of an image are transformed using Eqn.(2.8) to yield a new uniform set of levels z; that is

$$z = H(x) = \int_0^x P_x(s) ds \quad [2.10]$$

We know from the above discussion that $P_z(z)$ is a uniform density function. It is noted, however if we specify a density function $P_y(y)$ and apply Eqn.(2.8) we will obtain:

$$z = G(y) = \int_0^y P_y(s) ds \quad [2.11]$$

If the original image is first density equalized and the new (uniform) levels are inverse mapped using the function $G^{-1}(z)$, the result would be an image whose gray-levels have the desired $P_y(y)$. In terms of a transformation function $T(x)$ from x to y , by using Eqns.(2.10) and (2.11) we will end up with

$$y = T(x) = G^{-1} [H(x)] \quad [2.12]$$

It is noted that, when $G^{-1} [H(x)] = H(x)$, this expression reduces to histogram equalization. It is also noted that, the development have been carried out in continuous mathematics to simplify explanation. The discrete equivalents are then obtained by straightforward extensions of these results and are given below.

The discrete form of Eqn.(2.8) for histogram equalization is given by:

$$y_k = T(x_k) = \sum_{j=0}^k \frac{n_j}{n} \quad \begin{array}{l} 0 \leq x_k \leq 1 \\ k = 0, 1, \dots, L-1 \end{array} \quad [2.13]$$

where L is the number of levels, n_k is the number of times this level appears in the digital image, and n is the total number of pixels in the image.

The discrete form of Eqn. (2.12) for histogram specification is given by:

$$y_k = T(x_k) = G^{-1} [H(x_k)] \quad \begin{array}{l} 0 \leq x_k, y_k \leq 1 \\ k = 0, 1, \dots, L-1 \end{array} \quad [2.14]$$

B-IMAGE SMOOTHING.

Image smoothing is used for removing noise present in a digital image as a result of a poor sampling system or transmission channel. In this sub-section we will discuss methods of noise removal by the use of averaging schemes in the spatial-domain.

1. Neighborhood averaging.

If the noise can be distinguished from the signal, it becomes relatively easy to remove it, while leaving the signal more or less intact (this expression more or less intact will be explained by the following). For example if the noise consists of isolated dots ("salt and pepper"); while the ideal image does not contain such dots, we can detect the dots as pixels that differ greatly in gray-level from nearly all of their neighbors; such a pixel can then be replaced by the average of its neighbors. This technique is known as neighborhood averaging. The smoothed image is obtained by using the relation:

$$g(x,y) = \frac{1}{M} \sum_{(m,n) \in S} f(m,n) \quad [2.15]$$

for $x,y = 0,1,\dots,N-1$

where

$f(x,y)$: is the original image.

$g(x,y)$: is the smoothed image.

S : is the set of coordinates of points in the neighborhood of (but not including) the point (x,y) .

M : is the total number of points in S .

N : is the size of the image.

Fig. (2.2) shows one approach for extracting neighborhoods from an image array

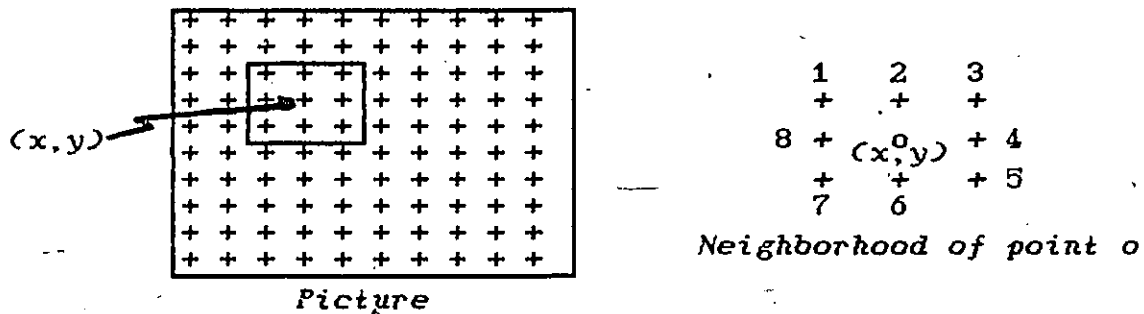


FIG. 2.2 EXTRACTION OF NEIGHBORHOODS FROM AN IMAGE ARRAY.

From Fig. (2.2) we have

$$S = \{(x-1, y-1); (x, y-1); (x+1, y-1); (x+1, y); (x+1, y+1); \\ (x, y+1); (x-1, y+1); (x-1, y)\}$$

and $M = 8$

The principal problem with the previous discussed method is that, in removing the noise it also blurs the edges. We will next discuss how to avoid averaging across edges.

2. Selective averaging.

To avoid blurring edges when locally averaging an image, we can use a thresholding procedure (the word thresholding will be explained in the image segmentation section see appendix (H)). Instead of using Eqn. (2.15), we form $g(x,y)$ according to the following criterion:

Note that all the discussed procedures treat thin curves, and round sharp corners as noise [27].

4 Averaging of multiple images.

There exist other techniques for smoothing operation, one of them is based on the averaging of multiple images.

We have

$$g(x,y) = f(x,y) + \eta(x,y) \quad [2.17]$$

$$\bar{g}(x,y) = \frac{1}{M} \sum_{i=0}^M g_i(x,y) \quad [2.18]$$

where M : is the total number of the same transmitted image.

$f(x,y)$: is the noise free image.

$\eta(x,y)$: is a 2-D noise process.

$g(x,y)$: is the original (given) image.

$\bar{g}(x,y)$: is the averaged image.

The assumption made in this method is that the 2-D noise process $\eta(x,y)$ is additive, and has zero average.

We'll have [28]:

$$E \{ \bar{g}(x,y) \} = f(x,y) \quad [2.19]$$

where $E\{\bar{g}(x,y)\}$ is the expected value of \bar{g} .

C. IMAGE SHARPENING.

Since averaging pixels locally tends to blur edges, we can expect that differentiation can sharpen an image (averaging is equivalent to integration)

1 Sharpening by differentiation.

The most commonly used method of differentiating in image processing is the gradient [21].

The gradient of a function $f(x,y)$ is defined mathematically by:

$$G \{ f(x,y) \} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} \quad [2.20]$$

The vector G points in the direction of maximum rate of increase of the function $f(x,y)$.

The magnitude of G is given by:

$$\text{Mag} [G(f(x,y))] = |G| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad [2.21]$$

for discrete images

$$|G| \approx \left\{ [f(x,y) - f(x+1,y)]^2 + [f(x,y) - f(x,y+1)]^2 \right\}^{1/2} \quad [2.22]$$

or

$$|G| \approx |f(x,y) - f(x+1,y)| + |f(x,y) - f(x,y+1)| \quad [2.23]$$

The other useful approximation for the gradient, called Roberts gradient is given by:

$$|G| = |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)| \quad [2.24]$$

The relationship between pixels in Eqns. (2.23) and (2.24) is shown in Fig. (2.4) (a) and (b), respectively.



FIG. 2.4. DISCRETE GRADIENTS.

2.4.1.2. FREQUENCY-DOMAIN METHODS.

The foundation of frequency domain techniques is the convolution theorem.

$$g(x,y) = h(x,y) * f(x,y) \quad [2.25]$$

where $g(x,y)$ is an image formed by the convolution of an image $f(x,y)$, and a position invariant operator $h(x,y)$ (the meaning of position invariant will be given in chapter III).

The Fourier transform of Eqn.(2.25) is given by:

$$G(u,v) = H(u,v).F(u,v) \quad [2.26]$$

G, H, and F are the Fourier transforms of g , h , and f respectively

A- IMAGE SMOOTHING BY LOWPASS FILTERING.

Noise, as well as, edges in an image are in the high-frequency content of its Fourier transform. Thus removing noise via frequency-domain can be achieved by attenuating those high-frequency components in the transform of the given image.

Image smoothing in the frequency-domain can be stated as a problem of selecting a specified function $H(u,v)$ which yields $G(u,v)$ by attenuating the high-frequency components of $F(u,v)$. The inverse transform of $G(u,v)$ will then yield the desired smoothed image $g(x,y)$. This technique is known as lowpass filtering since all high-frequency components of $F(u,v)$ will be removed.

Several lowpass filtering approaches can be found in the literature [21], [29], [30], here we are going to shortly describe the ideal lowpass filter (ILPF).

1- Ideal lowpass filter (ILPF).

A two-dimensional ideal lowpass filter has its transfer function given by:

$$H(u,v) = \begin{cases} 1 & \text{for } D(u,v) \leq D_0 \\ 0 & \text{for } D(u,v) > D_0 \end{cases} \quad [2.27]$$

where D_0 is a specified nonnegative quantity

and $D(u,v) = \sqrt{u^2 + v^2}$ (i.e, the distance from point (u,v) to the origin).

A 3-D perspective plot of $H(u,v)$ is shown in Fig.(2.5) (a), while the filter cross section is shown in Fig.(2.5) (b).

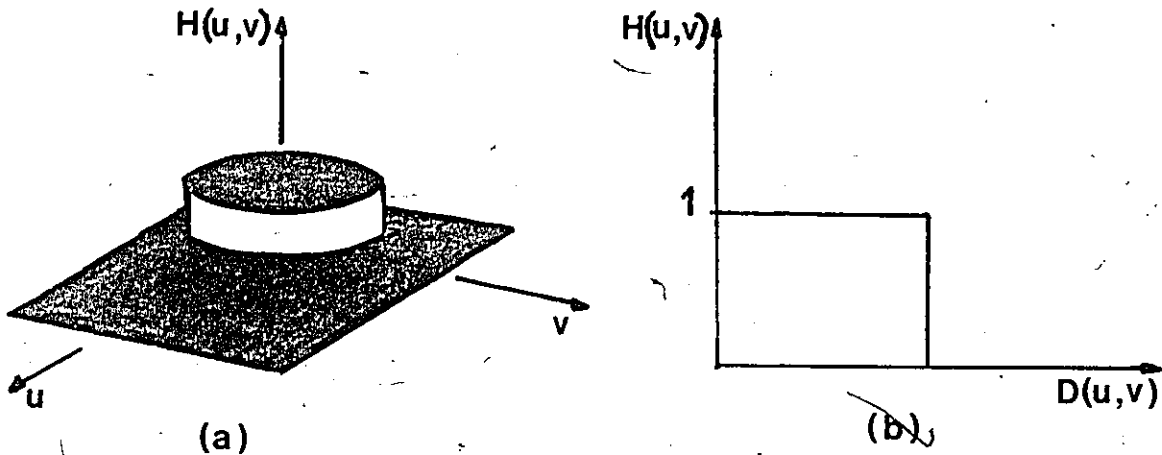


FIG. 2.5 a) PERSPECTIVE PLOT OF AN IDEAL LOWPASS FILTER TRANSFER FUNCTION. b) FILTER CROSS SECTION.

All the frequencies inside the circle of radius D_0 are passed without attenuation ($H(u,v)=1$), while all the other frequencies are completely attenuated ($H(u,v)=0$).

B- IMAGE SHARPENING BY HIGHPASS FILTERING.

It was shown in the previous subsection (A), that edges and other sharp transitions (such as noise) of an image are associated with the high-frequency components of its Fourier transform. Thus image sharpening via frequency-domain can be achieved by attenuating the low-frequency components without disturbing high-frequency information of the Fourier transform. The problem now is to select a function $H(u,v)$ which yields $G(u,v)$ by attenuating low-frequency components of $F(u,v)$. The inverse transform of $G(u,v)$ will then yield the desired sharpen image $g(x,y)$.

This technique is known as highpass filtering, since all low-frequency components of $F(u,v)$ will be removed. Similar to the lowpass filter, several approaches can be found in the literature [21], [29], [30], we are going to describe the ideal highpass filter (IHPF) following the same procedure as in subsection (A).

1. Ideal highpass filter (IHPF).

Its transfer function is given by:

$$H(u,v) = \begin{cases} 0 & \text{for } D(u,v) \leq D_0 \\ 1 & \text{for } D(u,v) > D_0 \end{cases} \quad [2.28]$$

where D_0 is the cut-off distance (radius), and $D = \sqrt{u^2 + v^2}$.

A 3-D plot of $H(u,v)$, and its radial cross section are shown by Figs. (2.6) (a), and (b), respectively.

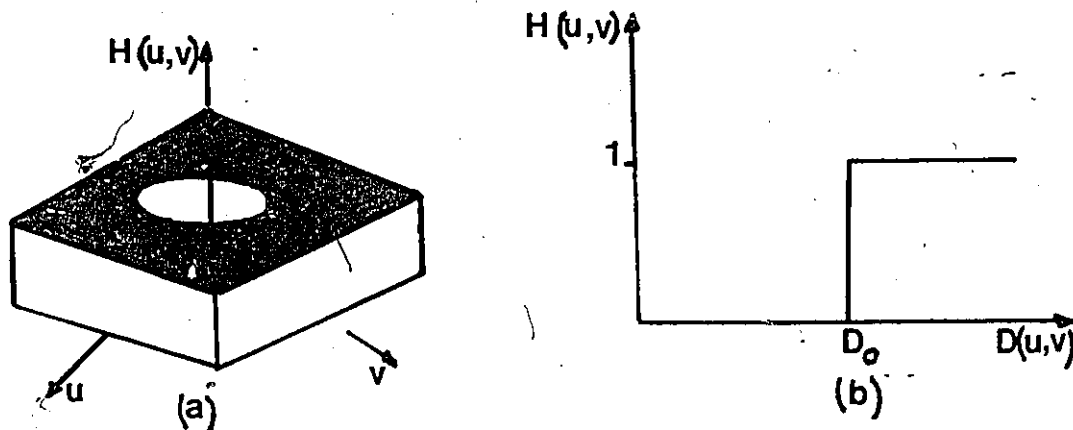


FIG. 2.6 a) PERSPECTIVE PLOT OF AN IDEAL HIGHPASS FILTER TRANSFER FUNCTION. b) FILTER CROSS SECTION.

All the frequencies inside the circle of radius D_0 are completely attenuated ($H(u,v) = 0$), while all other frequencies are passed without attenuation ($H(u,v) = 1$).

C- ENHANCEMENT BASED ON AN IMAGE MODEL: HOMOMORPHIC
FILTERING.

From section (2.1), we showed that the structure of images divides physically into two parts (illuminance, and reflectance), which combined according to the law of reflection form the image $f(x,y)$ given by:

$$f(x,y) = i(x,y) \cdot r(x,y) \quad [2.29]$$

Homomorphic filtering is a frequency-domain procedure which operates on the frequency components of illumination and reflectance separately. The separation of the two previously named components of the image is achieved by taking the logarithm of the image, because the Fourier Transform of the product of two functions is not separable i.e.,

$$F \{ f(x,y) \} \neq F \{ i(x,y) \} F \{ r(x,y) \} \quad [2.30]$$

but if we let

$$s(x,y) = \text{Ln} \{ f(x,y) \} = \text{Ln} \{ i(x,y) \} + \text{Ln} \{ r(x,y) \} \quad [2.31]$$

Hence

$$F \{ s(x,y) \} = F \{ \text{Ln} (i(x,y)) \} + F \{ \text{Ln} (r(x,y)) \}$$

or

$$F \{ s(x,y) \} = I(u,v) + R(u,v) = S(u,v) \quad [2.32]$$

If we process $S(u,v)$ by means of a filter with transfer function $H(u,v)$, it follows from Eqn.(2.26)

$$Z(u,v) = H(u,v).S(u,v)$$

OR

$$Z(u,v) = H(u,v).I(u,v) + H(u,v).R(u,v) \quad [2.33]$$

Back to the spatial domain we have the relation

$$z(x,y) = F^{-1} \{ Z(u,v) \}$$

OR

$$z(x,y) = F^{-1} \{ H(u,v).I(u,v) \} + F^{-1} \{ H(u,v).R(u,v) \}$$

$$z(x,y) = i'(x,y) + r'(x,y) \quad [2.34]$$

Since $s(x,y)$ was obtained by taking the logarithm of the original image $f(x,y)$, we have to perform the inverse operation to obtain the enhanced image $g(x,y)$; that is

$$g(x,y) = \exp (z(x,y)) = \exp (i'(x,y) + r'(x,y))$$

OR

$$g(x,y) = \exp (i'(x,y)).\exp (r'(x,y)) = i_0(x,y).r_0(x,y) \quad [2.35]$$

The method described above is summarized by the block diagram shown in Fig.(2.7)

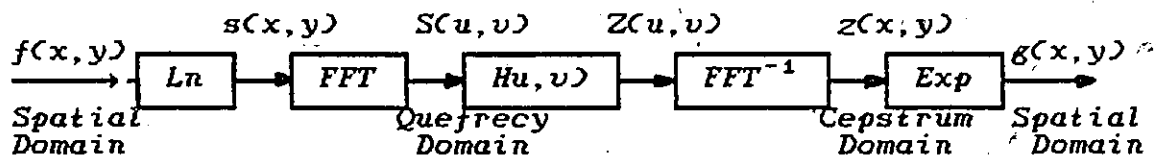


FIG. 2.7 HOMOMORPHIC FILTERING APPROACH FOR IMAGE ENHANCEMENT

The main problem now is how to select $H(u,v)$ the filter transfer function. What is known only is that the low-frequencies of the Fourier transform in the Quefrécy domain, are associated with the illumination components; while reflectance content high-frequency components.

A cross section of a filter function $H(u,v)$ which will affect low and high-frequency components of the Fourier transform in different ways is shown in Fig.(2.8).

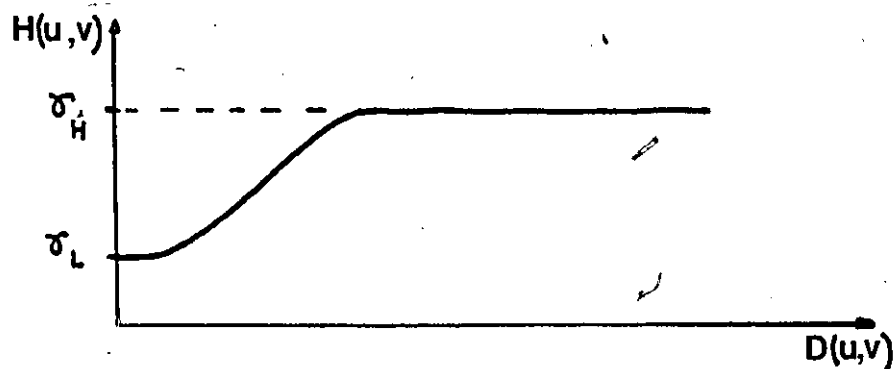


FIG. 2.8 FILTER CROSS SECTION.

A filter with a transfer function $H(u,v)$ like the one shown in Fig.(2.8) will decrease the low-frequencies and amplify the high ones. The result of a such filter is a simultaneous dynamic range compression in the brightness, together with an increase in contrast.

2.4.2 Conclusion.

In conclusion a block diagram which is shown in Fig. (2.9) summarizes, and classifies the image enhancement techniques developed in this section.

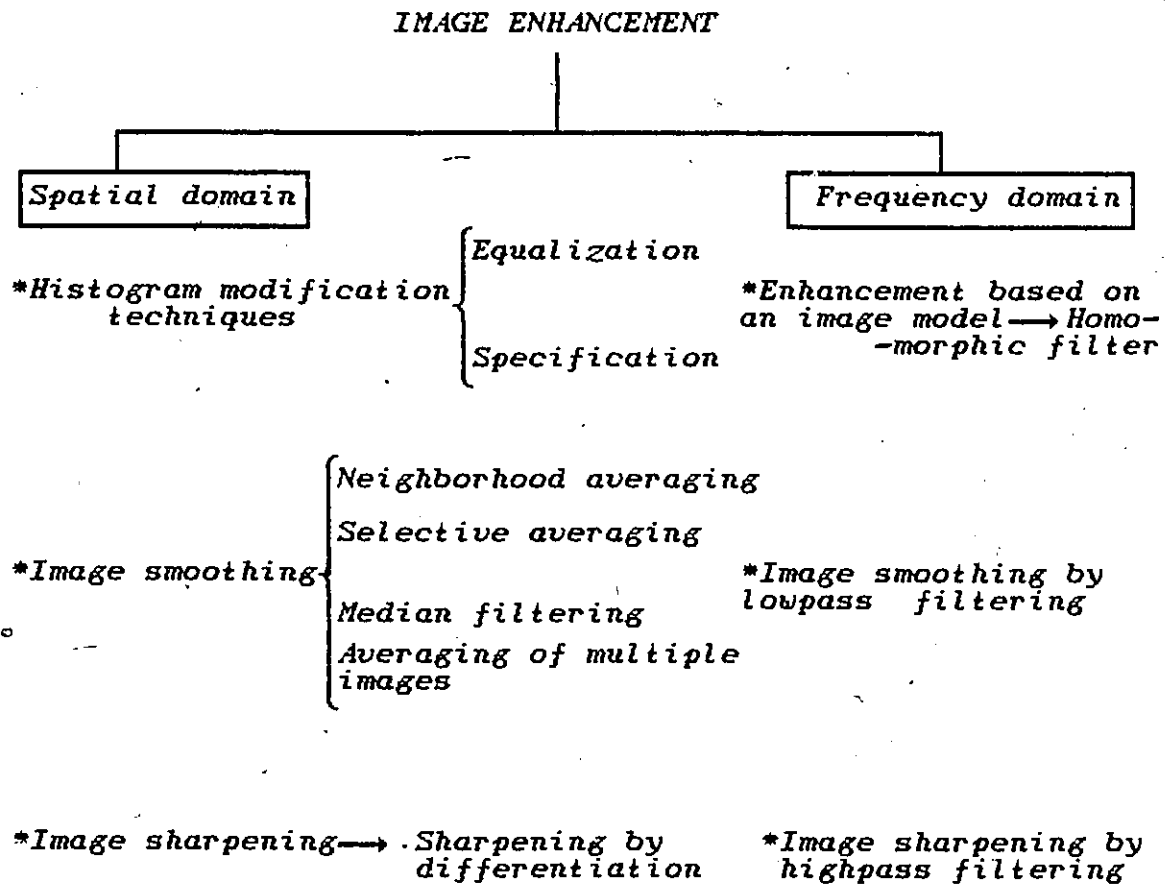


FIG. 2.9 BLOCK DIAGRAM OF IMAGE ENHANCEMENT TECHNIQUES.

chapter III
DEGRADATION PROCESS.

3.1 INTRODUCTION.

In order effectively to design a digital image restoration system, it is necessary quantitatively to characterize the image degradation effects before attempting to perform the inverse operation to obtain a restored image. It should be emphasized that the degree of restoration success is dependent on the degradation model. Since the restoration procedure is highly dependent on the appropriateness of modeling the degradation process, it is logic to start this chapter by explaining the different sources of degradation. However, since we are dealing in this work with an image forming system, we will emphasize on the three major sources of degradation namely:

- motion blur.
- out-of focus.
- atmospheric turbulence.

We will devote a large amount on the lens aberration (out-of focus) degradation.

3.2 DIFFERENT SOURCES OF DEGRADATION.

There are many sources of degradation including the problem of motion blur, atmospheric turbulence, and out-of-focus.

3.2.1 Motion blur.

Motion blur occurs when there is a relative motion between the object to be taken and the imaging system during exposure. The object appears at each instant as a translated, rotated version of what is termed the object plane. Thus the motion may be interpreted as relative rotation and translation of two constant-distance parallel planes. When the motion is merely along one spatial coordinate, the degraded image can be viewed as a family of identically degraded line images.

The point-spread function resulting from a uniform camera motion can be approximated by a rectangle whose orientation and length are indicative of the direction and extent of blur. In the case of a horizontal motion of length $2d$ the PSF function is given by [14]:

$$h(x,y) = \begin{cases} 0 & y \neq 0 & -\infty \leq x \leq +\infty \\ \frac{1}{2d} & y = 0 & -d \leq x \leq +d \end{cases} \quad [3.1]$$

The Fourier transform of the PSF for a motion blur of length d whose direction is θ degrees off the horizontal is given by [31], [32]:

$$H(u, v) = \frac{\sin \Pi d f}{\Pi d f} \quad [3.2]$$

where $f = u \cos \theta + v \sin \theta$

3.2.2 Atmospheric turbulence.

Of all the sources of degradation which ultimately limit the resolution in astronomical photography with large earth telescopes, turbulence in the atmosphere causes the greatest degradation. Most of the degradation is due to thermally induced refractive index variations in the incoming electromagnetic waves which is due to random variation in the refractive index of the medium between the object and the imaging system. Theoretical analysis of this problem shows that the effect of such turbulence is to reduce image contrast at the higher spatial frequencies of the object so that the atmosphere resembles a low-pass filter [33]. The turbulence produces a relatively constant attenuation beyond a certain spatial frequency. The amount of attenuation produced by the atmosphere depends upon the integration time of the exposure.

One of the methods to restore such degraded images is by means of high-pass filtering (if the attenuation is not so severe that the higher frequencies are lost in the detector noise) in the Fourier domain. The short time exposure results in an averaging of the random phase variations in the Fourier transform of the scene blurred by the atmospheric turbulence so that a simple amplitude filter is adequate to restore the higher frequencies. More complex filtering and averaging processes have been investigated. All of these are applied to individual photographs sequences of short time exposures taken with large telescopes in an effort to achieve substantially better resolution with these instruments than has been possible up until now.

3.2.3 Out-of focus blur.

Most lens systems are exactly focused at only one distance along each radius from the lens into the scene. The locus of exactly focused points forms a doubly curved, approximately spherical surface in three-dimensional space. Only when objects in the scene intersect this surface is their image exactly in focus; object distant from this surface of exact focus are blurred. The amount of defocus or blurring depends solely on the distance to the surface of exact focus, as the distance between the imaged point and the surface of exact focus increases, the imaged objects become progressively more defocused.

Fig.(3.1) shows the situation in which a lens of radius r is used to project a point at distance u onto an image plane at distance v_0 behind the lens. Given this configuration, the point would be focused at distance v behind the lens, but in front of the image plane. Thus a blur circle is formed on the image plane. In other words, when a point at distance $u > u_0$ (or $u < u_0$) is projected through the lens, it focuses at a distance $v > v_0$ (or $v < v_0$) so that a blur circle is formed [19].

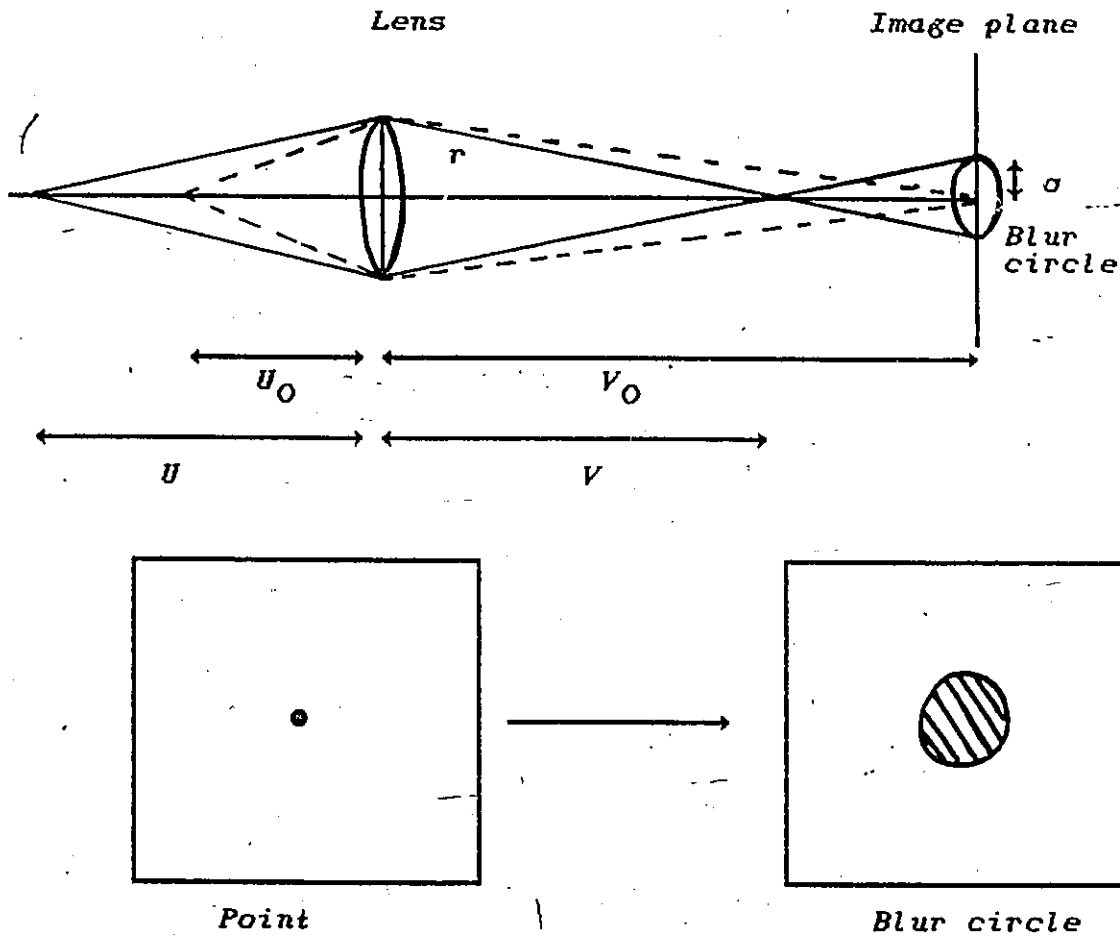


FIG. 3.1 GEOMETRY IMAGING.

From Fig. (3.1) we have:

- u : is the distance between a point in the scene and the lens.
- v : is the distance between the lens and the plane on which the image is in perfect focus.
- v_0 : is the distance between the image pulse and the lens.
- u_0 : is the distance between the lens and the locus of perfect focus.

3.3 IMAGING SYSTEM.

Every visual scene is an image. The eye is the first image formation system, and of course, is not the only one. Imaging devices consisting of lenses, mirrors, prisms, and so on can be considered to provide a deterministic transformation of an input spatial light to some output spatial light distribution. The formulation of a complete theory of optical imaging from the basic electromagnetic principles of diffraction theory is a complex and lengthy task; details may be found in reference [34]. In the following, only the important points of the formulation are presented. Most optical image formation systems are considered as to respond as a linear system in terms of the intensity of its input and output. In optical systems, the radiant light intensity reflected or emitted by the object is transformed by a set of lenses and apertures. The most common optical image formation system is a circular thin lens.

The relationship between the image intensity and the object intensity for the optical system can then be represented by the superposition integral equation

$$g(x, y) = \iint_{-\infty}^{+\infty} f(\alpha, \beta) h(x, y, \alpha, \beta) d\alpha d\beta \quad [3.3]$$

where $h(x, y, \alpha, \beta)$ represents the image intensity response to a point source of light and is known as the point-spread function (PSF).

If 'h' is position invariant, we will explain this later, we will have

$$h(x, y, \alpha, \beta) = h(x - \alpha, y - \beta) \quad [3.4]$$

Eqn. (3.3) reduces in this case to

$$g(x, y) = \iint_{-\infty}^{+\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad [3.5]$$

which is recognized as the convolution integral.

We can outline the principal elements of an image formation system into a diagram such as the one given in Fig. (3.2).

The Fourier transform of Eqn. (3.5) is given by

$$G(u, v) = H(u, v)F(u, v) \quad [3.6]$$

$H(u, v)$ is called the optical transfer function (OTF) of the imaging system (the OTF is the Fourier transform of the PSF).

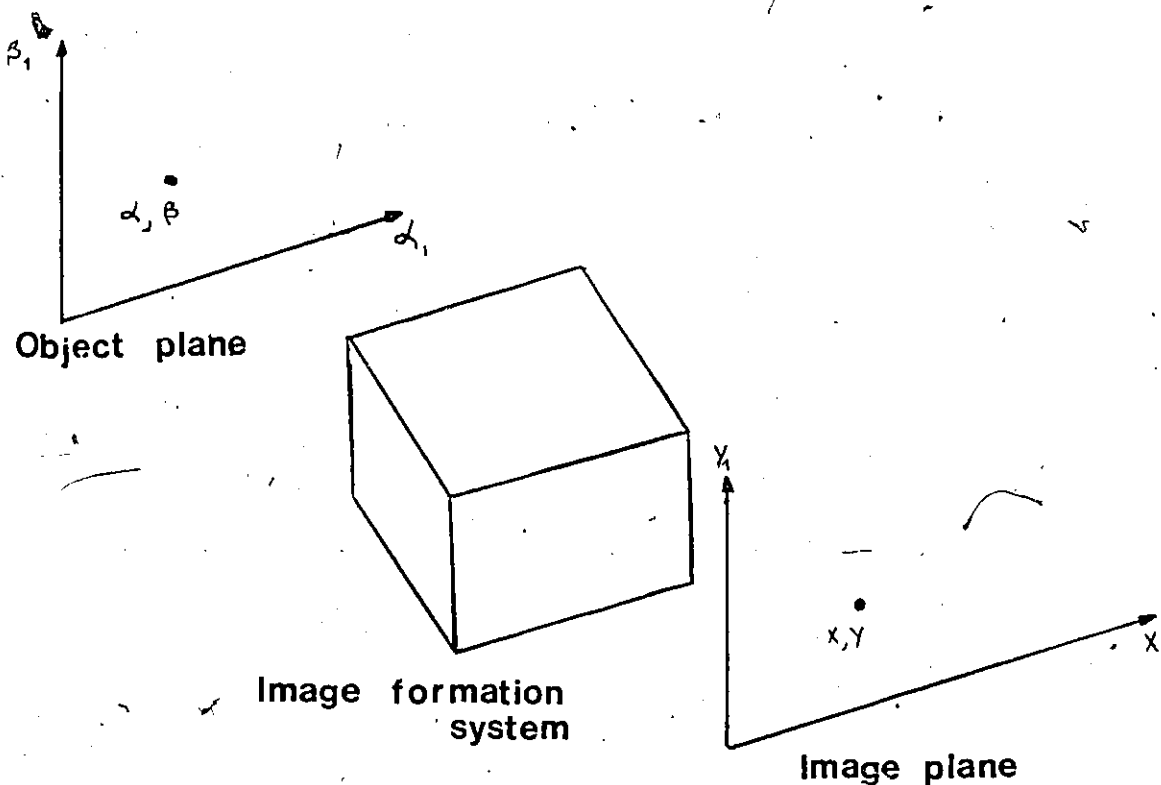


FIG. 3.2 SCHEMATIC OF IMAGE FORMATION SYSTEM.

3.4 SVPSF AND SIPSF.

We introduced in subsection 3.3, the notion of position invariance, here in the following we are going to define what is meant by space variant and space invariant for linear systems. If as the point source explores the object plane, the form of the impulse or point-spread function changes shape as well as position, then a spatially variant point spread function (SVPSF) results. If, however, as the point source explores the object plane, the point-spread function changes only position but maintains the same functional shape then a spatially invariant point-spread function (SIPSF) is said to exist [7].

3.5 THE POINT-SPREAD FUNCTION OF IMAGING SYSTEM.

Before developing restoration procedures the point-spread function for the imaging system is first studied by performing the following test experiment.

The Fourier transform of two blurred and in focus images shown in Figs. (3.3) and (3.4) are calculated.

It follows from Eqn. (3.6) that

$$G(u, v) = H(u, v) F(u, v) \quad [3.7]$$

where $G(u, v)$: Fourier transform of blurred image.
 $F(u, v)$: Fourier transform of in focus image.
 $H(u, v)$: Fourier transform of the imaging system point-spread function.

Hence

$$H(u, v) = \frac{G(u, v)}{F(u, v)} \quad [3.8]$$

The plots of the magnitude and phase of the optical transfer function $H(u, v)$ are shown in Figs. (3.5) and (3.6) for both cases.

Because of the clear dissimilarities of the transfer functions for the two cases, it can be concluded that the degradation introduced by the camera is not unique and has to be determined for every case.

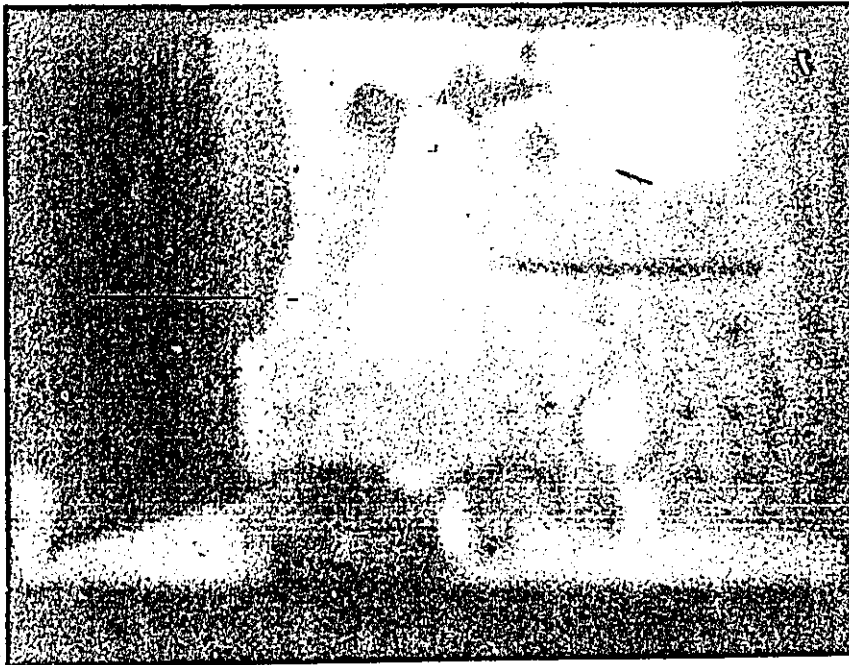


FIG 3.3 a) *OUT-OF FOCUS IMAGE.*

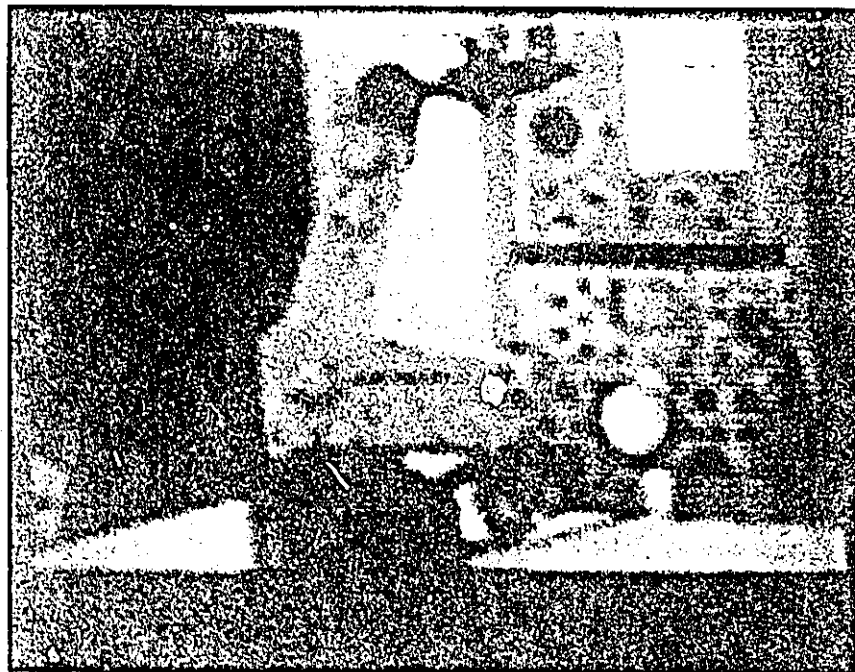


FIG 3.3 b) *IN FOCUS IMAGE.*



FIG 3.4 a) *OUT-OF FOCUS IMAGE.*



FIG 3.4 b) *IN FOCUS IMAGE.*

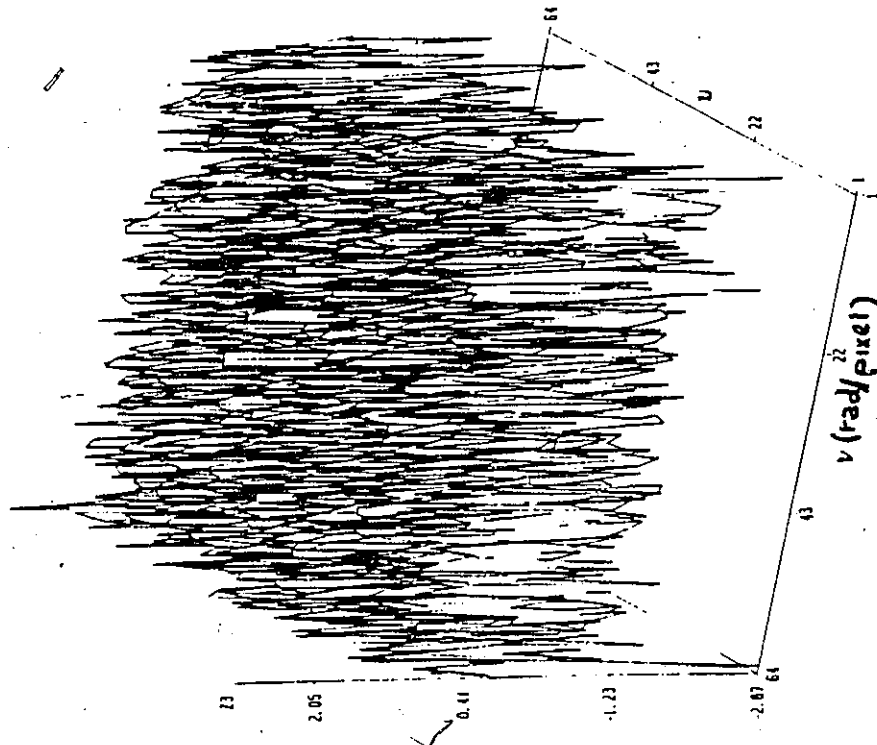


FIG. 3.5 b) PHASE OF $H(u, v)$

CORRESPONDING TO FIG. (3.3).

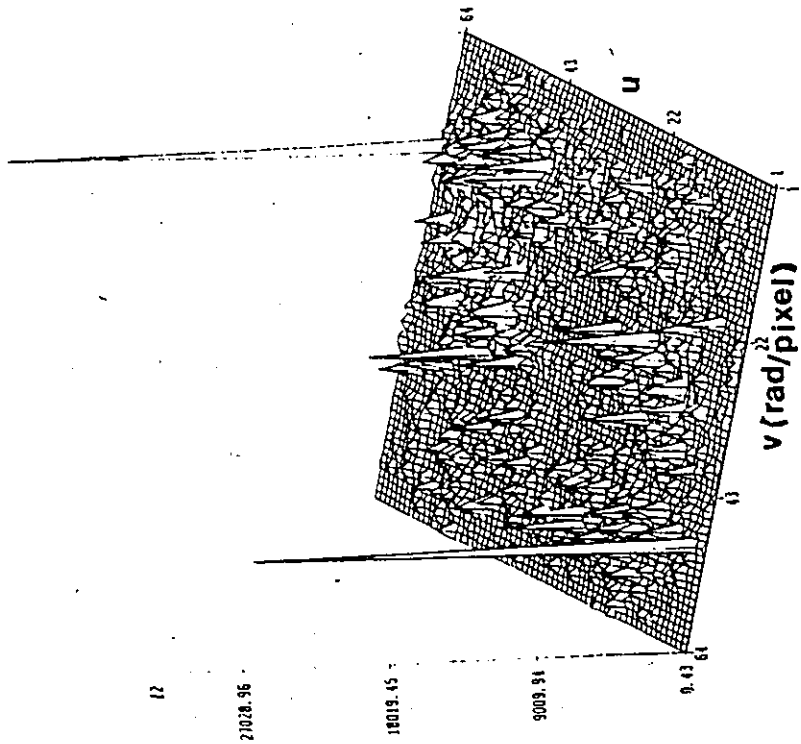


FIG. 3.5 a) MAGNITUDE OF $H(u, v)$

CORRESPONDING TO FIG. (3.3).

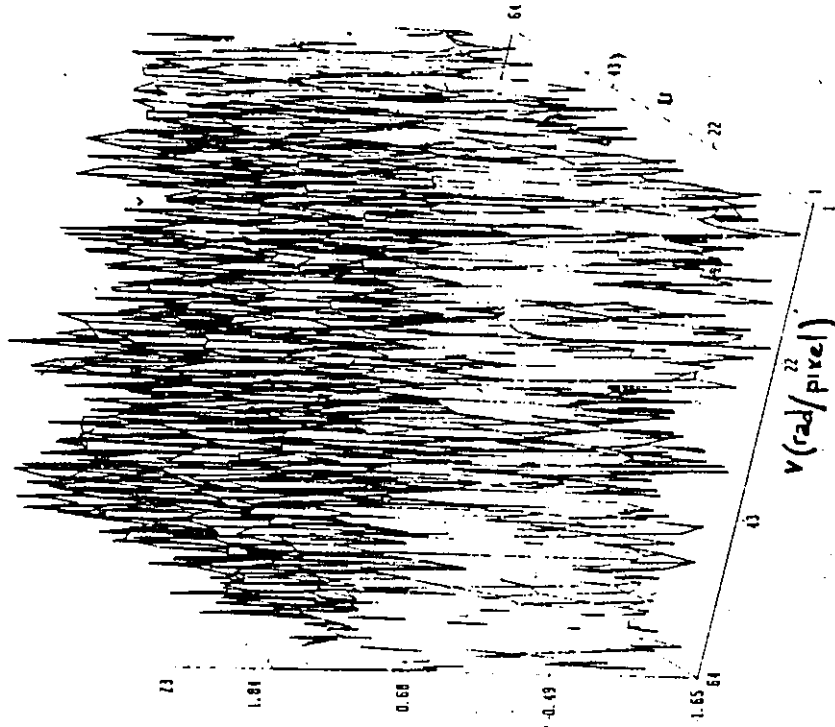


FIG. 3.6 a) MAGNITUDE OF $H(u, v)$
CORRESPONDING TO FIG. (3.4).

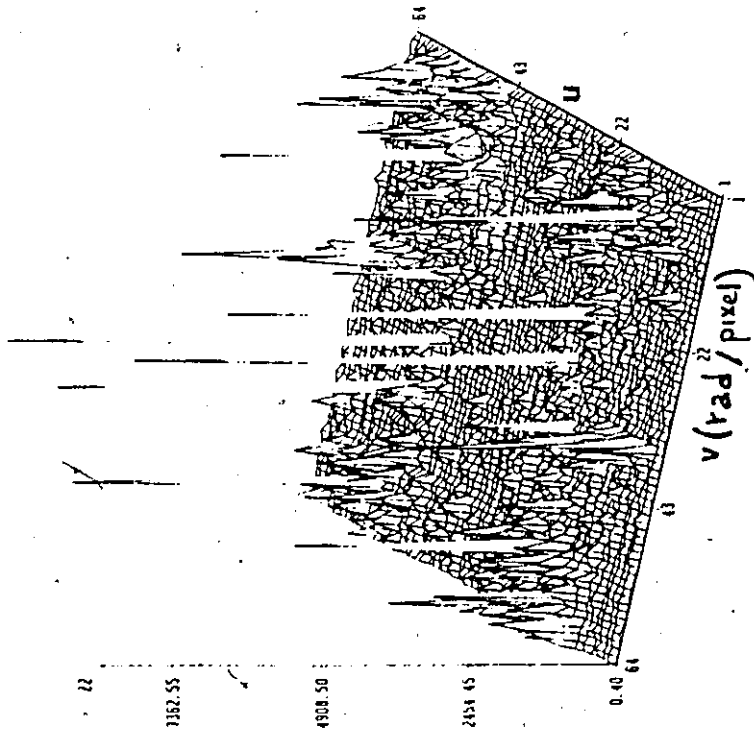


FIG. 3.6. b) PHASE OF $H(u, v)$
CORRESPONDING TO FIG. (3.4).

3.6 CONCLUSION.

The material developed in the previous sections is just an introduction. We considered only the problem from the point where a degraded digital image is given; thus topics dealing with sensor, digitizer and display were not considered in this chapter. These subjects, although of importance are outside the mainstream of our present discussion. For more details refer to the bibliography cited at the end of this thesis.

Chapter IV RESTORATION.

4.1 INTRODUCTION.

Millions of images are created every day. Most of them are of very fine quality. Some of them are of lesser quality; and for these which are of lesser quality it is feasible to consider the techniques by which the degrading phenomena may be removed. This is digital image restoration.

The problem of image restoration is the determination of the original object distribution 'f' given the recorded image 'g', and a knowledge about the point-spread function. The problem of the transformation or operator 'T' that maps 'f' into 'g' can be posed as

$$T \{ f \} \longrightarrow g \quad [4.1]$$

where for images we have

$$T \{ f \} = \iint_{-\infty}^{+\infty} h(x, y, \alpha, \beta) f(\alpha, \beta) d\alpha d\beta \quad [4.2]$$

The problem of image restoration is then to find the inverse transformation T^{-1} such that

$$T^{-1} \{ g \} \longrightarrow f \quad [4.3]$$

In a mathematical sense, the problem of image restoration corresponds to the existence and uniqueness of an inverse transformation.

By the end of this chapter a detailed view on the different approaches utilized in our work will be given. Also the results of applying the proposed algorithms, on simulated blur (by computer simulation), as well as, on real blurred images (by camera) will be provided.

We will start by giving an overview of some spatial image restoration techniques, and precisely inverse filtering, and Wiener filter.

4.2 ALGEBRAIC SPATIAL IMAGE RESTORATION TECHNIQUES.

One of the most common image restoration tasks is that of spatial image restoration to remove distortion introduced by an imaging system. Algebraic methods of spatial image restoration are analyzed in this section.

There are two ways of going about image restoration. The first way is called priori restoration, the assumption made is that the impulse response function of the blurring system is known. However in many practical situations one would not have knowledge of the PSF, hence the second way in image restoration which is called posteriori restoration. References [5-17],[35-41] contain surveys of image restoration methods.

4.2.1 Spatial filtering methods.

For the class of imaging systems in which the spatial degradation can be modeled by a linear-shift-invariant impulse response and where the noise is additive, the restoration can be performed by linear filtering techniques. These methods are described in the following sections.

4.2.1.1 INVERSE FILTERING.

The earliest attempts toward image restoration were based on the concept of inverse filtering in which the degrading transfer function is inverted to yield a restored image. Fig. (4.1) contains a block diagram for the analysis of inverse filtering.

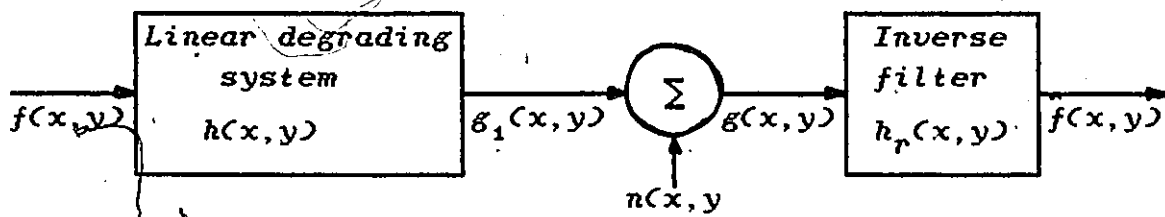


FIG. 4.1 INVERSE FILTERING IMAGE RESTORATION.

An ideal image $f(x,y)$ passes through a linear degrading system with impulse response $h(x,y)$, a noise $n(x,y)$ is added. The degraded image $g(x,y)$ can be represented by the convolution operation as (refer to chapter III)

$$g(x,y) = \iint_{-\infty}^{+\infty} f(x,y)h(x-\alpha,y-\beta)d\alpha d\beta + n(x,y) \quad [4.4a]$$

or

$$g(x,y) = f(x,y) * h(x,y) + n(x,y) \quad [4.4b]$$

The restoration system consists of a linear-shift invariant filter described by the impulse response $h_r(x,y)$. With this filter the restored image becomes

$$\hat{f}(x,y) = [f(x,y) * h(x,y) + n(x,y)] * h_r(x,y) \quad [4.5]$$

By the Fourier transform convolution theorem we will end up with:

$$\hat{F}(u,v) = [F(u,v) H(u,v) + N(u,v)] H_r(u,v) \quad [4.6]$$

where the capital letters represent the two-dimensional Fourier transforms of their respective small ones.

Now if the restoration filter transfer function $H_r(u,v)$ is chosen so that

$$H_r(u,v) = \frac{1}{H(u,v)} \quad [4.7]$$

Then from Eqn. (4.6) we will have

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \quad [4.8]$$

In the absence of noise, a perfect reconstruction results, but if noise is present we will get into trouble where $H(u,v)$ becomes small. There have been several proposals to take care of this problem inherent to inverse filtering. One of them [5] consist in choosing a restoration filter with a transfer function.

$$H_r(u,v) = \frac{H_k(u,v)}{H_d(u,v)} \quad [4.9]$$

Where $H_k(u,v)$ has a unity value over the range of (u,v) where the signal is greater than the noise and zero elsewhere.

4.2.1.2 WIENER FILTER.

Since inverse filtering performs poorly in the presence of noise. An improved restoration quality is possible with Wiener filtering techniques. The impulse response of the Wiener filter is chosen in such a way as to minimize the mean-square restoration error. The optimum filter has its transfer function given by:

$$H_r(u, v) = \frac{H^*(u, v) \phi_f(u, v)}{|H(u, v)|^2 \phi_f(u, v) + \phi_n(u, v)} \quad [4.10a]$$

or

$$H_r(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K} \quad [4.10b]$$

where H^* is the complex conjugate of H

ϕ_f and ϕ_n are the power spectral density of the ideal image 'f' and the noise 'n'.

$$\text{and } K = \frac{\phi_n}{\phi_f}$$

In the absence of noise (i.e., $\phi_n = 0$) Wiener filtering becomes inverse filtering.

4.2.1.3 PRESENTATION OF THE WORK AND RESULTS.

The purpose of this subsection is to restore a given blurred image by using inverse and Wiener filtering techniques.

First $H(u,v)$ the optical transfer function of the blurring system should be found. This was done by using the same approach described in reference [21] in page 200.

A white cardboard with a black circle in the middle formed the image $f(x,y)$, which was digitized and stored (Fig.(4.2a)). The camera was then placed out-of focus, and a new image $g(x,y)$ of the blurred black circle, was digitized and stored (Fig.(4.2b)).

The optical transfer function $H(u,v)$ can be approximated by:

$$H(u,v) = \frac{G(u,v)}{F(u,v)} \quad [4.11]$$

where $G(u,v)$: Fourier transform of $g(x,y)$.

$F(u,v)$: Fourier transform of $f(x,y)$.

To test inverse filtering a face (TIMKENT) was placed in front of the camera at the same distance, as that of the cardboard, while the camera is still out-of focus. The degraded image obtained is shown in Fig.(4.3a).

Since $H(u,v)$ has been determined, the restored image could then be obtained by using Eqn.(4.8). The result obtained is shown in Fig.(4.3b). While the restored image obtained using Wiener filter (Eqn.(4.10b)), is shown in Fig.(4.3c)

4.2.1.4 COMMENTS AND CONCLUSION.

In examining the results, obtained which are shown in Fig.(4.3). It can be commented that the Wiener filter performs better than the inverse filter. The inverse filter technique is very sensitive to the noise.

It is clear that neither of the above two techniques yielded satisfactory results for the following reasons:

- 1- The camera system is space variant as concluded in section 3.5.
- 2- Both types of filtering techniques are prone to noise.

In conclusion, the work done in this subsection was based on a deconvolution process via the Fourier transform technique. This method is limited by speed, memory requirements, and end results.

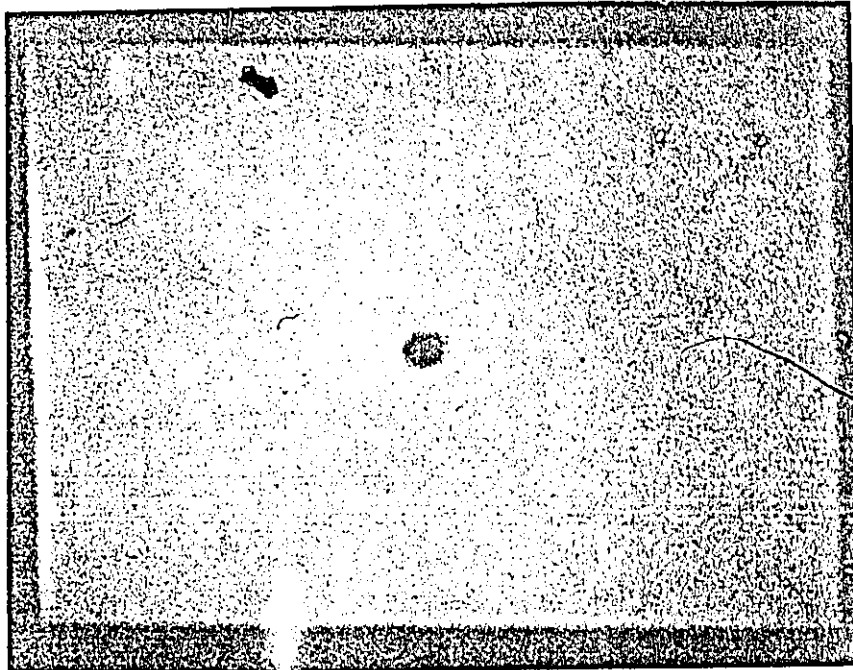


FIG. 4.2 a) ORIGINAL IMAGE OF A POINT SOURCE.

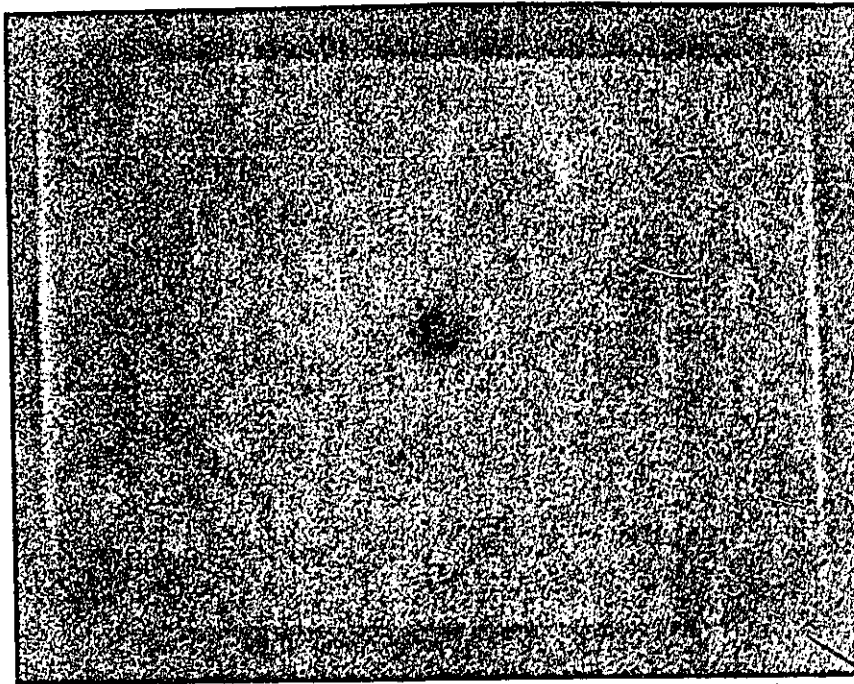


FIG. 4.2 b) BLURRED VERSION OF THE POINT SOURCE.

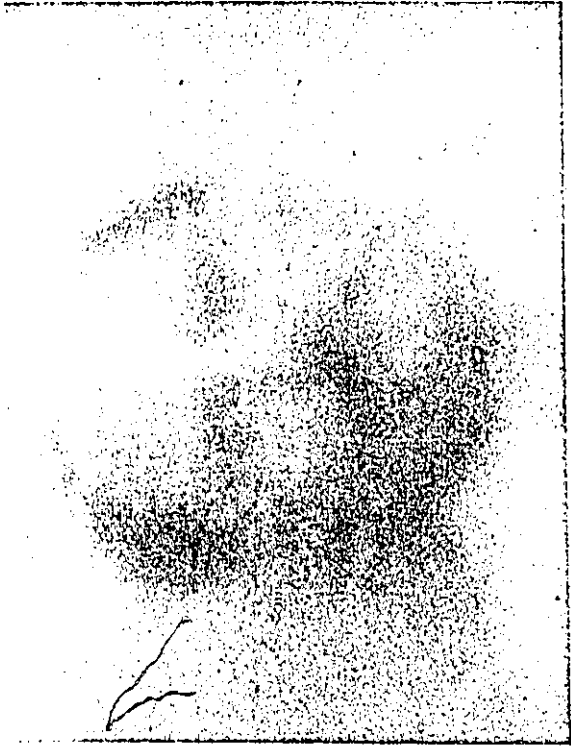


FIG. 4.3 a) DEGRADED IMAGE TO BE RESTORED.

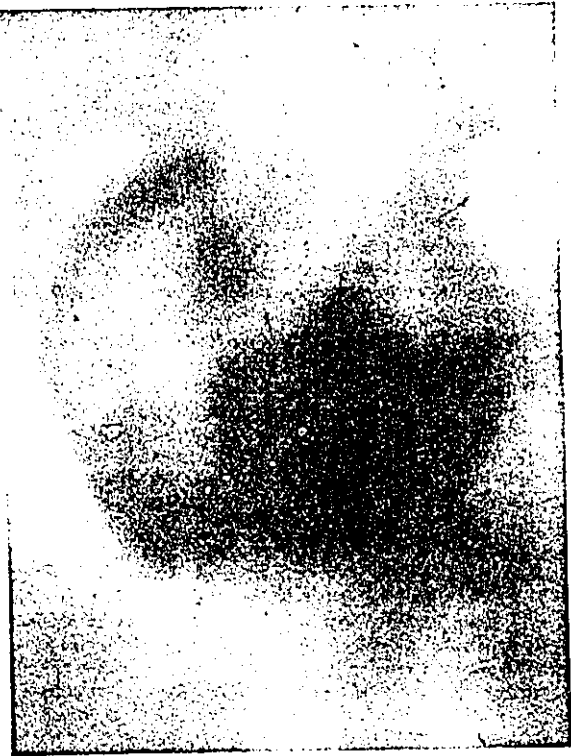


FIG. 4.3 b) RESTORED IMAGE BY INVERSE FILTERING. FIG. 4.3 c) RESTORED IMAGE BY WIENER FILTERING.

4.3 PROBLEM FORMULATION AND SYNTHESIS.

4.3.1 Problem formulation.

The input-output relationship of an imaging system (if $h(x,y)$ is position invariant) is given by:

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \quad [4.12]$$

Where

g is the recorded or degraded image.

h is the point-spread function of the imaging system.

f is the ideal image.

n is a random 2-D noise.

$*$ is the 2-D convolution.

Eqn.(4.12) reduces to

$$g(x,y) = \iint_{-\infty}^{+\infty} h(x-\alpha, y-\beta) f(\alpha, \beta) d\alpha d\beta + n(x,y) \quad [4.13]$$

The discret approximation of Eqn(4.13) is given by:

$$g(j,k) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} h(j-m, k-n) f(m,n) + n(j,k) \quad [4.14]$$

It can be verified directly that Eqn.(4.14) is represented as the vector matrix equation [9]:

$$g = H f + n \quad [4.15]$$

The difficulty of solving Eqn.(4.15) directly is evident in the extremely large size of the linear system (for an image of size 128*128, the system of equations resulting from Eqn.(4.15) is of order 16384). Thus the use of least-squares technique estimation.

The z-transform of Eqn.(4.12) when neglecting the noise term ($n(x,y) = 0$) is given by:

$$G(z_1, z_2) = H(z_1, z_2) F(z_1, z_2) \quad [4.16]$$

We would like to be able to synthesize a two-dimensional recursive filter to perform the restoration operation

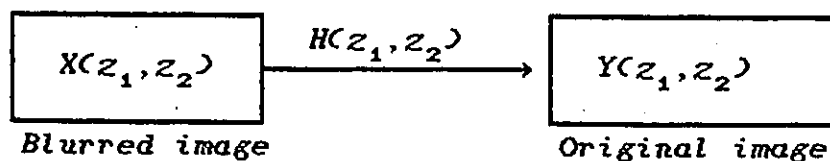
4.3.2 Synthesis.

In this subsection we will present a synthesis technique in the spatial-domain.

two cases are considered:

4.3.2.1.FIRST CASE: BLURRED + ORIGINAL

AND BLURRED + SIMILAR.



Let $H(z_1, z_2)$ be the desired impulse response (PSF) from the block diagram given above we have:

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} \quad [4.17]$$

Let the approximating two-dimensional recursive filter transfer function be:

$$F(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)} = \frac{\sum_{i=0}^N \sum_{j=0}^N a_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=0}^N \sum_{j=0}^N b_{ij} z_1^{-i} z_2^{-j}} \quad [4.18]$$

$$b_{00} = 1.0$$

where N is the order of the IIR filter

We want to approximate the PSF by a two-dimensional recursive filter

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{\sum_{i=0}^N \sum_{j=0}^N a_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=0}^N \sum_{j=0}^N b_{ij} z_1^{-i} z_2^{-j}} \quad [4.19]$$

$$b_{00} = 1.0$$

The time domain synthesis, is to find the $2(N+1)^2-1$ coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$ such that $f(i, j)$ approximates $h(i, j)$ over the range $i, j = 0, 1, \dots, N$.

From Eqn. (4.19) we have

$$Y(z_1, z_2) \sum_{i=0}^N \sum_{j=0}^N b_{ij} z_1^{-i} z_2^{-j} = X(z_1, z_2) \sum_{i=0}^N \sum_{j=0}^N a_{ij} z_1^{-i} z_2^{-j}$$

or

$$\sum_{i=0}^N \sum_{j=0}^N b_{ij} y(m-i, n-j) = \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i, n-j) \quad [4.20]$$

for $m, n = 0, 1, \dots, M$

Since $b_{00} = 1.0$, Eqn. (4.20) implies

$$y(m, n) + \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} y(m-i, n-j) = \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i, n-j) \quad [4.21]$$

for $m, n = 0, 1, \dots, M$

For an exact recursive realization [42]: $M = 2*(N+1)^2 - 1$

We want to find the coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$ in such a way as to approximate 'y' by 'y_d' (desired output), and such as to minimize the total squared error.

Let $y(m, n) \approx y_d(m, n) \quad \forall m, n$

Where

$y_d(m, n)$ is the desired output image.

$y(m, n)$ is the actual o/p of the transfer function.

$x(m, n)$ is the blurred image.

Hence from Eqn.(4.21) we have:

$$y_d(m,n) + \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} y_d(m-i,n-j) \approx \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i,n-j) \quad [4.22]$$

for $m,n = 0,1,\dots,M$

Introducing the error $e(m,n)$, that we shall add to the right hand side of Eqn.(4.22) to produce the equality

$$y_d(m,n) + \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} y_d(m-i,n-j) = e(m,n) + \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i,n-j) \quad [4.23]$$

for $m,n = 0,1,\dots,M$

or

$$e(m,n) = y_d(m,n) + \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} y_d(m-i,n-j) - \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i,n-j) \quad [4.24]$$

for $m,n = 0,1,\dots,M$

The total squared error is equal to:

$$E = \sum_{m=0}^M \sum_{n=0}^M \left\{ e(m,n) \right\}^2$$

The total squared error ,E, can be minimized by setting its partial derivatives with respect to the parameters $\{a_{ij}\}$ and $\{b_{ij}\}$ equal to zero.

Hence

$$\left\{ \begin{array}{l} \frac{\partial E}{\partial a_{lk}} = 2 \sum_{m=0}^M \sum_{n=0}^M e(m,n) \frac{\partial e(m,n)}{\partial a_{lk}} = 0; \text{ for } l,k = 0,1,\dots,N \quad [4.25] \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial E}{\partial b_{lk}} = 2 \sum_{m=0}^M \sum_{n=0}^M e(m,n) \frac{\partial e(m,n)}{\partial b_{lk}} = 0; \text{ for } l,k = 0,1,\dots,N \quad [4.26] \\ \text{but } l+k \neq 0 \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial e(m,n)}{\partial a_{lk}} = -x(m-l, n-k) \quad \text{for } l,k = 0,1,\dots,N \quad [4.27] \\ \text{and } m,n = 0,1,\dots,M \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial e(m,n)}{\partial b_{lk}} = y_d(m-l, n-k) \quad \text{for } l,k = 0,1,\dots,N; \quad l+k=0 \quad [4.28] \\ \text{and } m,n = 0,1,\dots,M \end{array} \right.$$

By utilizing Eqns. (4.25) to (4.28) we will end up with:

$$\sum_{m=0}^M \sum_{n=0}^M \left\{ \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i, n-j) x(m-l, n-k) \right\} -$$

$$\sum_{m=0}^M \sum_{n=0}^M \left\{ \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} y_d(m-i, n-j) x(m-l, n-k) \right\} =$$

$$\sum_{m=0}^M \sum_{n=0}^M y_d(m, n) x(m-l, n-k) \quad [4.29]$$

for $l, k = 0, 1, \dots, N$

and

$$\sum_{m=0}^M \sum_{n=0}^M \left\{ \sum_{i=0}^N \sum_{j=0}^N a_{ij} x(m-i, n-j) y_d(m-l, n-k) \right\} -$$

$$\sum_{m=0}^M \sum_{n=0}^M \left\{ \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} y_d(m-i, n-j) y_d(m-l, n-k) \right\} =$$

$$\sum_{m=0}^M \sum_{n=0}^M y_d(m, n) y_d(m-l, n-k) \quad [4.30]$$

for $l, k = 0, 1, \dots, N$; but $l+k \neq 0$

Eqns. (4.29) and (4.30) will provide us a system of $2(N+1)^2 - 1$ linear equations which have to be solved to get the coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$.

4.3.2.2 SECOND CASE: BLURRED IMAGE ONLY.

From Eqn. (4.19) we have

$$H(z_1, z_2) = \frac{\sum_{i=0}^N \sum_{j=0}^N a_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=0}^N \sum_{j=0}^N b_{ij} z_1^{-i} z_2^{-j}} \quad [4.31]$$

$b_{00} = 1.0$

To find the coefficients of the two-dimensional IIR filter we will use Shanks method [20] described by the following steps:

From Eqn. (4.31) we have

$$H(z_1, z_2) \sum_{i=0}^N \sum_{j=0}^N b_{ij} z_1^{-i} z_2^{-j} = \sum_{i=0}^N \sum_{j=0}^N a_{ij} z_1^{-i} z_2^{-j} \quad [4.32]$$

or

$$\sum_{i=0}^N \sum_{j=0}^N b_{ij} h(m-i, n-j) = a_{mn} \quad [4.33]$$

for $m, n = 0, 1, \dots, M$

where

N : order of the filter.

M : size of the image.

Since $a_{mn} = 0$ outside the square $m, n = 0, 1, \dots, N$ (region R defined in Fig. (4.4)).

Hence Eqn. (4.33) becomes:

$$\sum_{i=0}^N \sum_{j=0}^N b_{ij} h(m-i, n-j) = 0 \quad [4.34]$$

for $m > N$ or $n > N$

or we have $b_{00} = 1.0$; Hence Eqn. (4.34) give us:

$$h(m, n) + \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h(m-i, n-j) = 0 \quad [4.35]$$

for $(m, n) \in R$

or

$$h(m, n) = - \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h(m-i, n-j) \quad [4.36]$$

for $(m, n) \in R$

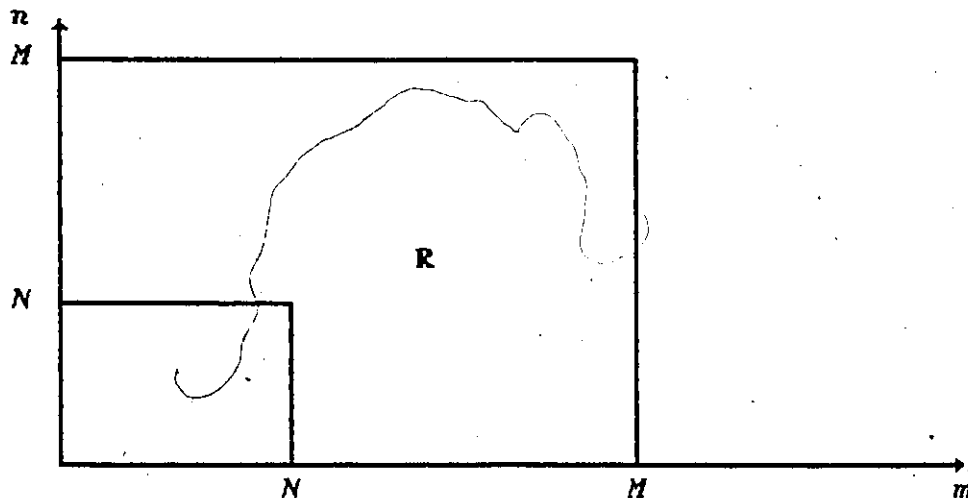


FIG. 4.4 REGION R OVER WHICH $a_{mn} = 0$.

We have to find the denominator coefficients $\{b_{ij}\}$ of the filter, in a way that $h(m,n)$ will approximate the desired impulse response $h_d(m,n)$, and such as to minimize the total squared error.

Hence

$$h_d(m,n) \approx - \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h_d(m-i, n-j) \quad [4.37]$$

for m, n over R

Introducing the error signal $e(m,n)$.

$$h_d(m,n) = e(m,n) - \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h_d(m-i, n-j) \quad [4.38]$$

$m > N$ or $n > N$

or

$$e(m,n) = h_d(m,n) + \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h_d(m-i, n-j) \quad [4.39]$$

for $m > N$ or $n > N$.

Let E be the total squared error

$$E = \sum_{(m,n) \in R} \sum \left\{ e(m,n) \right\}^2 \quad [4.40a]$$

$$E = \sum_{(m,n) \in R} \sum \left\{ \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h_d(m-i, n-j) + h_d(m,n) \right\}^2 \quad [4.40b]$$

If we differentiate , E , with respect to the denominator coefficients $\{b_{ij}\}$, remembering ($b_{00} = 1.0$), we will get the following set of linear equations.

$$\sum_{(m,n) \in R} \sum_{(m,n) \in R} \left\{ \sum_{\substack{i=0 \\ i+j \neq 0}}^N \sum_{j=0}^N b_{ij} h_d^{(m-i, n-j)} h_d^{(m-l, n-k)} \right\} =$$

$$\sum_{(m,n) \in R} \sum_{(m,n) \in R} h_d^{(m,n)} h_d^{(m-l, n-k)} \quad [4.41]$$

for $l, k = 0, 1, \dots, N$; but $l+k \neq 0$

Solving Eqn.(4.41) we will find the denominator coefficients $\{b_{ij}\}$. To compute the numerator coefficients $\{a_{ij}\}$, we have to use Eqn.(4.33)

$$a_{mn} = \sum_{i=0}^N \sum_{j=0}^N b_{ij} h_d^{(m-i, n-j)} \quad [4.42]$$

for $m, n = 0, 1, \dots, N$.

Before being able to compute the coefficients $\{b_{ij}\}$ and $\{a_{ij}\}$ from the recursions given in Eqns.(4.41) and (4.42), respectively, we have to find the impulse response of the imaging system (PSF) from the available image.

The only assumption made is that the PSF is approximated by a two-dimensional Gaussian.

$$G(r, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{r^2}{2\sigma^2} \right) \quad [4.43]$$

Where

σ : spatial constant.

r : radial distance.

The only unknown in Eqn.(4.43) is ' σ ', the point-spread function spatial parameter (i.e., the radius of the imaged point's "blur circle").

Measurement of ' σ ' has been determined using a similar approach to that of Pentland [19] with some modifications. The different steps to measure ' σ ' are described by the following.

-Measurement of σ :

The measurement of ' σ ' presents a problem, because ' σ ' itself is function on both the characteristics of the scene and those of the lens system. To eliminate the first effect we have to look for places in the image with known characteristics, like edges or isolated points, the data surrounding them can be used to determine ' σ '.

An estimate for ' σ ' can be obtained from a degraded edge. We will use this approach to find ' σ '.

We define $C(x,y)$ to be the gradient of the given (blurred) image to detect all the edges.

From [19] we have

$$C(x,y) = \nabla [G(r,\sigma) * I(x,y)]$$

Where $I(x,y)$ is a vertical step edge in the image of magnitude δ

$$C(x,y) = \iint_{-\infty}^{+\infty} \nabla G\left(\sqrt{(x-u)^2 + (y-v)^2}, \sigma\right) I(u,v) du dv \quad [4.44]$$

or

$$C(x,y) = \delta G(x,\sigma) = \frac{\delta}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2}{2\sigma^2}\right] \quad [4.45]$$

Taking the absolute value, then the natural logarithm, we will end up with:

$$\ln\left[\frac{\delta}{\sqrt{2\pi\sigma^2}}\right] - \frac{x^2}{2\sigma^2} = \ln |C(x,y)| \quad [4.46]$$

$$\text{We can reformulate Eqn. (4.46) as: } Ax^2 + B = C \quad [4.47]$$

Where

$$A = \frac{-1}{2\sigma^2}; \quad B = \ln\left[\frac{\delta}{\sqrt{2\pi\sigma^2}}\right]; \quad \text{and } C = \ln |C(x,y)|$$

From Eqn. (4.47), we can estimate A, and thus obtain σ

The solution of the linear regression given in Eqn. (4.47) is:

$$A = \frac{\sum_{i=0}^N (x_i^2 - \bar{x}^2) C_i}{\sum_{i=0}^N (x_i^2 - \bar{x}^2) x_i^2} ; \quad B = \bar{C} - \bar{x}^2 A \quad [4.48]$$

Where

$$\bar{x}^2 = \frac{1}{N} \sum_{i=0}^N x_i^2$$

and N: extent of the blur

An estimate for 'σ' is:

$$\sigma = \frac{1}{\sqrt{-2A}} \quad [4.49]$$

After determining 'σ', the impulse response of the imaging system is fully determined. Now we can calculate $\{b_{ij}\}$ and $\{a_{ij}\}$ the filter coefficients by using Eqns. (4.41) and (4.42).

An estimate 'f-hat' for 'f' can be found from:

$$\hat{f}(m,n) = \frac{1}{a_{00}} \left\{ \sum_{i=0}^N \sum_{j=0}^N b_{ij} g(m-i, n-j) - \sum_{\substack{i=0 \\ i+j=0}}^N \sum_{j=0}^N a_{ij} \hat{f}(m-i, n-j) \right\} \quad [4.50]$$

for $m, n = 0, 1, \dots, M$

where

\hat{f} : is the restored image.

and g : is the recorded or degraded image.

4.4 PRESENTATION OF THE WORK AND RESULTS.

In this section we will present simulations on images blurred by computer simulation and experiments on camera blurred images, by using the previously discussed methods.

4.4.1 Procedure.

4.4.1.1 FOCUSING OF IMAGES USING BLURRED+ORIGINAL PICTURES

The experiments were performed on an IBM AT personnel computer. The coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$ of the IIR filter were obtained by using the recursions given in Eqns.(4.29) and (4.30). The algorithm was tried on 4 different images with various amount of simulated blur (by computer). The images considered have a resolution of $256*256$. The coefficients were found globally (using all the image), and locally by using sub-images of $128*128$ pixels each.

Figs.(4.5), (4.6), (4.7), and (4.8) show the results obtained.

Finally we present restoration of true camera blurred images by using the same procedure as previously. The results are shown in Figs.(4.9), and (4.10).

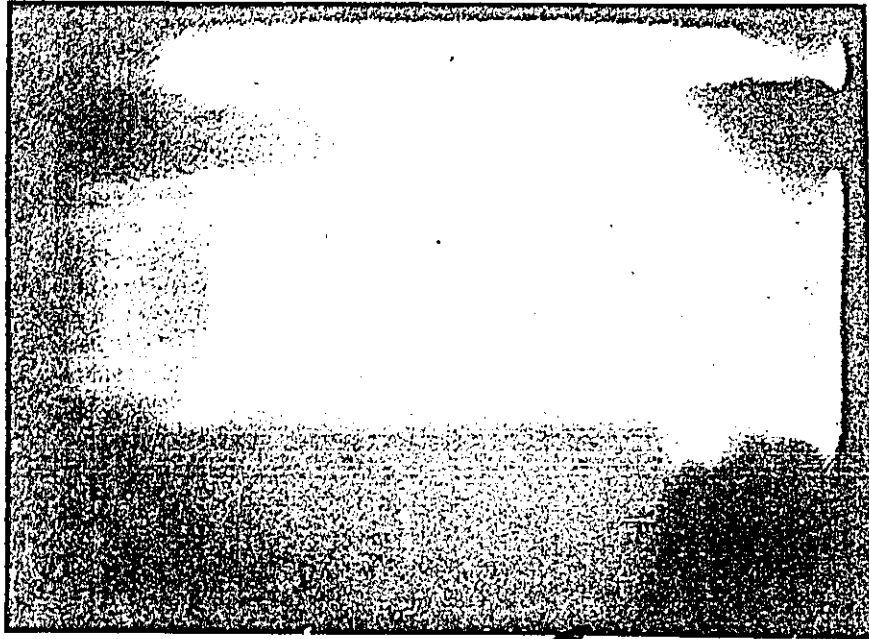


FIG. 4.5 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*

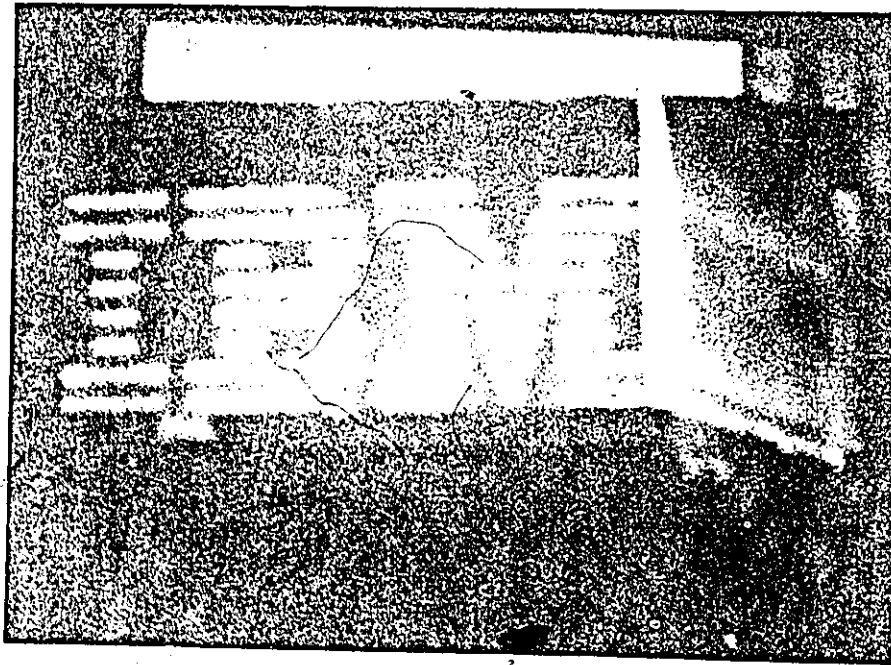


FIG. 4.5 b) *ORIGINAL IMAGE.*

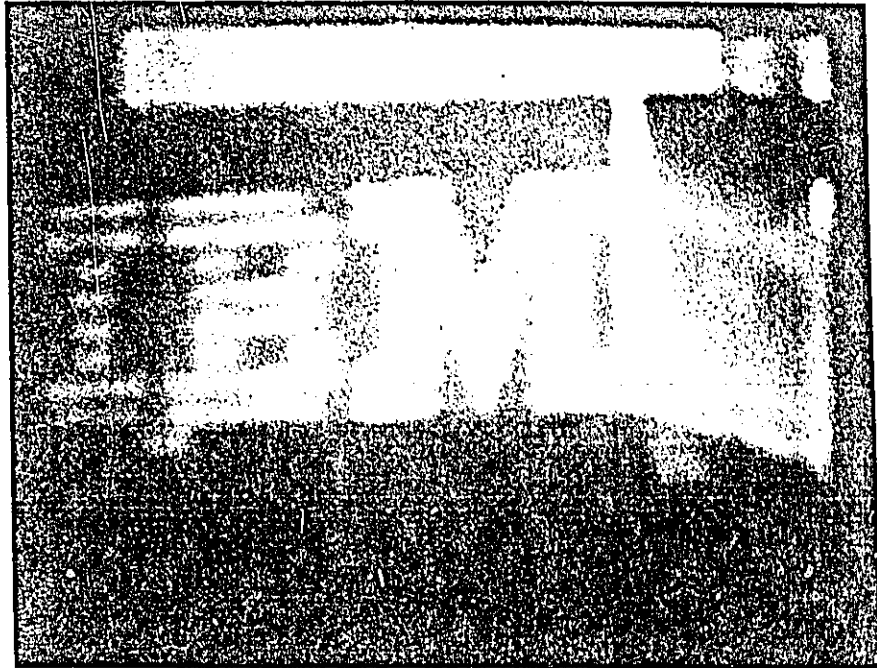


FIG. 4.5 c) RESTORED IMAGE (GLOBAL).

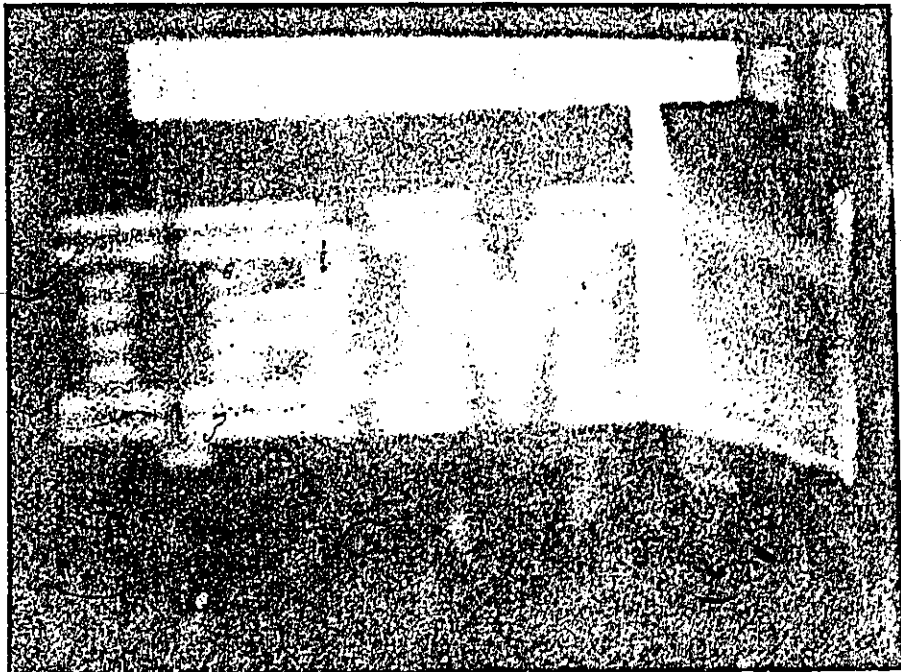


FIG. 4.5 d) RESTORED IMAGE (LOCAL).

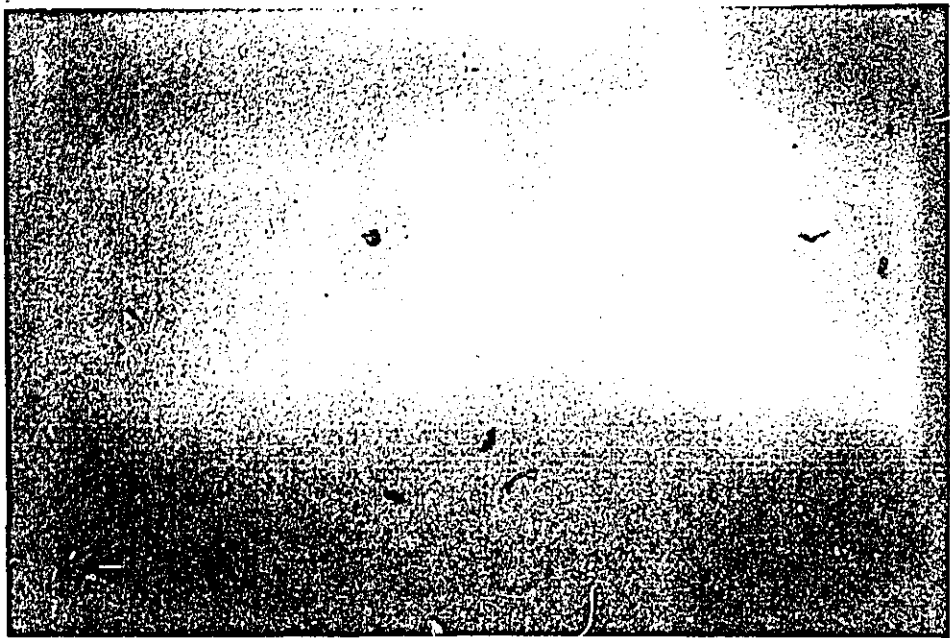


FIG. 4.6 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*

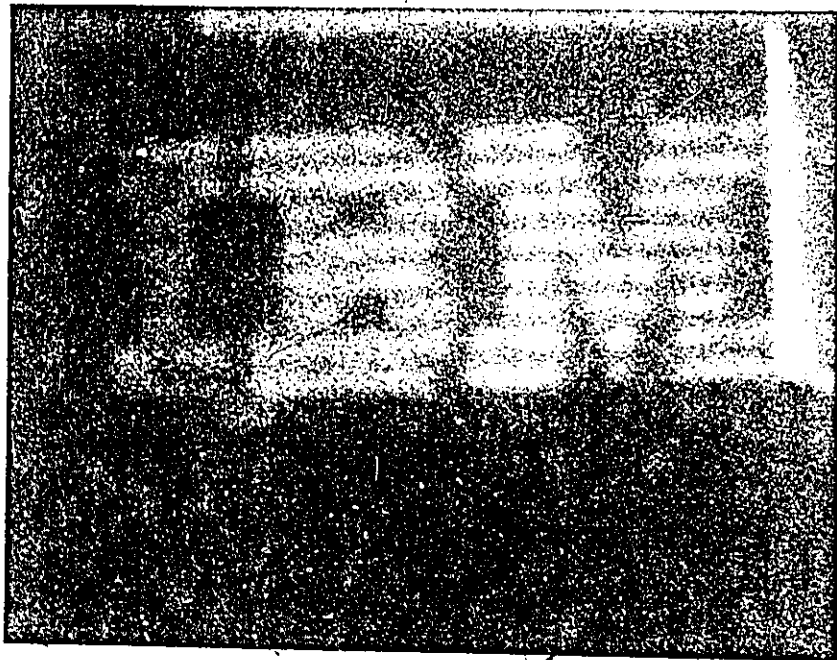


FIG. 4.6 b) *ORIGINAL IMAGE.*

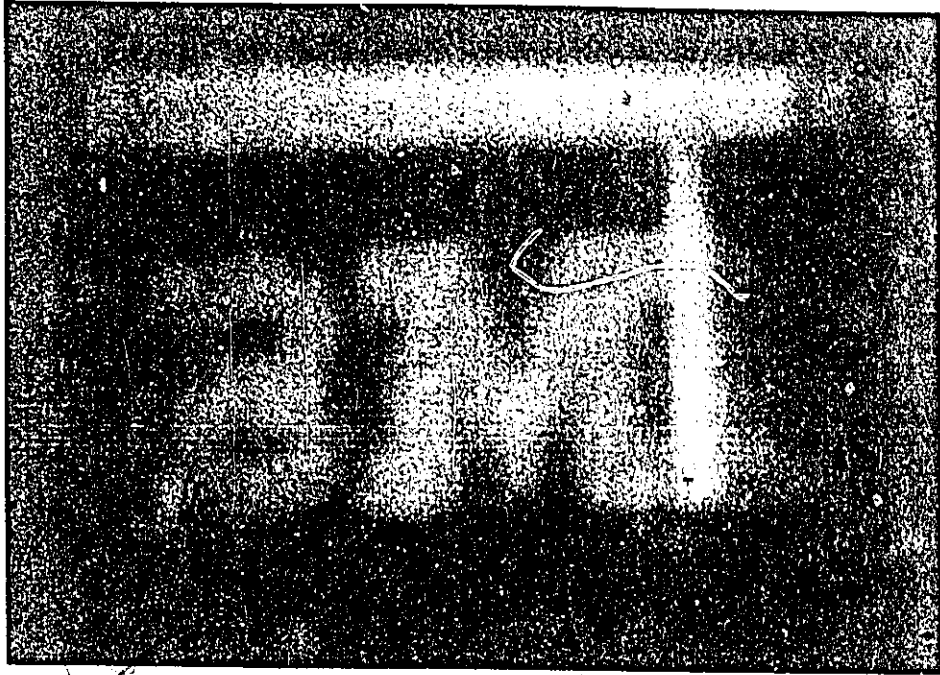


FIG. 4.6 c) RESTORED IMAGE (GLOBAL).



FIG. 4.6 d) RESTORED IMAGE (LOCAL).

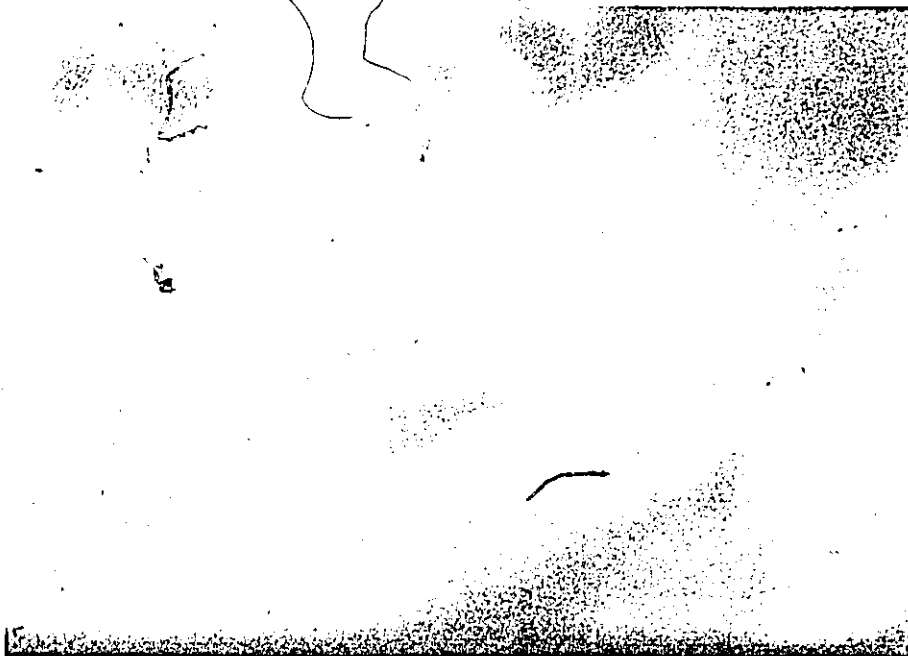


FIG. 4.7 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*



FIG. 4.7 b) *ORIGINAL IMAGE.*



FIG. 4.7 c) RESTORED IMAGE (GLOBAL).



FIG. 4.7 d) RESTORED IMAGE (LOCAL).



FIG. 4.8 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*



FIG. 4.8 b) *ORIGINAL IMAGE.*



FIG. 4.8 c) RESTORED IMAGE (GLOBAL).



FIG. 4.8 d) RESTORED IMAGE (LOCAL).

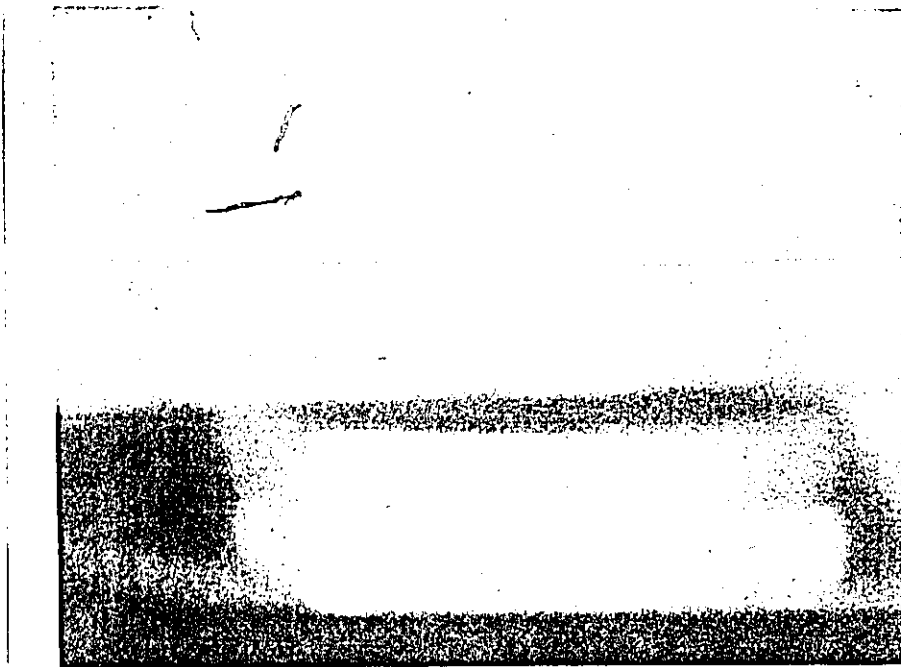


FIG. 4.9 a) *BLURRED IMAGE BY IMAGING SYSTEM (CAMERA).*

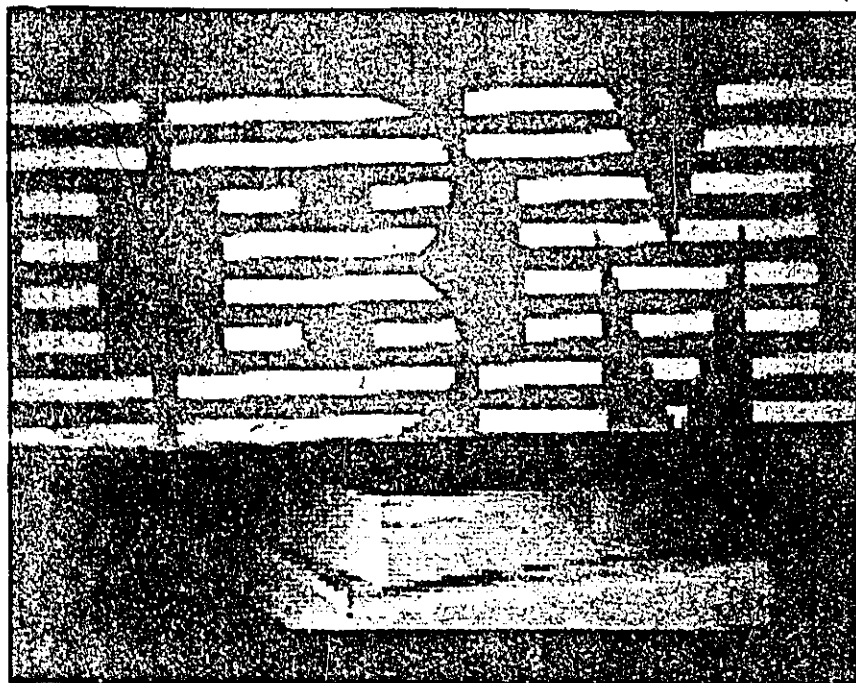


FIG. 4.9 b) *ORIGINAL IMAGE.*

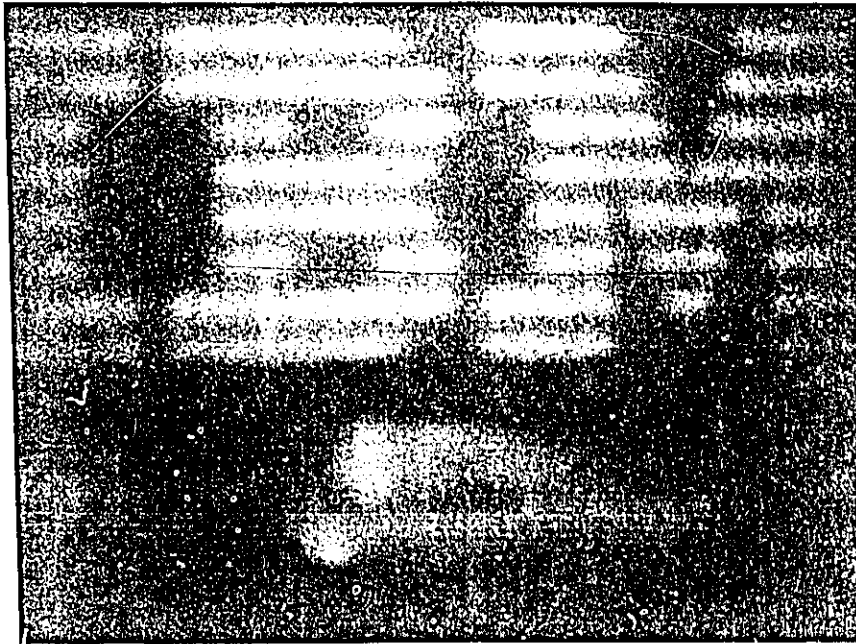


FIG. 4.9 c) RESTORED IMAGE (GLOBAL).

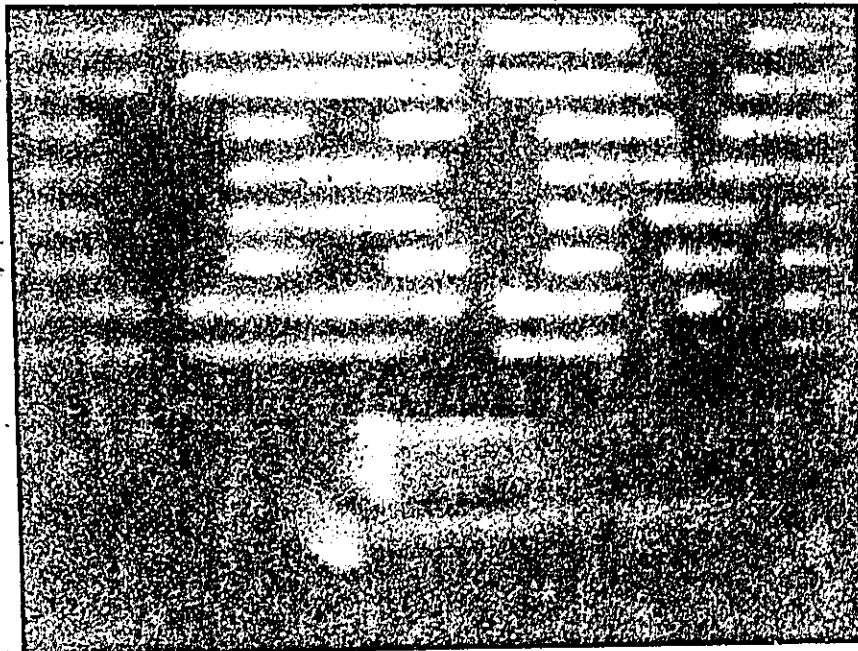


FIG. 4.9 d) RESTORED IMAGE (LOCAL).

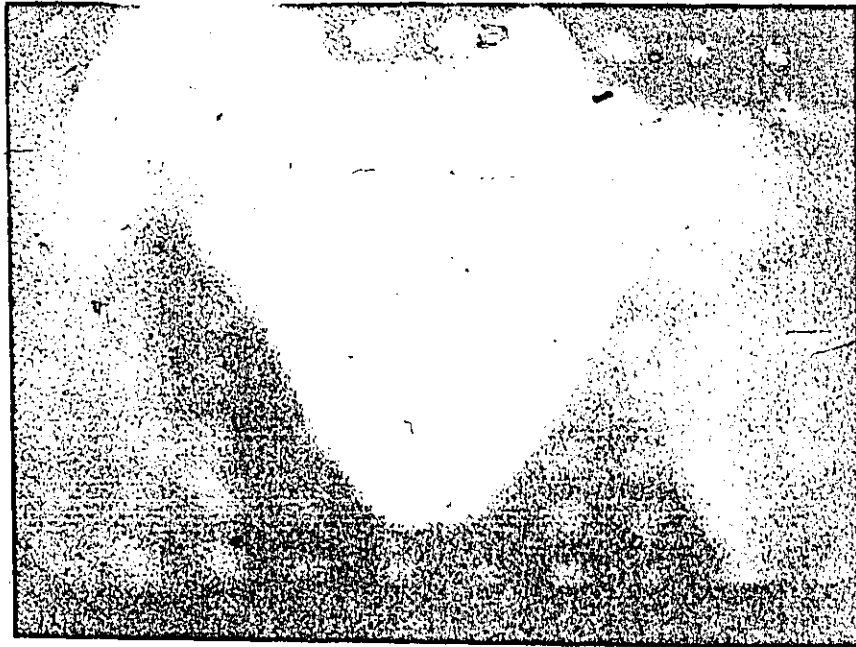


FIG. 4.10 a) *BLURRED IMAGE BY IMAGING SYSTEM (CAMERA).*

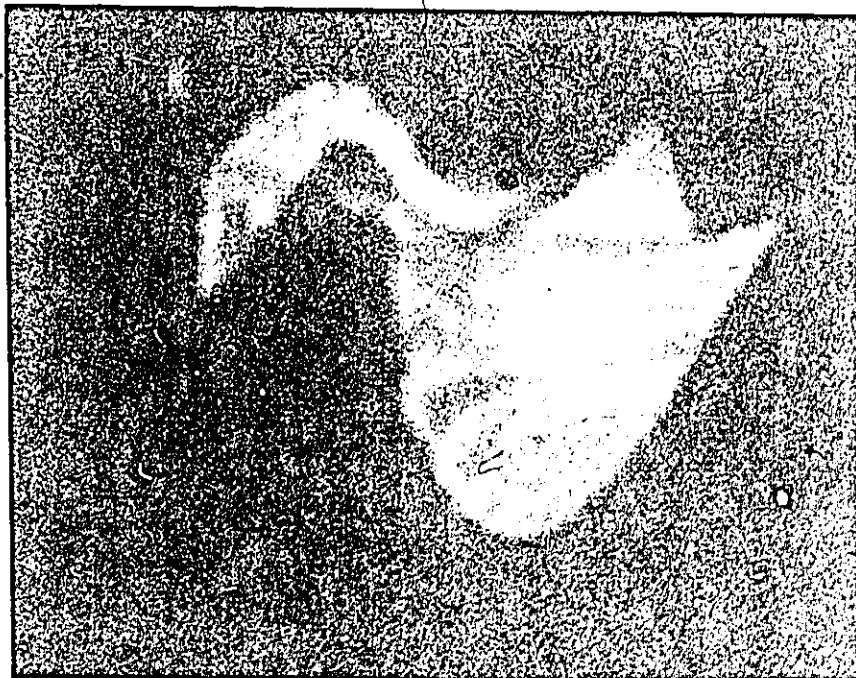


FIG. 4.10 b) *ORIGINAL IMAGE.*

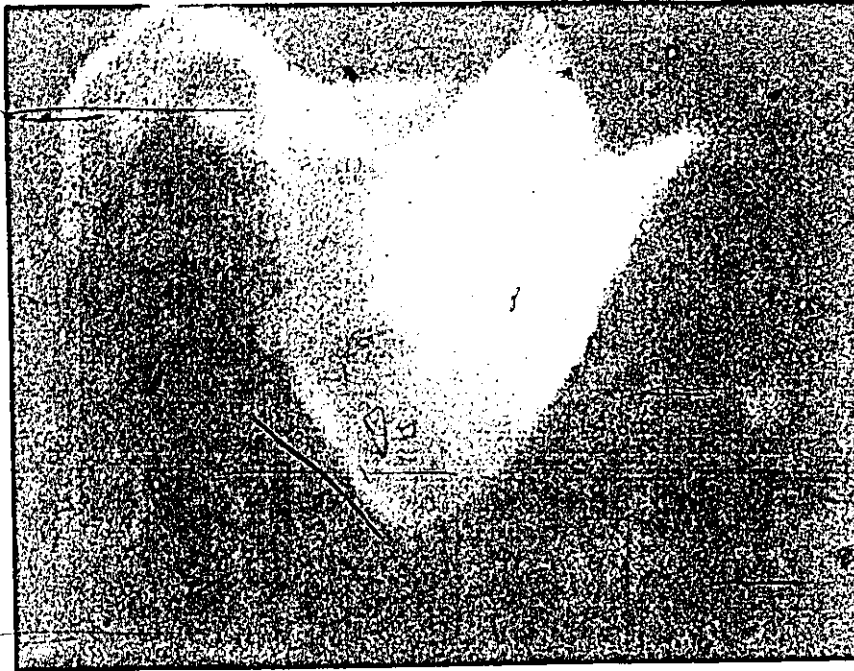


FIG. 4.10 c) RESTORED IMAGE (GLOBAL).

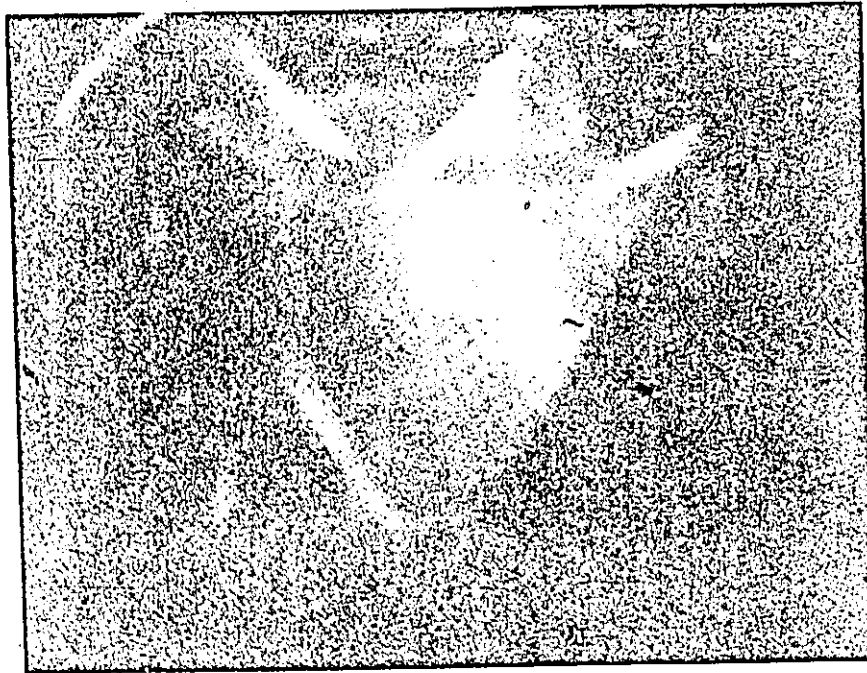


FIG. 4.10 d) RESTORED IMAGE (LOCAL).

4.4.1.2 FOCUSING OF IMAGES USING SIMILAR PICTURES.

A/ Definition of similar picture:

Two different images are statistically similar if they have similar autocorrelation function [16], i.e., $F(u,v) \cdot F^*(u+\Delta u, v+\Delta v)$ is similar for both images.

The technique of using two similar images was first applied for restoring old acoustic recordings (signal processing), and then generalized for the case of images degraded by some common form of blur (for example motion blur and out-of focus) [13]. However unlike the ear, the eye is sensitive to phase distortion. The major weakness of the approach is the inability to correct for the unknown phase distortions in the general case. Cole [43] applied with some success the approach by assuming optical transfer functions of zero phase. Cannon [14], and Morton and Andrews [16] used the same approach but calculated the phase by dividing the image into sub-images and then by assuming that windowing gives resulting differences in phases that are relatively small.

Fig.(4.11) contains four images which are statistically similar. The results of the corresponding calculation of the autocorrelation functions are given in Figs.(4.12a), (4.12b), (4.12c), and (4.12d).



FIG. 4.11 SIMILAR IMAGES.

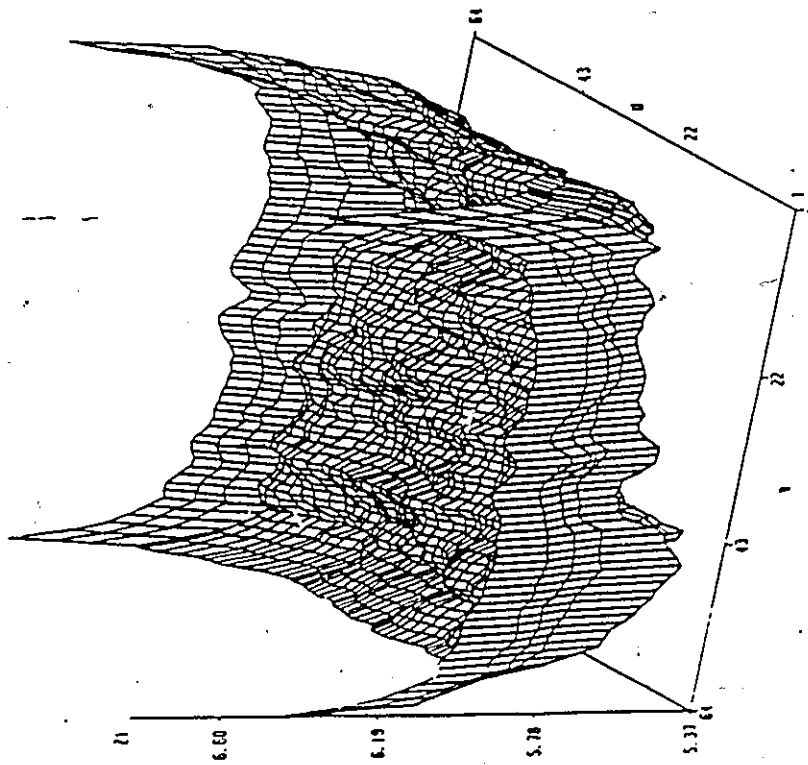


FIG. 4.12 a) AUTOCORRELATION FUNCTION
CORRESPONDING TO FIG. (4.11a)

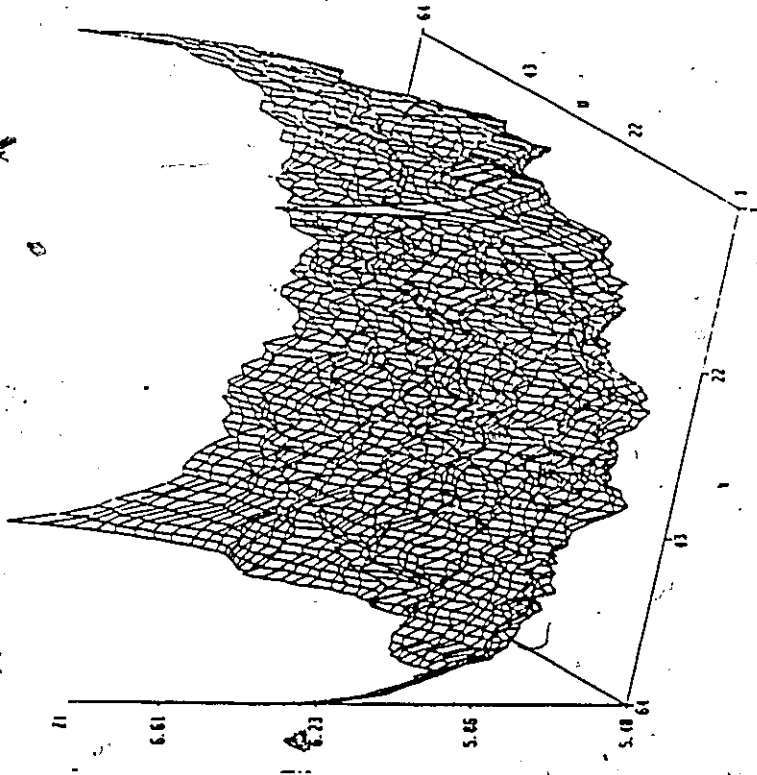


FIG. 4.12 b) AUTOCORRELATION FUNCTION
CORRESPONDING TO FIG. (4.11b)

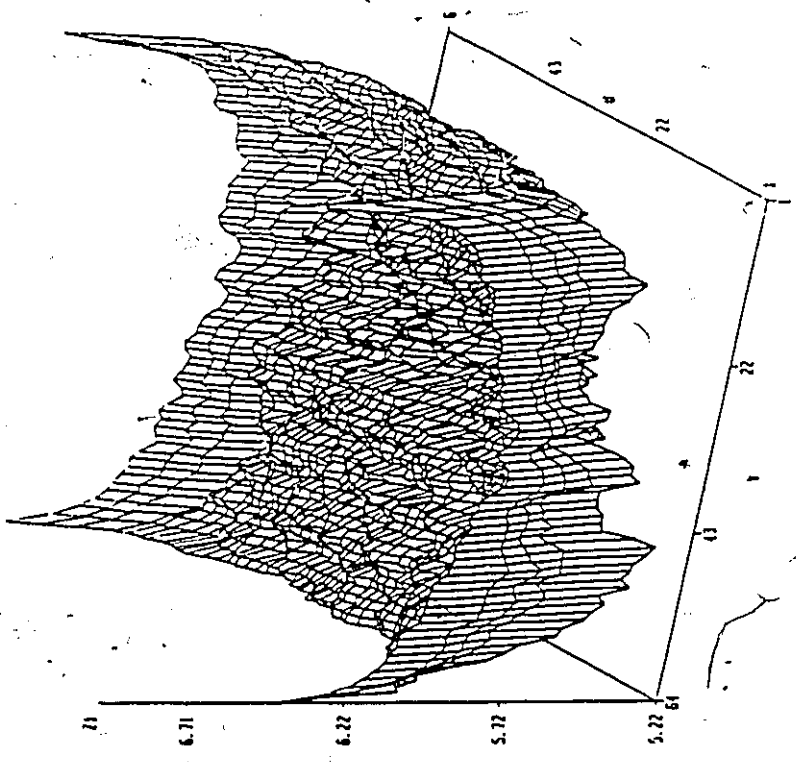


FIG. 4.12 d) AUTOCORRELATION FUNCTION
CORRESPONDING TO FIG. (4.11d)

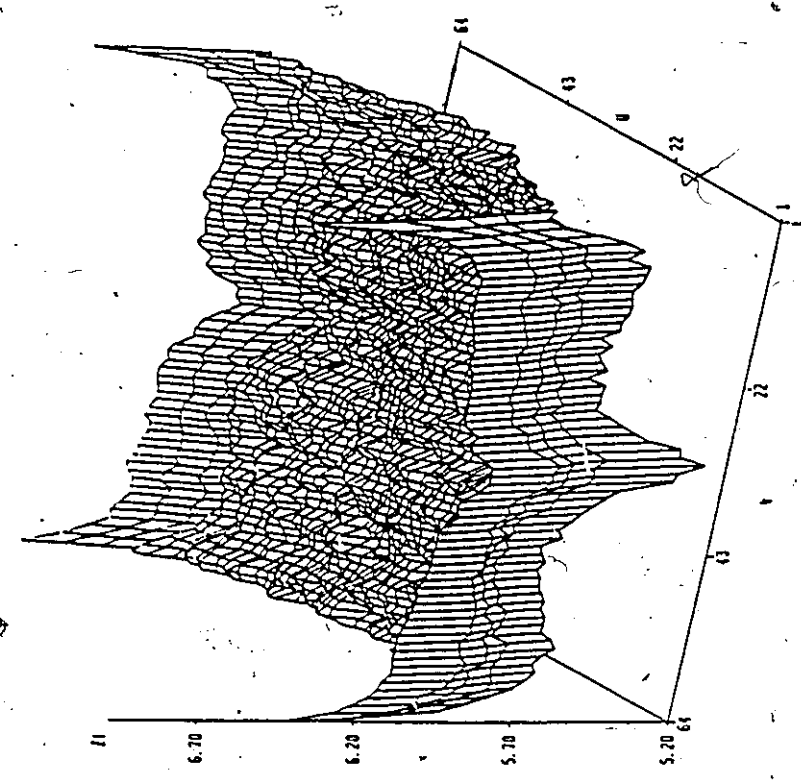


FIG. 4.12 c) AUTOCORRELATION FUNCTION
CORRESPONDING TO FIG. (4.11c)

B/ Experimental results:

A global transfer function (PSF) was obtained from two similar images (one in focus and the other out-of focus) using the recursions given in Eqns.(4.29), and (4.30). The determined PSF was applied locally by using sub-images of 128*128 pixels each.

Figs.(4.13c), and (4.14c) present the results of the restoration procedure by using a third order filter, on the images shown in Figs(4.13a), and (4.14a), and by using the image in Figs.(4.13b), and (4.14b) as the similar in focus image. The images in this experiment are of size 256*256 pixels.

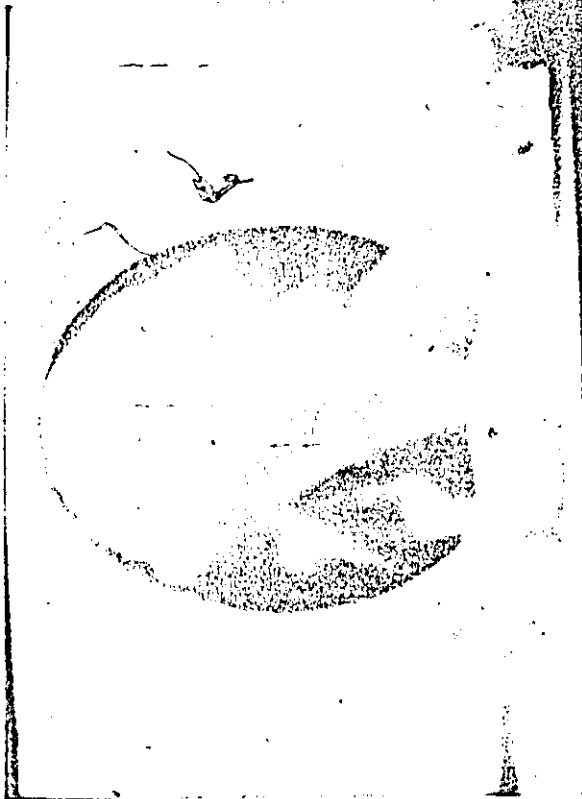


FIG. 4.13 a) BLURRED IMAGE BY CAMERA.



FIG. 4.13 c) RESTORED IMAGE (LOCAL).



FIG. 4.13 b) SIMILAR IN FOCUS IMAGE.

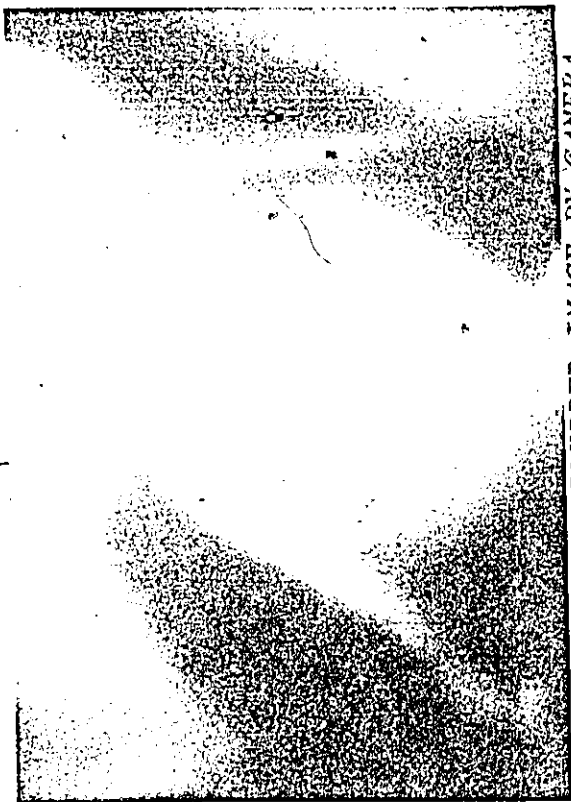


FIG. 4.14 a) BLURRED IMAGE BY CAMERA.



FIG. 4.14 c) RESTORED IMAGE (LOCAL).

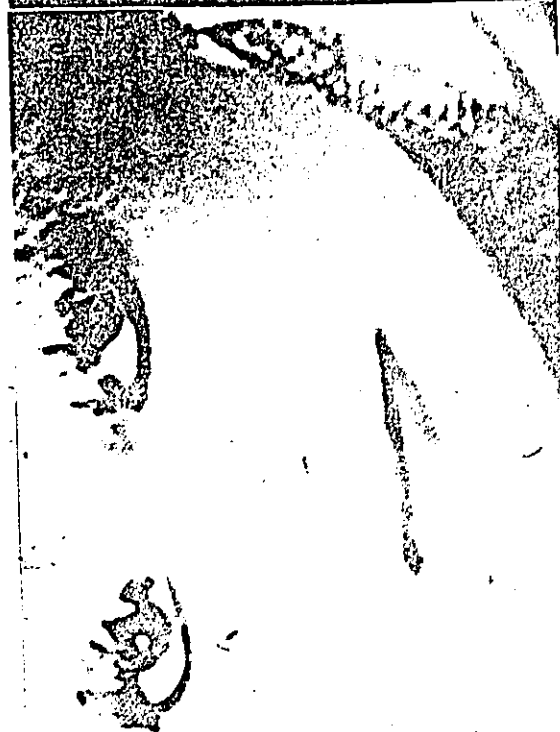


FIG. 4.14 b) SIMILAR IN FOCUS IMAGE.

4.1.3.3 FOCUSING OF IMAGES USING THE BLURRED IMAGE ONLY.

A/ ALGORITHM.

1/ Estimation of σ

Step 1: Threshold the blurred image, $g(x,y)$, (the algorithm provided in [44] was utilized). Use the thresholded image to detect the location of the edges points. The second derivative approach is not utilized in this step because of its sensitivity to noise.

Step 2: Find the gradient of the blurred image $C(x,y)$.

Step 3: By using the coordinates of the border points located in the first step, calculate N the extent of the blur in pixels.

Step 4: By using Eqn.(4.48) find 'A', and thus ' σ ' from Eqn.(4.49).

Having estimated ' σ ', the point-spread function can be determined.

2/ Calculation of the coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$ of the corresponding 2-D IIR filter.

Step 5: By the use of the recursions given in Eqns.(4.41), and (4.42) compute $\{b_{ij}\}$ and $\{a_{ij}\}$ respectively.

3/ Determination of ' \hat{f} ' an estimate for 'f' by inverse filtering.

Step 6: Compute $\hat{f}(x,y)$ an estimate for $f(x,y)$ by the use of Eqn.(4.50).

B/ APPLICATION.

The algorithm given above, was applied on various images of size $256*256$, globally, and locally by using windows of size $128*128$ with 0, and 50% overlapping.

The results of the restoration procedure for images blurred by computer simulation (simulated blur) are shown in Figs.(4.15), (4.16), (4.17), and (4.18); while Figs.(4.19), (4.20), (4.21), and (4.22) show the restoration of images blurred by the camera (real blur).

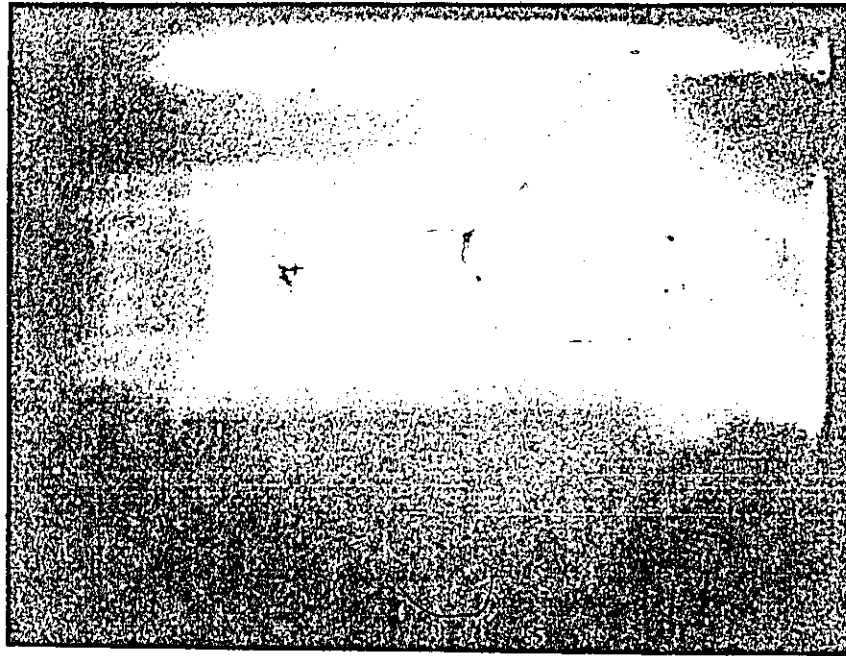


FIG. 4.15 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*

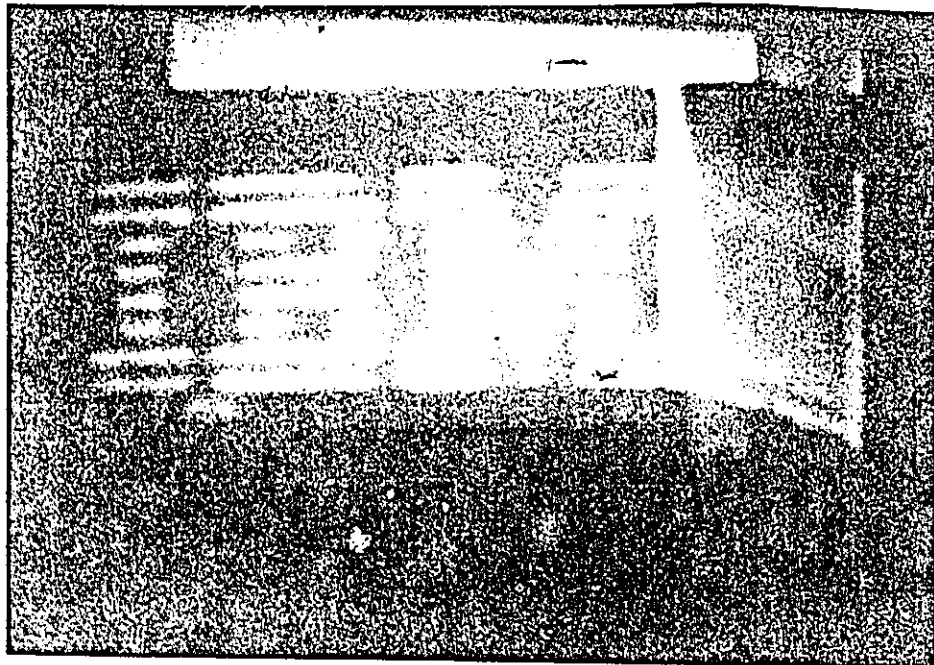


FIG. 4.15 b) *RESTORED IMAGE (GLOBAL).*

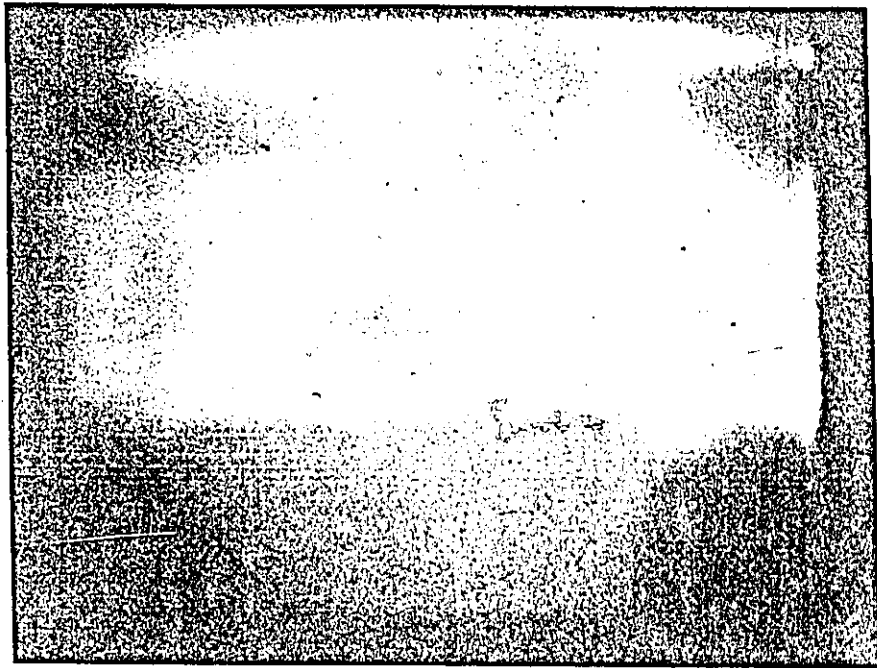


FIG. 4.16 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*

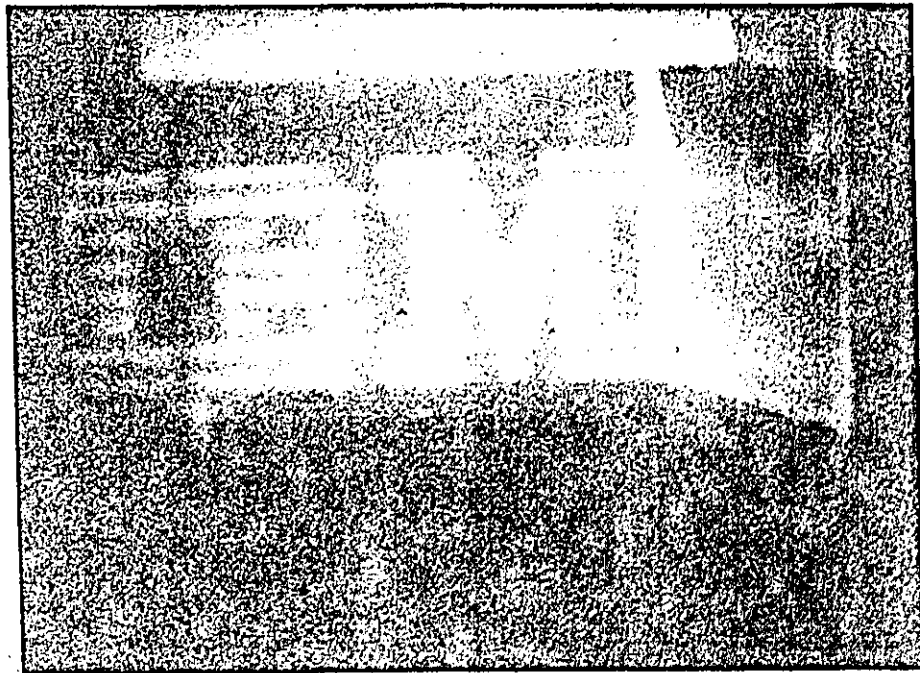


FIG. 4.16 b) *RESTORED IMAGE (GLOBAL).*

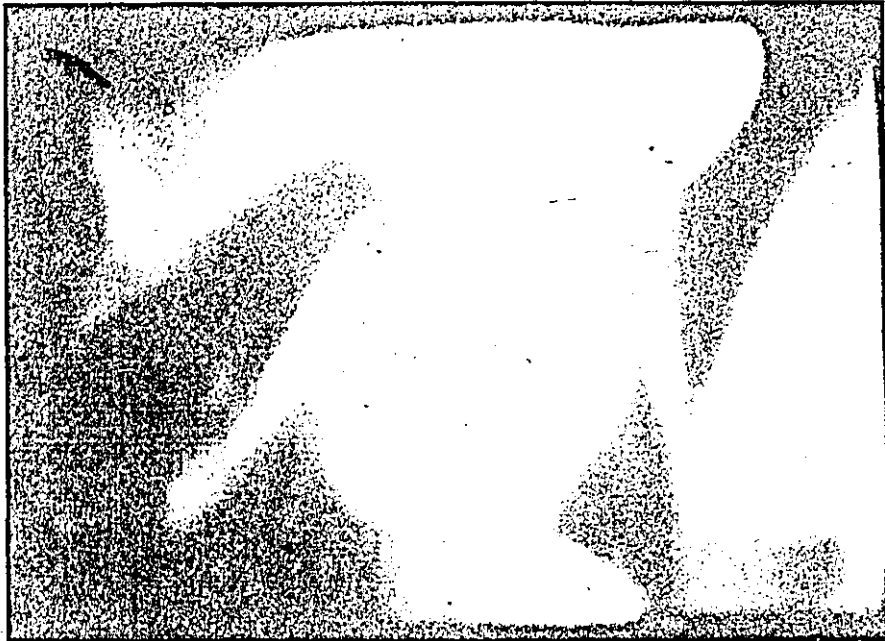


FIG. 4.17 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*



FIG. 4.17 b) *RESTORED IMAGE (GLOBAL).*



FIG. 4.18 a) *BLURRED IMAGE BY COMPUTER SIMULATION.*



FIG. 4.18 b) *RESTORED IMAGE (GLOBAL).*

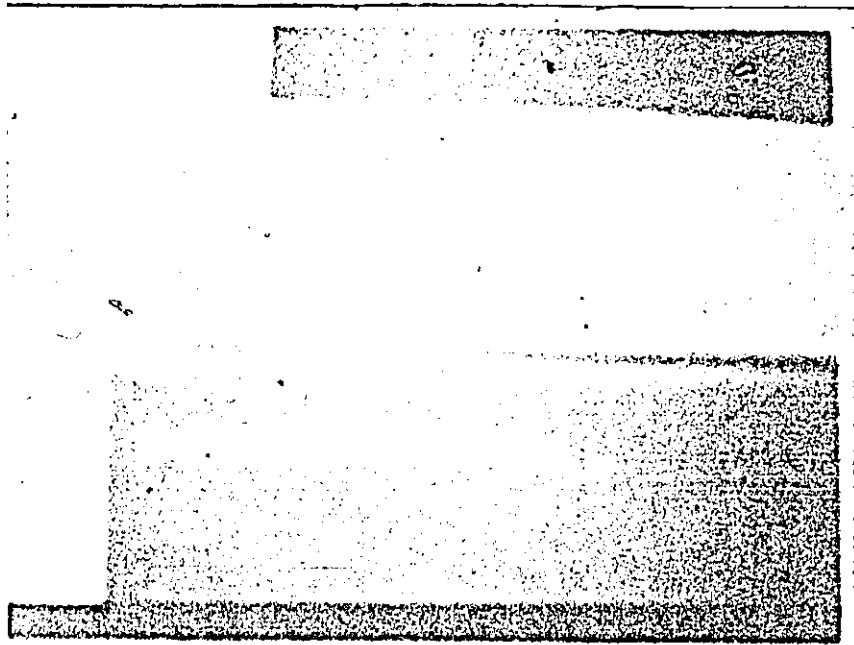


FIG. 4.19 a) *BLURRED IMAGE BY CAMERA.*

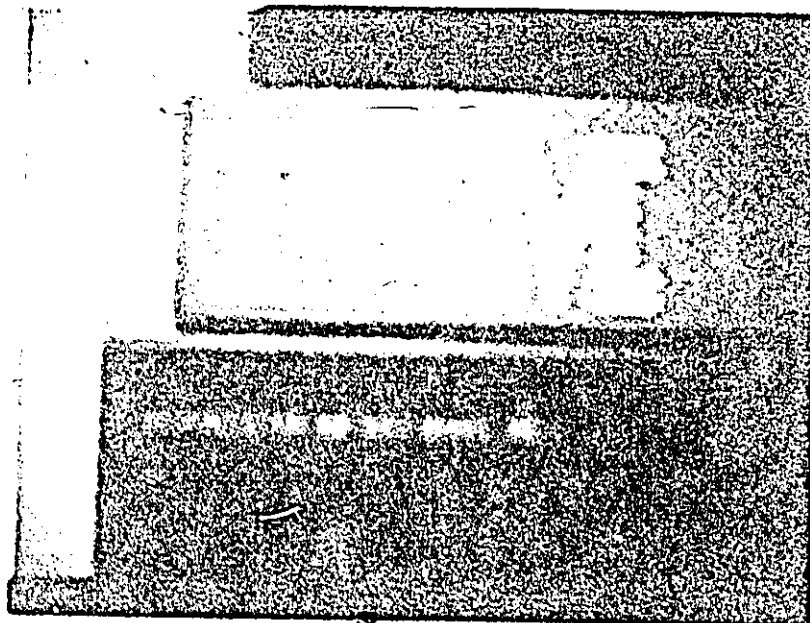


FIG. 4.19 b) *RESTORED IMAGE (GLOBAL).*

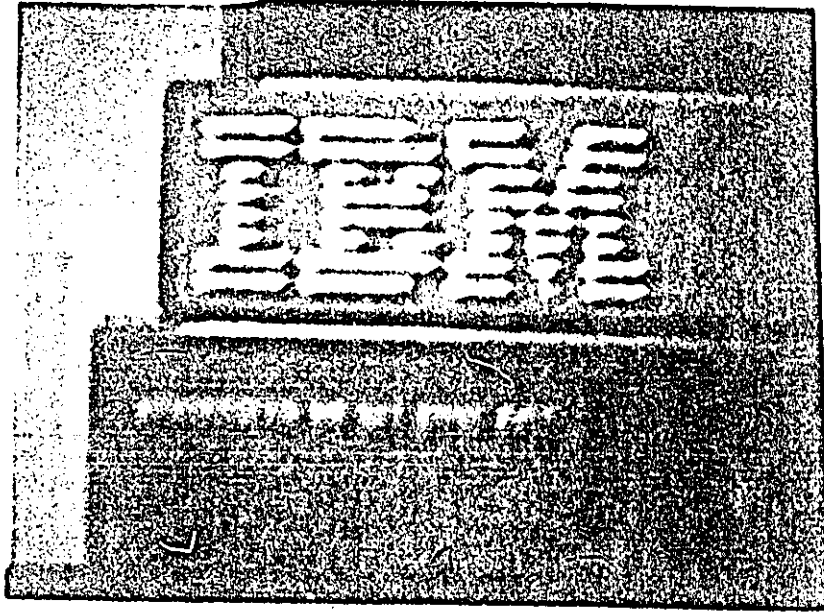


FIG. 4.19 c) RESTORED IMAGE (LOCAL 0% OVERLAPPING).

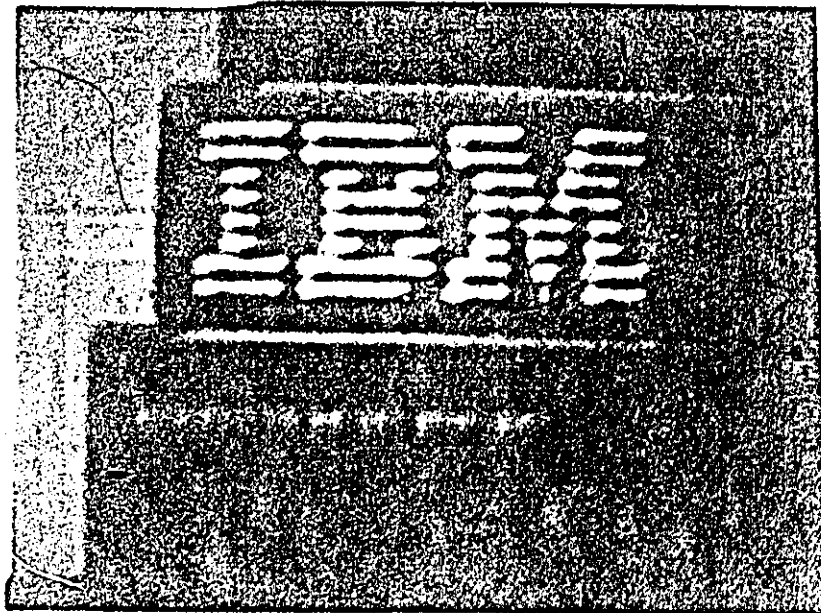


FIG. 4.19 d) RESTORED IMAGE (LOCAL 50% OVERLAPPING).



FIG. 4.20 a) *BLURRED IMAGE BY CAMERA.*



FIG. 4.20 b) *RESTORED IMAGE (GLOBAL).*

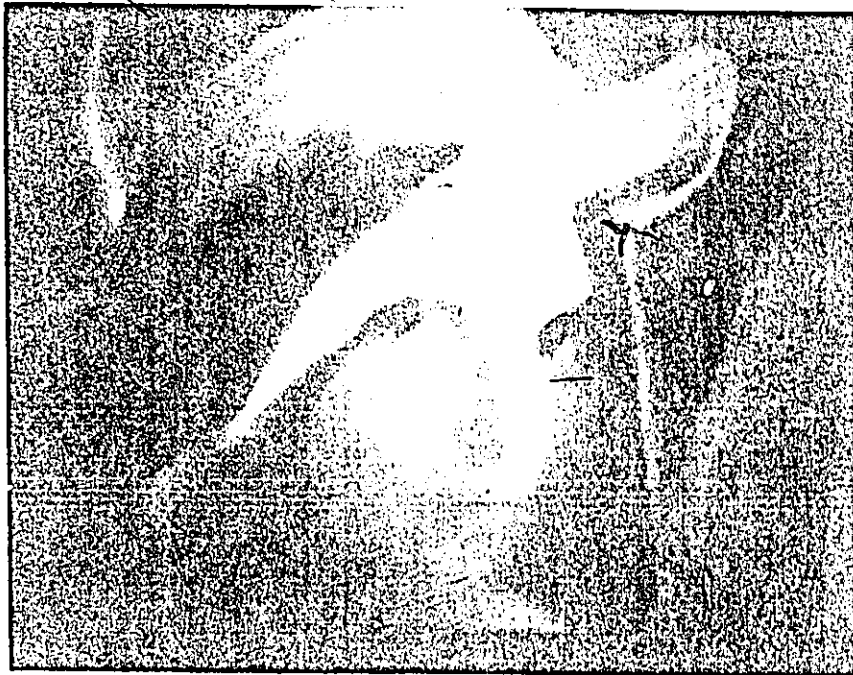


FIG. 4.20 c) RESTORED IMAGE (LOCAL 0% OVERLAPPING).



FIG. 4.20 d) RESTORED IMAGE (LOCAL 50% OVERLAPPING).

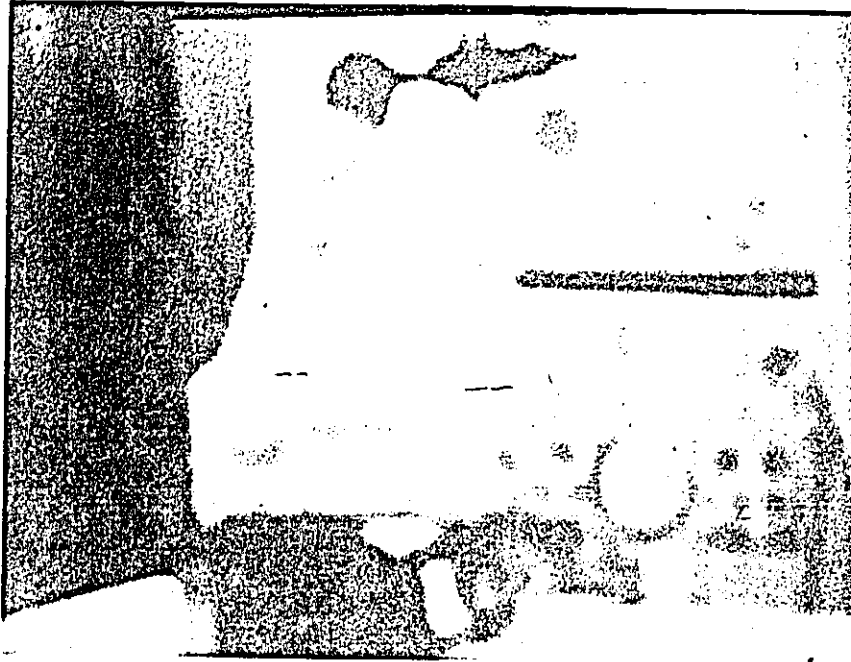


FIG. 4.21 a) *BLURRED IMAGE BY CAMERA.*

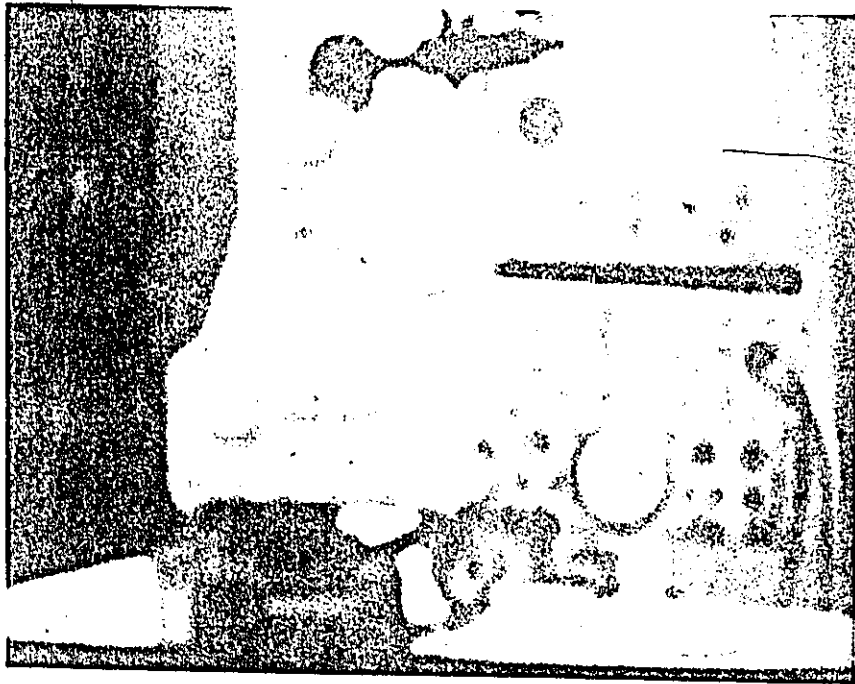


FIG. 4.21 b) *RESTORED IMAGE (GLOBAL).*

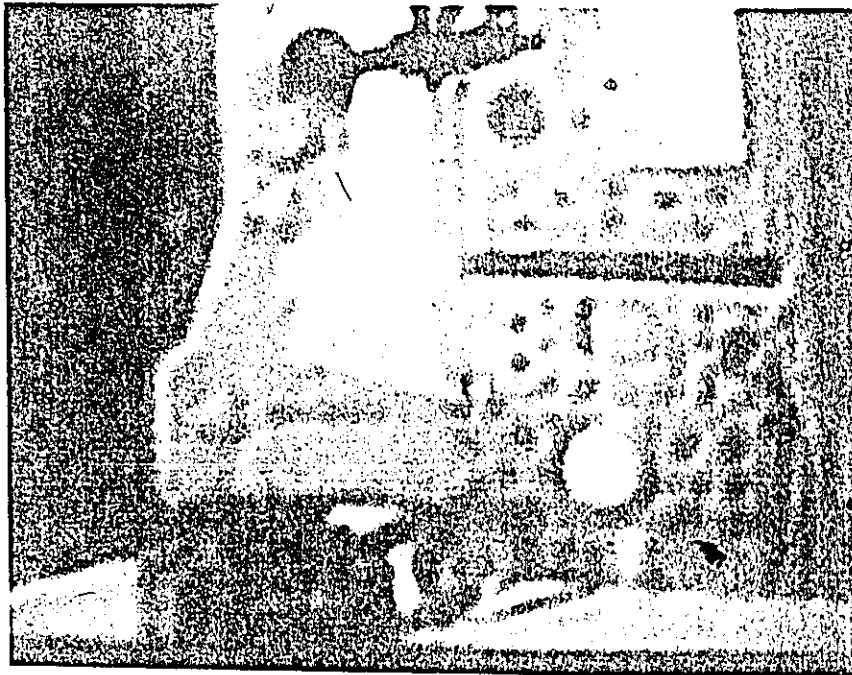


FIG. 4.21 c) RESTORED IMAGE (LOCAL 0% OVERLAPPING).



FIG. 4.21 d) RESTORED IMAGE (LOCAL 50% OVERLAPPING).

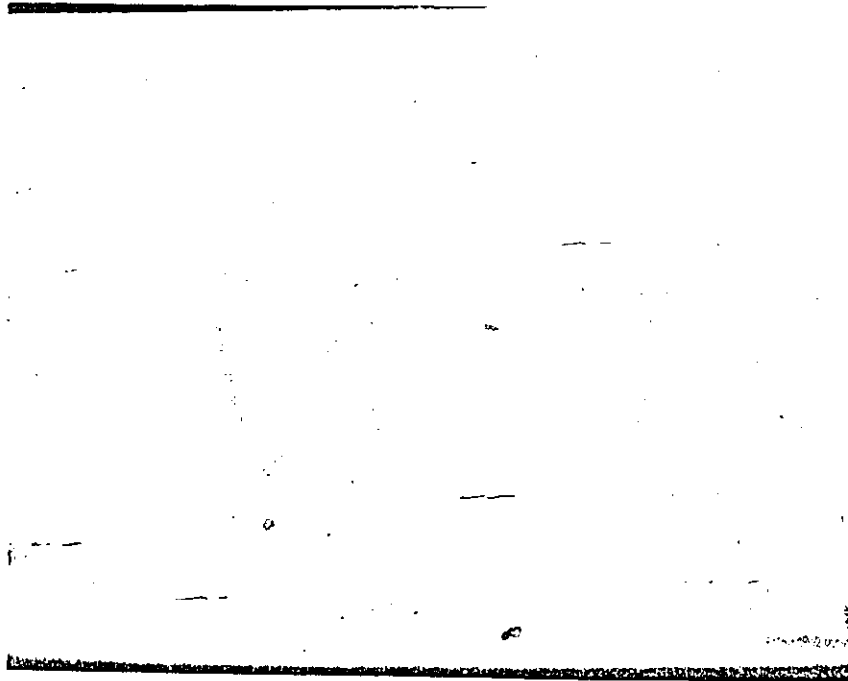


FIG. 4.22 a) *BLURRED IMAGE BY CAMERA.*



FIG. 4.22 b) *RESTORED IMAGE (GLOBAL).*



FIG. 4.22 c) RESTORED IMAGE (LOCAL 0% OVERLAPPING).



FIG. 4.22 d) RESTORED IMAGE (LOCAL 50% OVERLAPPING).

4.4.2 Conclusion and comments.

In this section algorithms were provided for focusing images suffering from the out-of focus blur. Three cases were considered.

The first case where the blurred and original images are given, was just an intermediate step to check if inverse filtering in the space domain through the use of 2-D IIR filters is valid. The results obtained, which are shown in Figs.(4.5) to (4.10), give enough evidence of the validity of the approach.

For the second case restoration has been performed, and the results are highly promising, but the principal problem with this method is to find a similar in focus picture to the blurred one, and this was not an easy task. To obtain the results shown in Figs.(4.13), and (4.14), we tried the restoration process on 20 different images. The major weakness of the solution is its inability to correct for the unknown phase distortions in the general case.

The third case requires only the blurred image and the only assumption made was that the PSF can be approximated by a two-dimensional Gaussian. The validity of using a Gaussian to describe the point-spread function is described in [19]. The results obtained by this method which are shown in Figs.(4.15) to (4.22), are as good as the ones obtained by the second approach (similar).

From the results obtained it can be concluded that the third case, where only the blurred image is available, gives results which are highly promising and is the most realistic case. It can be noticed also that local operators performs better than the global ones.

The results obtained by focusing of images blurred by computer simulation are almost perfect. The difference in sharpness between the blurred and restored images is evident.

The results obtained for images blurred by camera are not as good as the ones obtained for images blurred by computer. This can be easily explained by looking at Figs.(3.5), and (3.7) which show the plots of the magnitude and phase of the imaging system's optical transfer function. The plots show that the PSF introduced by the camera is scene dependent. A way to overcome this constraint was by using local operators. For example for the third case, the blurred image was divided into sub-images, then we calculated the corresponding 2-D IIR filter coefficients for the most blurred sub-image (i.e., the sub-image which has the greatest ' σ '). After that the filter was applied on all sub-images.

Table 1 shows the different values of ' σ ' before and after processing some of the images.

<i>Figure number</i>	<i>σ before proc.</i>	<i>σ after proc.</i>
4.15	1.20	1.04
4.16	1.40	1.11
4.17	1.30	1.04
4.18	1.70	1.25
4.19	1.60	1.20
4.20	1.37	1.07
4.21	1.34	1.05
4.22	1.43	1.10

TABLE 1 DIFFERENT VALUES OF ' σ ' BEFORE AND AFTER PROCESSING.

Chapter V.

SUMMARY AND CONCLUSIONS.

5.1 SUMMARY OF THE WORK DONE.

This work has dealt with image restoration. The problem considered was: " FOCUSING OF IMAGES USING SPATIAL OPERATORS. ".

A spatial domain technique for restoring out-of focus images, with a minimum priori knowledge about the point-spread function of the imaging system was presented. The restoration has been achieved by mean of 2-D recursive (IIR) digital filters designed to meet the PSF specifications in a least-squares sense. The procedure was applied to different blurred images for different cases. The results obtained demonstrated the effectiveness of the method. Also for a matter of comparison frequency domain techniques (inverse filtering and Wiener filter) were presented. The difference in sharpness and resolution between the different results obtained by the two approaches is evident. This can be explained by the following:

It was pointed out in chapter III that the PSF introduced by the camera is scene dependent and therefore the degradation process or PSF has to be determined for every case. This fact was not utilized for the inverse and Wiener filtering approaches. The optical transfer function of the imaging system was obtained from one scene, then was applied to restore a second one, as described by other researchers in the field [21].

5-2 CONCLUSIONS.

From the above work it is concluded that 2-D recursive (IIR) filters can be applied for restoring out-of focus images.

The restoration procedure required only a modest amount of computing time with respect to the frequency domain approach. The restoration algorithms, for images of size 128×128 , required less than three minutes on an IBM AT personal computer running at 6 Mhz. While for the frequency domain it required between 13 to 15 minutes.

It was our experience that for reasonable amount of blur the algorithms converged, and we obtained a good restored image. However for severe distortions the algorithms began to fail. Also like it was pointed out at the beginning of this work we considered only the out-of focus blur for the case where the image is relatively noise free. Other methods need to be explored for other sources of blurs and for the noisy images.

REFERENCES.

- [1] M. D. McFarlane, " Digital Pictures Fifty Years Ago", Proc. IEEE, Vol. 60, No. 7, pp. 768-770, July 1972.
- [2] A. Marechal, P. Croce, and K. Dietzel, " Amelioration du Contraste des Details des Images Photographiques par Filtrage des Frequences Spatiales", Opt. Acta, Vol. 5, pp. 256-262, 1958.
- [3] R. Nathan, " Picture Enhancement for the Moon, Mars, and Man", in Pictorial Pattern Recognition, G. C. Cheung, Ed. Washington, D.C.: Thompson, 1968.
- [4] T. S. Huang, W. F. Schreiber, and O. J. Tretiak, " Image Processing", Proc. IEEE, Vol. 59, pp. 1586-1609, November 1971.
- [5] J. L. Harris, Sr, " Image Evaluation and Restoration", J. Opt. Soc. Am., Vol. 56, pp. 569-574, May 1966.
- [6] C. W. Helstrom, " Images Restoration by the Method of Least-squares", J. Opt. Soc. Am., Vol. 57, pp. 297-303, July 1967.
- [7] M. M. Sondhi, " Image Restoration: The Removal of Spatially Invariant Degradations", Proc. IEEE, Vol. 60, pp. 842-853, July 1972.
- [8] D. Slepian, " Linear Least-Squares Filtering", J. Opt. Soc. Am., Vol. 57, pp. 918-922, July 1967.

- [9] B. R. Hunt, "The Application of Constrained Least-Squares Estimation to Image Restoration by Digital Computer", IEEE Trans. on Computers, Vol. C-22, No. 9, pp. 805-812, September 1973.
- [10] N. N. Abdelmalek, T. Kasvand, J. Olmstead, and M. M. Tremblay, "Direct Algorithm for Digital Image Restoration", Applied Optics, Vol. 20, No. 24, pp. 4227-4232, December 1981.
- [11] W. K. Pratt, "Digital Image Processing", Wiley, New York, 1978.
- [12] H. C. Andrews, and B. R. Hunt, "Digital Image Restoration", Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
- [13] T. G. Stockham, Jr., T. M. Cannon, and R. B. Ingrebetsen, "Deconvolution Through Digital Signal Processing", Proc. IEEE, Vol. 63, No. 4, pp. 678-692, April 1975.
- [14] T. M. Cannon, "Blind Deconvolution of Spatially Invariant Image Blur with Phase", IEEE Trans. Acoust., Speech., Signal Process., Vol. ASSP-24, No. 1, pp. 58-63, February 1976.
- [15] K. von der Heide, "Least-Squares Image Restoration", Opt. Commun., Vol. 31, pp. 279-284, 1979.

- [16] J. B. Morton, and H. C. Andrews, " A Posteriori Method of Image Restoration", J. Opt. Soc. Am., Vol. 69, No. 2, pp. 280-290, February 1979.
- [17] R. K. Ward, and B. E. Saleh, " Deblurring Random Blur", IEEE Trans. Acoust., Speech, Signal Process., Vol. ASSP-35, No. 1, pp. 1494-1498, October 1987.
- [18] D. E. Dudgeon, R. M. Mersereau, " Multidimensional Digital Signal Processing", Prentice-Hall, Inc., Englewood Cliffs, New Jersey 1984.
- [19] A. Pentland, " A New Sense for Depth of Field", IEEE Trans. on PAMI, Vol. PAMI-9, No. 4, pp. 523-531, July 1987.
- [20] J. L. Shanks, S. Treitel, and J. H. Justice, " Stability and Synthesis of Two-Dimensional Recursive Filters", IEEE Trans. on Audio and Electroacoustics, Vol. AU-20, No. 2, pp. 115-128, June 1972.
- [21] R. C. Gonzalez, and P. Wintz, " Digital Image Processing", Addison-Wesley Publishing Company, Inc., 1977.
- [22] R. E. Woods, and R. C. Gonzalez, " Real-time Digital Image Enhancement", Proc. IEEE, Vol. 69, No. 5, pp. 643-654, May 1981.
- [23] T. G. Stockham, Jr., " Image Processing in the Context of a Visual Model", Proc. IEEE, Vol. 60, No. 7, pp. 828-842, July 1972.

- [24] R. A. Hummel, "Image Enhancement by Histogram Transformation", Comput. Graph. Image Proc., Vol. 6, pp. 184-195, June 1977.
- [25] W. Frei, "Image Enhancement by Histogram Hyperbolization", Comput. Graph. Image Proc., Vol. 6, pp. 286-294, June 1977.
- [26] R. C. Gonzalez, and B. A. Fittes, "Grey-level Transformations for Interactive Image Enhancement", Mechanism Mach. Theory, Vol. 12, pp. 111-122, December 1977.
- [27] A. Rosenfeld, "Image Pattern Recognition", Proc. IEEE, Vol. 69, No. 5, pp. 596-605, May 1981.
- [28] A. Papoulis, "Probability Random Variables, and Stockastic Processes", McGraw-Hill, New York, 1965.
- [29] L. Weinberg, "Network Analysis and Synthesis", McGraw-Hill, New York, 1962.
- [30] A. Budak, "Passive and Active Network Analysis and Synthesis", Houghton-Mifflin, Boston, 1974.
- [31] A. O. Aboutaleb, and L. M. Silverman, "Restoration of Motion Degraded Images", IEEE Trans. Circuits and Systems, Vol. CAS-22, No. 3, pp. 278-286, March 1975.

- [32] A. O. Aboutaleb, M. S. Murphy, and L. M. Silverman, "Digital Restoration of Images Degraded by General Motion Blurs", IEEE Trans. Automatic Control, Vol. AC-22, No. 3, pp. 294-302, June 1977.
- [33] D. A. O'Handley, and W. B. Green, "Recent Developments in Digital Image Processing Laboratory at the Jet Propulsion Laboratory", Proc. IEEE, Vol. 60, No. 7, pp. 821-828, July 1972.
- [34] M. Born, and E. Wolf, "Principles of Optics", London: Pergamon Press, Fifth Edition, 1975.
- [35] G. M. Robbins, and T. S. Huang, "Inverse Filtering for Linear Shift-Variant Imaging Systems", Proc. IEEE, Vol. 60, No. 7, pp. 862-872, July 1972.
- [36] M. P. Ekstrom, "A Numerical Algorithm for Identifying Spread Functions of Shift-Invariant Imaging Systems", IEEE Trans. on Computers, Vol. C-22, No. 4, pp. 322-327, April 1973.
- [37] A. A. Sawchuk, "Space-Variant Image Restoration by Coordinate Transformations", J. Opt. Soc. Am., Vol. 64, No. 2, pp. 138-144, February 1974.
- [38] B. R. Hunt, "Digital Image Processing", Proc. IEEE, Vol. 63, No. 4, pp. 693-708, April 1975.
- [39] S. S. Reddi, "Constrained Least-Squares Image Restoration: An Improved Computational Scheme", Applied Optics, Vol. 17, No. 15, pp. 2340-2341, August 1978.

- [40] R. K. Ward, and B. E. A. Saleh, " Restoration of Images Distorted by Systems of Random Impulse Response", J. Opt. Soc. Am. A, Vol. 2, No. 8, pp. 1254-1259, August 1985.
- [41] A. M. Tekalp, H. Kaufman, and J. W. Woods, " Identification of Image and Blurs Parameters for the Restoration of Noncausal Blurs", IEEE Trans. Acoust., Speech, Signal Process., Vol. ASSP-34, NO. 4, pp. 963-972, August 1986.
- [42] C. S. Burrus, and T. W. Parks, " Time Domain Design of Recursive Digital Filters", IEEE Trans. on Audio and electroacoustics, Vol. AU-18, No. 2, pp. 137-141, June 1970.
- [43] E. R. Cole, " The Removal of Unknown Image Blurs By Homomorphic Filtering", Comput. Sci. Dep., Univ. Utah, Salt Lake City, UTEC_CSc_74-029, June 1973.
- [44] H. J. Trussel, " Comments on 'Picture Thresholding Using An Iterative Selection Method'", IEEE Trans. on SMC, Vol. SMC-9, pp. 311, May 1979.
- [45] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, " Numerical Recipes", Cambridge University Press, pp. 31-43, 1986.
- [46] J. S. Wezcka, R. N. Nagel, and A. Rosenfeld, " A Threshold Selection Technique", IEEE Trans. on Comput., Vol. C-23, pp. 1322-1326, 1974.

- [47] M. A. Janjua, M. A. Sid-Ahmed, G. A. Jullien, and N. Rajendran, " Multi-level Thresholding Using Image Transformation", Can. Elect. Eng. J., Vol. 19, No. 1, pp. 29-34, 1984.
- [48] N. Rajendran, M. A. Sid-Ahmed, " Selection of a Thresholding Scheme for On-line Quality Inspection", Can. Elect. Eng. J., Vol. 8, No. 4, pp. 130-134, 1983.
- [49] M. A. Janjua, " Image Thresholding and Feature Extraction Techniques", Master Thesis, Univ. of Windsor, Dept. of Elect. Engg., 1982.
- [50] N. Rajendran, " Surface Flaw Detection Using Digital Image Processing Technique", Master Thesis, Univ. of Windsor, Dept. of Elect. Engg., 1984.

APPENDIX - A -

C---- This program will perform the inverse filtering operation in the frequency domain by the mean of an FFT. It performs the deconvolution of two functions. See chapter IV for details.

```
REAL Y(128,128),X(128,128),GMAG(128,128),GPHA(128,128)
```

```
REAL A(10,10),B(10),Z(10)
```

```
CHARACTER G(128,128)
```

```
CHARACTER*13 FILN
```

```
PI=6.283185307179586/2.
```

```
NSIZ=128
```

```
M=7
```

```
WRITE(*,51)
```

```
51 FORMAT(' INPUT FILENAME FOR THE FIRST IMAGE ----> '\)
```

```
READ(*,'(A13)')FILN
```

```
OPEN(1,FILE=FILN,FORM='BINARY',STATUS='OLD')
```

```
READ(1)((G(I,J),J=1,NSIZ),I=1,NSIZ)
```

```
CLOSE(1)
```

C---- The first input image is read and is written in X (real) and Y (imaginary).

```
DO 1 I=1,NSIZ
```

```
DO 1 J=1,NSIZ
```

```
    X(I,J)=ICHAR(G(I,J))
```

```
    Y(I,J)=0.
```

```
1 CONTINUE
```

```
CALL TRANS(X,Y,NSIZ,M,1)
```

```
WRITE(*,*) ' THE FIRST DFT IS DONE'
```

```

DO 8 I=1, NSIZ
DO 8 J=1, NSIZ
    TX=X(I, J)
    GMAG(I, J)=SQRT(X(I, J)**2+Y(I, J)**2)
    GPHA(I, J)=ATAN(Y(I, J)/X(I, J))
    IF(TX.LT.0.)GPHA(I, J)=GPHA(I, J)+PI
8   CONTINUE
    WRITE(*, 52)
52  FORMAT(' INPUT FILENAME FOR THE SECOND IMAGE ---> '\)
    READ(*, ' (A13)')FILN
    OPEN(2, FILE=FILN, FORM='BINARY', STATUS='OLD')
    READ(2)((G(I, J), J=1, NSIZ), I=1, NSIZ)
    CLOSE(2)

```

C---- The second input image is read and is written in X
(real) and Y (imaginary).

```

DO 11 I=1, NSIZ
DO 11 J=1, NSIZ
    X(I, J)=ICHR(G(I, J))
    Y(I, J)=0.
11  CONTINUE
    CALL TRANS(X, Y, NSIZ, M, 1)
    WRITE(*, *) ' THE SECOND DFT IS DONE'
DO 18 I=1, NSIZ
DO 18 J=1, NSIZ
    TX=X(I, J)
    IF(GMAG(I, J).EQ.0.)GO TO 132
    GMAG(I, J)=SQRT(X(I, J)**2+Y(I, J)**2)/GMAG(I, J)
132  GPHA(I, J)=ATAN(Y(I, J)/X(I, J))-GPHA(I, J)
    IF(TX.LT.0.)GPHA(I, J)=GPHA(I, J)+PI
18  CONTINUE

```

✓

```

WRITE(*,*) ' H(U,V) IS DETERMINED'
WRITE(*,53)
53  FORMAT(' INPUT FILENAME OF THE BLURRED IMAGE ---> '\)
READ(*,' (A13)')FILN
OPEN(3,FILE=FILN,FORM='BINARY',STATUS='OLD')
READ(3)((G(I,J),J=1,NSIZ),I=1,NSIZ)
CLOSE(3)

```

C---- The blurred image is read and is written in X (real) and Y (imaginary).

```

DO 111 I=1,NSIZ
DO 111 J=1,NSIZ
X(I,J)=ICHAR(G(I,J))
Y(I,J)=0.
111 CONTINUE
CALL TRANS(X,Y,NSIZ,M,1)
WRITE(*,*) ' THE THIRD DFT IS DONE'
DO 118 I=1,NSIZ
DO 118 J=1,NSIZ
TX=X(I,J)
IF(GMAG(I,J).EQ.0.)GO TO 1132
GMAG(I,J)=SQRT(X(I,J)**2+Y(I,J)**2)/GMAG(I,J).
1132 GPHA(I,J)=ATAN(Y(I,J)/X(I,J))-GPHA(I,J)
IF(TX.LT.0.)GPHA(I,J)=GPHA(I,J)+PI
118 CONTINUE
WRITE(*,*) ' OUT(U,V) IS DETERMINED'
DO 136 I=1,NSIZ
DO 136 J=1,NSIZ
X(I,J)=GMAG(I,J)*COS(GPHA(I,J))
Y(I,J)=GMAG(I,J)*SIN(GPHA(I,J))
136 CONTINUE

```

```

CALL TRANS(X,Y,NSIZ,M,-1)
WRITE(*,*) ' THE INVERSE DFT IS DONE'
DO 14 I=1,NSIZ
DO 14 J=1,NSIZ
    X(I,J)=SQRT(X(I,J)**2+Y(I,J)**2)
14 CONTINUE
MAX=X(1,1)
MIN=X(1,1)
DO 15 I=1,NSIZ
DO 15 J=1,NSIZ
    IF(MAX.LT.X(I,J))MAX=X(I,J)
    IF(MIN.GT.X(I,J))MIN=X(I,J)
15 CONTINUE
WRITE(*,*) 'MAX=',MAX,'MIN=',MIN
RANGE=MAX-MIN
DO 177 I=1,NSIZ
DO 177 J=1,NSIZ
177 X(I,J)=(255.*(X(I,J)-MIN))/RANGE
WRITE(*,54)
54 FORMAT(' INPUT FILENAME FOR THE OUTPUT IMAGE ---> ')
READ(*,'(A13)')FILN
OPEN(7,FILE=FILN,FORM='BINARY',STATUS='NEW')
DO 160 I=1,NSIZ
DO 160 J=1,NSIZ
    NX=X(I,J)
    G(I,J)=CHAR(NX)
    WRITE(7)G(I,J)
160 CONTINUE
CLOSE(UNIT=7)
STOP
END

```


SUBROUTINE TRANS(X,Y,N,M,KODE)

C----- The subroutine TRANS will perform the 2-D DFT (OR IDFT). This is obtained by taking a transform (1-D DFT) along each row of F(X,Y), the desired result (2-D DFT) is then obtained by taking a transform along each column of F(X,V).

```

    DIMENSION X(128,128),Y(128,128),XX(128),YY(128)
    DO 9 J=1,N
      DO 10 I=1,N
        YY(I)=Y(I,J)
        XX(I)=X(I,J)
10      CONTINUE
      CALL FFT1(XX,YY,N,M,KODE)
      DO 11 I=1,N
        Y(I,J)=YY(I)
        X(I,J)=XX(I)
11      CONTINUE
9      CONTINUE
    DO 12 I=1,N
      DO 13 J=1,N
        XX(J)=X(I,J)
        YY(J)=Y(I,J)
13      CONTINUE
      CALL FFT1(XX,YY,N,M,KODE)
      DO 14 J=1,N
        Y(I,J)=YY(J)
        X(I,J)=XX(J)
14      CONTINUE
12     CONTINUE
    RETURN
    END

```

SUBROUTINE FFT1(X,Y,N,M,KODE)

C---- KODE = 1 for DFT and -1 for IDFTX and Y are the input
and later the output.

REAL X(128),Y(128)

N2=N

DO 10 K=1,M

N1=N2

N2=N2/2

E=6.283185307179586/N1

A=0.

DO 20 J=1,N2

C=COS(A)

S=SIN(A)

S=S*KODE

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---- Digit reverse.

```
100 J=1
    N1=N-1.
    DO 104 I=1,N1
        IF(I.GE.J)GO TO 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)GO TO 103
    J=J-K
    K=K/2
    GO TO 102
103 J=J+K
104 CONTINUE
    RETURN
    END
```

APPENDIX - B -

C This program determines the filter coefficients by solving a matrix determined from the minimization of the total squared error between the original (in focus) and blurred (out-of focus) images (see chapter IV for the derivation of the equations).

```
REAL*4 Y(256,256), X(256,256), AA(50,50), MIN, MAX, A(50,50)
REAL*4 B(50), Z(50), BB(50,50) SUM1, SUM2, SUM3, SUM4
CHARACTER G(128,128)
```

```
CHARACTER*13 filn
```

```
WRITE(*,*)'ENTER THE SIZE OF THE IMAGE'
```

```
READ(*,*)NSIZ
```

```
write(*,33)
```

```
33 format(' input filename ----> '\)
```

```
read(*,'(a13)')FILN
```

```
OPEN(1,FILE=FILN,FORM='BINARY',STATUS='OLD')
```

```
READ(1)((G(I,J),J=1,NSIZ),I=1,NSIZ)
```

```
CLOSE(1)
```

C The out-of focus image is read and is written in X.

```
DO 1 I=1,NSIZ
```

```
DO 1 J=1,NSIZ
```

```
    X(I,J)=ICHAR(G(I,J))
```

```
1 CONTINUE
```

```
WRITE(*,*)'THE FIRST IMAGE IS READ'
```

```
WRITE(*,34)
```

```
34 FORMAT(' input filename ----> '\)
```

```
READ(*,'(A13)')FILN
```

```
OPEN(2,FILE=FILN,FORM='BINARY',STATUS='OLD')
```

```
READ(2)((G(I,J),J=1,NSIZ),I=1,NSIZ)
```

```
CLOSE(2)
```

C The in focus image is read and is written in Y.

```
      DO 11 I=1,NSIZ
      DO 11 J=1,NSIZ
          Y(I,J)=ICHR(G(I,J))
11  CONTINUE
      WRITE(*,*)'THE SECOND IMAGE IS READ'
      OPEN(4,FILE='COEFFS.DAT',STATUS='NEW')
      WRITE(*,*)'ENTER THE ORDER OF THE FILTER'
      READ(*,*)N
      WRITE(*,*)'ENTER NR THE SIZE OF THE SUB-IMAGES'
      READ(*,*)NR
      WRITE(*,*)'ENTER NNR (1-->0%,2-->50%,4-->75% OVERLAPPING)'
      READ(*,*)NNR
      WRITE(*,*)'ORDER=',N
      N1=N+1
      NN1=2*N1*N1-1
      WRITE(*,*)'NN1=',NN1
```

C Set the coefficients equal to zero.

```
      DO 39 I=1,NN1
      DO 390 J=1,NN1+1
          A(I,J)=0.
390 CONTINUE
      B(I)=0.
39  CONTINUE
      KK=1
      NR1=NR-1
      NR2=INT(NR/NNR)
      NNI=1
      NNJ=1
```

C...Determine the matrix.

2000 DO 2 L=1,N1

DO 2 K=1,N1

L1=L-1

K1=K-1

DO 3 I=1,N1

DO 3 J=1,N1

I1=I-1

J1=J-1

SUM1=0.

SUM2=0.

SUM3=0.

SUM4=0.

DO 4 IN=NNI,NNI+NN1

DO 4 IM=NNJ,NNJ+NN1

IF(IN.LE.I1.OR.IM.LE.J1)GO TO 4

IF(IN.LE.L1.OR.IM.LE.K1)GO TO 4

SUM1=SUM1+X(IN-I1,IM-J1)*X(IN-L1,IM-K1)

SUM2=SUM2+Y(IN-I1,IM-J1)*X(IN-L1,IM-K1)

SUM3=SUM3+X(IN-I1,IM-J1)*Y(IN-L1,IM-K1)

SUM4=SUM4+Y(IN-I1,IM-J1)*Y(IN-L1,IM-K1)

4

CONTINUE

MN=(N+1)*I1+J

NN=(N+1)*L1+K

KN=NN+(N+1)**2-1

LN=MN+(N+1)**2-1

A(NN,MN)=SUM1

A(NN,KN)=-SUM2

A(KN,MN)=SUM3

A(KN,KN)=-SUM4

```

                IF (I1.EQ.0.AND.J1.EQ.0) THEN
                    B(NN)=SUM2
                    B(KN)=SUM4
                ENDIF
3           CONTINUE
2           CONTINUE

```

C The matrix which has to be solved is written in A.

```

                DO 160 I=1,NN1
                    A(I,NN1+1)=B(I)
160          CONTINUE

```

C The matrix is solved using Gauss-Jordan with partial pivoting. The filter coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$ are written in AA and BB respectively, and are stored in COEFFS.DAT.

```

                CALL GAUSS(A,NN1,Z)
                K=0
                DO 62 I=1,N1
                    DO 62 J=1,N1
                        K=K+1
                        AA(I,J)=Z(K)
                        WRITE(4,*)AA(I,J)
62          CONTINUE
                DO 63 I=1,N1
                    IF(I.EQ.1) THEN
                        NX1=N
                    ELSE
                        NX1=N1
                    ENDIF

```

```

        DO 63 J=1,NX1
            K=K+1
            JJ=J
            IF (I.EQ.1) JJ=J+1
            BB(I,JJ)=Z(K)
63      CONTINUE
        BB(1,1)=1.00
        DO 143 I=1,N1
            DO 143 J=1,N1
                WRITE(4,*)BB(I,J)
143    CONTINUE
        KK=KK+1
        NNJ=NNJ+NR2
        IF (NNJ.LE.NSIZ-NR1) GO TO 2000
        NNJ=1
        NNI=NNI+NR2
        IF (NNI.LE.NSIZ-NR1) GO TO 2000
        CLOSE (UNIT 4)
        STOP
        END

        SUBROUTINE GAUSS(A,NN1,X)
        REAL A(50,50),X(50)

```

C A is the augmented matrix.
The 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
  ENDIF
20 CONTINUE
  IF (BIG.LT.1E-07) THEN
    WRITE(*,*) 'NO UNIQUE SOLUTION'
    RETURN
  ENDIF
  DO 30 K=1,N+1
    TEMP=A(J,K)
    A(J,K)=A(L,K)
30  A(L,K)=TEMP
    DO 40 K=J+1,N+1.
40  A(J,K)=A(J,K)/A(J,J)
    A(J,J)=1.0
    DO 10 I=1,N
      IF (I.EQ.J) GO TO 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

```

APPENDIX - C -

C---- A program for filtering large images given coefficients of a 2-D recursive filter. Provided by Dr Sid-Ahmed Dept. of Elect. Engg. Univ. of Windsor.

```
REAL*4 A(5,5);B(5,5)
INTEGER*4 IX(5,512), IY(5,512), IYT, MIN, MAX
CHARACTER IARRAY(512)
CHARACTER*13 FILN
MIN=0
MAX=255
WRITE(*,51)
51 FORMAT(' ENTER SIZE OF IMAGE. (128X128,256X256 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
B    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))

```

```

DO 23 M=1,NS
  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=IY(I,J)
        IF(IYT.LT.MIN)IYT=MIN
        IF(IYT.GT.MAX)IYT=MAX
        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.O)) THEN
  DO 22 I=K,1,-1
    DO 15 J=1,NSIZ
      IYT=IY(I,J)
      IF (IYT.LT.MIN) IYT=MIN
      IF (IYT.GT.MAX) IYT=MAX
      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)
      STOP
      END

```

APPENDIX - D -

C---- This pascal program is used to threshold the out-of-focus image which is needed for the location of the edges. It was provided by Dr Sid-Ahmed Dept of Elect. Engg. Univ. of Windsor.

```
program THRESHOLD(input,output);

var
    S1,S2,C1,C2,EM1,EM2 : real;
    ARR: array[1..128,1..255] of byte;
    F1,F2:file;
    READNAME: string[64];
    WRITENAME: STRING[64];
    V : array[0..255] of integer;
    LL: real;
    K1,K2,ITEMP,ITHR,NCN,I,J,K : byte;

procedure THRESHOLD;
label 100;
begin
    LL:=1.0;
    NCN:=1;
    ITHR:=128;
    for I:=0 to 255 do
        begin
            VEI1:=0;
        end;
end;
```

```

write('Enter input filename : ');
readln(READNAME);
assign(F1,READNAME);
reset(F1);
  for K:= 1 to 2 do
  begin
    blockread(F1,ARR,255);
    for I:=1 to 128 do
      for J:=1 to 255 do
        begin
          K1:=ARR[I,J];
          V[K1]:=V[K1]+1;
        end;
      end;
    end;
  close(F1);
100:
S1:=0;S2:=0;C1:=0;C2:=0;

  for K:=0 to ITHR do
  begin
    S1:=S1+LL*V[K]*K;
    C1:=C1+V[K];
  end;

  for K:=ITHR+1 to 255 do
  begin
    S2:=S2+LL*V[K]*K;
    C2:=C2+V[K];
  end;

```

```

EM1:=S1/C1;
EM2:=S2/C2;
ITEMP:=round((EM1+EM2)/2);
if ITEMP<>ITHR then
begin
    ITHR:=ITEMP;
    NCN:=NCN+1;
    GOTO 100;
end;

writeln(ITHR :4 , NCN:10);

Write('Enter output filename : ');
readln(WRITENAME);
assign(F2,WRITENAME);
reset(F1);
rewrite(F2);
for K:=1 to 2 do
begin
    blockread(F1,ARR,255);
for I:=1 to 128 do
for J:=1 to 255 do
begin
    if ARR[I,J]> ITHR then
        ARR[I,J]:=100
    else
        ARR[I,J]:=0;
end;
end;

```



```
    blockwrite(F2,ARR,255);  
end;  
    close(F1);  
    blockwrite(F2,ARR,2);  
    close(F2);
```

```
end;
```

```
begin
```

```
    THRESHOLD
```

```
end.
```

APPENDIX - E -

C---- This program is used to calculate σ the spatial constant of the Gaussian point-spread function from the degraded image see chapter IV for the derivation of the equations.

REAL X(128,128),SIGX(100),Y(128,128),AA,XC(128)

REAL C(128,128),SIGMA(128),SIMA(100),SIMI(100)

CHARACTER G(128,128)

CHARACTER*13 FILN

WRITE(*,*)'ENTER THE SIZE OF THE IMAGE NSIZ'

READ(*,*)NSIZ

WRITE(*,*)'ENTER NI'

READ(*,*)NI

WRITE(*,*)'ENTER XE'

READ(*,*)XE

WRITE(*,*)'ENTER NR THE SIZE OF THE SUB-IMAGES'

READ(*,*)NR

WRITE(*,*)'ENTER NNR (1-->0%,2 -->50%) OVERLAPPING,'

READ(*,*)NNR

WRITE(*,33)

33 FORMAT(' INPUT FILENAME----> '\)

READ(*, '(A13)')FILN

OPEN(1,FILE=FILN,FORM='BINARY',STATUS='OLD')

READ(1)((G(I,J),J=1,NSIZ),I=1,NSIZ)

CLOSE(1)

C---- The out-of focus image is read and is written in X.

DO 1 I=1,NSIZ

DO 1 J=1,NSIZ

X(I,J)=ICHR(G(I,J))

1 CONTINUE

WRITE(*,*)'THE FIRST IMAGE IS READ'

```

WRITE(*,34)
34  FORMAT(' INPUT FILENAME FOR THE THRESHOLDED IMG--> '/')
    READ(*,'(A13)')FILN
    OPEN(3,FILE=FILN,FORM='BINARY',STATUS='OLD')
    READ(3)((G(I,J),J=1,NSIZ),I=1,NSIZ)
    CLOSE(3)
C---- The thresholded version of the degraded image is read
      and is written in Y.

    DO 10 I=1,NSIZ
    DO 10 J=1,NSIZ
        Y(I,J)=ICHAR(G(I,J))
10  CONTINUE
    OPEN(2,FILE='SIGMA.DAT',STATUS='NEW')
C---- The gradient, in the x-direction, of the image is
      determined and is written in C.

    DO 11 I=1,NSIZ
    DO 11 J=1,NSIZ-1
        C(I,J)=ABS(X(I,J)-X(I,J+1))
11  CONTINUE
    DO 12 I=1,NSIZ
        C(I,NSIZ)=C(I,NSIZ-1)
12  CONTINUE
    KK=1
    NR1=NR-1
    NR2=INT(NR/NNR)
    NNI=1
    NNJ=1
    NI1=NI-1

```

```

2000 DO 13 I=NNI,NNI+NR1
      DO 2 J=NNJ+NI1,NNJ+NR1-NI1
        IF(Y(I,J).NE.Y(I,J+1))THEN
          Jo=J
          GO TO 24
        ELSE

```

C-----

c ---- this if statement is for detecting any empty row.

C-----

```

          IF(J.EQ.NNJ+NR1-NI1)GO TO 134-
          GO TO 2
        ENDIF

```

```

2      CONTINUE
24     I1=Jo-NI1
       I2=Jo+NI1

```

C---- Determine σ the spatial constant of the Gaussian PSF

```

      DO 20 J=I1,I2
        IF(C(I,J).LE.XE)GO TO 20
        Io=J
        CI=C(I,J)
        GO TO 23
      20  CONTINUE
      23  DO 3 J=I0,I2
          I01=Io-1

```

```

        IF (C(I,J).LT.CI) THEN
        XC(J)=0.
        IN=J-I01
        GO TO 124
        ELSE
        D=ABS(C(I,J))
        XC(J-I01)=ALOG(D)
        ENDIF
3      CONTINUE
124    SUM=0.
        DO 4 J=1, IN+1
            SUM=SUM+J**2
4      CONTINUE
        XB=SUM/(IN+1)
        A=0.
        B=0.
        DO 160 J=1, IN+1
            SUM1=J**2-XB
            SUM2=SUM1*J**2
            A=SUM1*XC(J)+A
            B=SUM2+B
160   CONTINUE
        IF (SUM1.EQ.0.) THEN
        AA=0.
        ELSE
        AA=(A/B)
        ENDIF

```

```

IF (AA.GE.0.)GO TO 134
SIGMA(I)=1./SQRT(-2.*AA)
GO TO 13
134  SIGMA(I)=-1.
13  CONTINUE
    XMIN=100.
    XMAX=0.
    DO 14 I=NNI,NNI+NR1
        IF (SIGMA(I).GE.0.) THEN
            IF (SIGMA(I).LT.XMIN) XMIN=SIGMA(I)
            IF (SIGMA(I).GT.XMAX) XMAX=SIGMA(I)
        ENDIF
14  CONTINUE
    SIMI(KK)=XMIN
    SIMA(KK)=XMAX
    KK=KK+1
    NNJ=NNJ+NR2
    IF (NNJ.LE.NSIZ-NR1)GO TO 2000
    NNJ=1
    NNI=NNI+NR2
    IF (NNI.LE.NSIZ-NR1)GO TO 2000
    DO 1300 KI=1, KK-1
        WRITE(2,*) SIMI(KI)
1300 WRITE(*,*) 'SIMI(', KI, ') = ', SIMI(KI)
    CLOSE(2)
    STOP
    END

```

APPENDIX - F -

C---- This program determines the filter coefficients by solving a matrix determined from the minimization of the total squared error by using Shanks method (see chapter IV and reference [20]).

```

REAL Y(256,256),X(256,256),AA(10,10),YY(256,256)
REAL MAX,MIN,BB(10,10),SIG
DIMENSION A(50,50),ALUD(50,50),INDX(50),B(50),BU(50)
WRITE(*,*)'ENTER THE SIZE OF THE IMAGE'
READ(*,*)NSIZ
PI=3.1415926
WRITE(*,*)'ENTER NR THE SIZE OF THE SUB-IMAGES'
READ(*,*)NR
WRITE(*,*)'ENTER NNR (1-->0%, 2-->50%,) OVERLAPPING)'
READ(*,*)NNR
NR1=NR-1
NR2=INT(NR/NNR)
KK=1
NNI=1
NNJ=1
OPEN(7,FILE='DATA.DAT',STATUS='NEW')
OPEN(2,FILE='SIGMA.DAT',STATUS='OLD')
WRITE(*,*)'ENTER THE ORDER OF THE FILTER'
READ(*,*)N
WRITE(*,*)'ORDER=',N
N1=N+1
NN1=N1*N1-1
WRITE(*,*)'NN1=',NN1
2000 READ(2,*)SIG
WRITE(*,*)'SIG=',SIG
IF(SIG.EQ.100.)GO TO 2001

```

```
SS=SIG*SIG
```

```
S=2*SS
```

C---- The ideal impulse response (Gaussian PSF) is read and is written in Y.

```
DO 11 I=1, NR
```

```
DO 11 J=1, NR
```

```
G=(I-1)**2+(J-1)**2
```

```
GS=G/S
```

```
GP=S*PI
```

```
GP=SQRT(GP)
```

```
GP=1./GP
```

```
Y(I, J)=GP*EXP(-GS)
```

```
11 CONTINUE
```

```
WRITE(*,*) 'THE SECOND IMAGE IS READ'
```

C---- Set the coefficients equal to zero.

```
DO 39 I=1, NN1
```

```
DO 390 J=1, NN1
```

```
A(I, J)=0.
```

```
390 CONTINUE
```

```
B(I)=0.
```

```
39 CONTINUE
```

C---- Determine the matrix to calculate the denominator coefficients $\{b_{ij}\}$ of the 2-D IIR filter.

```
DO 2 L=1, N1
```

```
DO 2 K=1, N1
```

```
L1=L-1
```

```
K1=K-1
```

```
IF (L1+K1.EQ.0) GO TO 2
```



```

DO 3 I=1,N1
DO 3 J=1,N1
  I1=I-1
  J1=J-1
  IF (I1+J1.EQ.0)GO TO 3
  SUM1=0.
  SUM2=0.
  DO 4 IN=1,NR
  DO 4 IM=1,NR
    IF (IN.LE.NI.AND.IN.LE.NI)GO TO 4
    IF (IN-I1.LE.0.OR.IM-J1.LE.0)GO TO 4
    IF (IN-I1.LE.0.OR.IM-K1.LE.0)GO TO 4
    SUM1=SUM1+Y(IN-I1,IM-J1)*Y(IN-L1,IM-K1)
    SUM2=SUM2+Y(IN,IM)*Y(IN-L1,IM-K1)
4  CONTINUE
  MN=(N+1)*I1+J1
  NN=(N+1)*L1+K1
  A(NN,MN)=SUM1
  B(NN)=-SUM2
3  CONTINUE
2  CONTINUE

```

C---- The matrix which has to be solved is written in ALUD and BU.

```

DO 160 I=1,NN1
DO 161 J=1,NN1
  ALUD(I,J)=A(I,J)
161 CONTINUE
  BU(I)=B(I)
160 CONTINUE

```

C--- The matrix is solved using LU decomposition of a rowwise permutation of itself. The routine LUDCMP is used in combination with LUBKSB to solve the linear system of equations. The used subroutines can be found in Reference [45].

```
CALL LUDCMP (ALUD, NN1, 50, INDX, D)
CALL LUBKSB (ALUD, NN1, 50, INDX, BU)
CALL MPROVE (A, ALUD, NN1, 50, INDX, B, BU)
K=0
DO 63 I=1, N1
  IF (I.EQ.1) THEN
    NX1=N
  ELSE
    NX1=N1
  ENDIF
  DO 63 J=1, NX1
    K=K+1
    JJ=J
    IF (I.EQ.1) JJ=J+1
    BB (I, JJ)=BU (K)
63  CONTINUE
    BB (1, 1)=1.00
    DO 5 IN=1, N1
      DO 5 IM=1, N1
        SUM3=0.
        DO 6 I=1, N1
          DO 6 J=1, N1
            IF (IN.LT.I.OR.IM.LT.J) GO TO 6
            SUM3=SUM3+BB (I, J)*Y (IN-I+1, IM-J+1)
6          CONTINUE
          AA (IN, IM)=SUM3
5        CONTINUE
```

```

GO TO 2002
2001 DO 2101 I=1,N1
      DO 2101 J=1,N1
          AA(I,J)=0.
          BB(I,J)=0.
2101 CONTINUE
      AA(1,1)=1.
      BB(1,1)=1.
2002 DO 143 I=1,N1
      DO 143 J=1,N1
          WRITE(7,*)AA(I,J)
143 CONTINUE
      DO 145 I=1,N1
      DO 145 J=1,N1
          write(7,*)BB(I,J)
145 CONTINUE
      KK=KK+1
      NNJ=NNJ+NR2
      IF(NNJ.LE.NSIZ-NR1)GO TO 2000
      NNJ=1
      NNI=NNI+NR2
      IF(NNI.LE.NSIZ-NR1)GO TO 2000
      CLOSE(UNIT=7)
      STOP
      END

```

SUBROUTINE LUDCMP (ALUD, N, NP, INDX, D)

PARAMETER (NMAX=100, TINY=1.0E-20)

DIMENSION ALUD (NP, NP), INDX (50), VV (NMAX)

C---- This subroutine replaces A by its LU decomposition of
a rowwise permutation of itself.

D=1.

DO 12 I=1, N

AAMAX=0.

DO 11 J=1, N

IF (ABS (ALUD (I, J)) .GT. AAMAX) AAMAX=ABS (ALUD (I, J))

11 CONTINUE

IF (AAMAX.EQ.0.) PAUSE 'SINGULAR MATRIX'

VV (I)=1./AAMAX

12 CONTINUE

DO 19 J=1, N

IF (J.GT.1) THEN

DO 14 I=1, J-1

SUM=ALUD (I, J)

IF (I.GT.1) THEN

DO 13 K=1, I-1

SUM=SUM-ALUD (I, K) *ALUD (K, J)

13 CONTINUE

ALUD (I, J)=SUM

ENDIF

14 CONTINUE

ENDIF

AAMAX=0.

```

DO 16 I=J,N
  SUM=ALUD(I,J)
  IF(J.GT.1) THEN
    DO 15 K=1,J-1
      SUM=SUM-ALUD(I,K)*ALUD(K,J)
15    CONTINUE
      ALUD(I,J)=SUM
    ENDIF
    DUM=VV(I)*ABS(SUM)
    IF(DUM.GE.AAMAX) THEN
      IMAX=I
      AAMAX=DUM
    ENDIF
16  CONTINUE
    IF(J.NE.IMAX) THEN
      DO 17 K=1,N
        DUM=ALUD(IMAX,K)
        ALUD(IMAX,K)=ALUD(J,K)
        ALUD(J,K)=DUM
17  CONTINUE
        D=-D
        VV(IMAX)=VV(J)
      ENDIF
      INDX(J)=IMAX
      IF(J.NE.N) THEN
        IF(ALUD(J,J).EQ.0.) ALUD(J,J)=TINY
        DUM=1./ALUD(J,J)
        DO 18 I=J+1,N
          ALUD(I,J)=ALUD(I,J)*DUM
18  CONTINUE
        ENDIF

```

```

19 CONTINUE
   IF (ALUD(N,N).EQ.0.) ALUD(N,N)=TINY
   RETURN
   END
   SUBROUTINE LUBKSB(ALUD,N,NP,INDX,BU)
   DIMENSION ALUD(NP,NP),INDX(50),BU(50)

```

C-----Solve the set of linear equations $A.X = B$. Here ALUD is the LU decomposition of the input matrix A, determined by the routine LUDCMP

```

      II=0
      DO 12 I=1,N
         LL=INDX(I)
         SUM=BU(LL)
         BU(LL)=BU(I)
         IF(II.NE.0) THEN
            DO 11 J=II,I-1
               SUM=SUM-ALUD(I,J)*BU(J)
11          CONTINUE
            ELSE IF (SUM.NE.0.) THEN
               II=I
            ENDIF
         BU(I)=SUM
12      CONTINUE
         DO 14 I=N,1,-1
            SUM=BU(I)
            IF(I.LT.N) THEN
               DO 13 J=I+1,N
                  SUM=SUM-ALUD(I,J)*BU(J)
13          CONTINUE
            ENDIF

```

```

        BU(I)=SUM/ALUD(I,I)
14      CONTINUE
        RETURN
        END

        SUBROUTINE MPROVE(A,ALUD,N,NP,INDX,B,X)
        PARAMETER (NMAX=100)
        DIMENSION A(NP,NP),ALUD(NP,NP),INDX(50),B(50),X(50),R(NMA)
        REAL*8 SDP

```

C---- This routine is used to improve a solution X of the linear set of equations $A.X = B$.

```

        DO 12 I=1,N
            SDP=-B(I)
            DO 11 J=1,N
                SDP=SDP+DBLE(A(I,J))*DBLE(X(J))
11          CONTINUE
            R(I)=SDP
12          CONTINUE
            CALL LUBKSB(ALUD,N,NP,INDX,R)
            DO 13 I=1,N
                X(I)=X(I)-R(I)
13          CONTINUE
        RETURN
        END

```

APPENDIX - G -

C---- This program is used for filtering images given coefficients of the 2-D recursive filter in A and B.

```
REAL*4 A(10,10),B(10,10),SUM
REAL*4 X(256,256),Y(256,256),IYT,MIN,MAX
CHARACTER G(256,256)
CHARACTER*13 FILN
WRITE(*,51)
51 FORMAT(' ENTER SIZE OF IMAGE. (128X128,256X256 ETC.)? \')
READ(*,*)NSIZ
WRITE(*,3)
3 FORMAT(' INPUT FILE NAME---> ? \')
READ(*,'(A13)')FILN
OPEN(1,FILE=FILN,FORM='BINARY',STATUS='OLD')
```

C---- The degraded image is read and is written in X.

```
DO 10 I=1,NSIZ
DO 10 J=1,NSIZ
READ(1)G(I,J)
X(I,J)=ICHAR(G(I,J))
Y(I,J)=0.
10 CONTINUE
CLOSE(1)
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,STATUS='OLD')
   WRITE(*,*)' INPUT ORDER OF FILTER'
   READ(*,*) N
   WRITE(*,*) N
   WRITE(*,*)' ENTER NR THE SIZE OF THE SUB-IMAGES.'
   READ(*,*)NR
   WRITE(*,*)'ENTER NNR (1-->0%, 2-->50%) OVERLAPPING'
   READ(*,*)NNR
   NR1=NR-1
   NR2=INT(NR/NNR)
   KK=1
   NNI=1
   NNJ=1

```

C---- The coefficients $\{a_{ij}\}$ and $\{b_{ij}\}$ of the 2-D IIR filter are read and written in A and B.

```

2000 DO 143 I=1,N+1
      DO 143 J=1,N+1
      READ(7,*)A(I,J)
143  CONTINUE
      DO 144 I=1,N+1
      DO 144 J=1,N+1
      READ(7,*)B(I,J)
144  CONTINUE
      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---- Filtering procedure starts here.

```
DO 1 IN=NNI,NNI+NR1
DO 1 IM=NNJ,NNJ+NR1
  SUM=0.0
  DO 2 I=1,N+1
  DO 2 J=1,N+1
    I1=I-1
    J1=J-1
    IF(IN.LT.I.OR.IM.LT.J)GO TO 2
    SUM=SUM+B(I,J)*X(IN-I1,IM-J1)
    IF((I.EQ.1).AND.(J.EQ.1))GOTO 2
    SUM=SUM-A(I,J)*B(IN-I1,IM-J1)
  2 CONTINUE
  IF(Y(IN,IM).NE.0.)THEN
    Y(IN,IM)=(Y(IN,IM)+SUM)/2.
  ELSE
    Y(IN,IM)=SUM/A(1,1)
  ENDIF
1 CONTINUE
  KK=KK+1
  NNJ=NNJ+NR2
  IF(NNJ.LE.NSIZ-NR1)GO TO 2000
  NNJ=1
  NNI=NNI+NR2
  IF(NNI.LE.NSIZ-NR1)GO TO 2000
  MIN=0.
  MAX=255.
```

```
DO 15 I=1,NSIZ
DO 15 J=1,NSIZ
  IF(Y(I,J).LT.0.)Y(I,J)=0.
  IF(Y(I,J).GT.255.)Y(I,J)=255.
15 CONTINUE
WRITE(*,*)'MAX=',MAX,'MIN=',MIN
DO 16 I=1,NSIZ
DO 16 J=1,NSIZ
  NX=Y(I,J)
  G(I,J)=CHAR(NX)
  WRITE(2)G(I,J)
16 CONTINUE
CLOSE(2)
CLOSE(7)
STOP
END
```

APPENDIX - H -

A- IMAGE SEGMENTATION.

1- Introduction.

The purpose of segmentation is to transform a continuous-tone image (gray-levels 0,1,...,255) to a binary (or multi) level image, and at the same all necessary features of the original image must be retained.

In this appendix we will consider the segmentation problem as a point dependent process. This category deals with methods which are based on examining an image on a pixel-by-pixel basis. There exist another category which deals with techniques which utilize the image information in a prescribed neighborhood. However in this appendix we will consider only the first category.

2- Point-dependent techniques.

2.1. GRAY-LEVEL THRESHOLDING.

This approach consists on dividing the gray-level scale into bands, and then using thresholds to determine regions or to obtain boundary points. The threshold value, is the gray-level value which separate between the object and the background of an image. All pixels with gray-level value below the threshold value T , shown in Fig.(H.1) are declared as '0' and the levels above T as '255'(this is valid for the binary transformation). This technique is called single-level thresholding and T is known as the threshold value of the image. However there might be cases

where the image might contain more than two distinct populations, as shown by the graph given in Fig.(H.2). In these cases, it is then required to group the levels into more than two populations.

Then we have a multi-level thresholding with different threshold values (for the histogram given as an example in Fig.(H.2), we have two threshold values T_1 and T_2).

For the single level thresholding the objective is to select T (threshold value), such that the band B_1 (see Fig.(H.1)) will contain as closely as possible, levels associated with the background, while B_2 will contain the levels of the object.

Threshold selection for images whose histogram are of shape as the one given in Fig.(H.1) or (H.2) is quite straightforward. The threshold is selected at the bottom of the valley between two peaks [46]. There exist methods, which consist of transforming the histogram of an image (when the task of selecting a good threshold would be quite difficult) to a shape where threshold selection would not pose a problem, as the ones shown in Figs.(H.1), and (H.2). Such techniques have been investigated by several authors for more details refer to [21], [47-50].

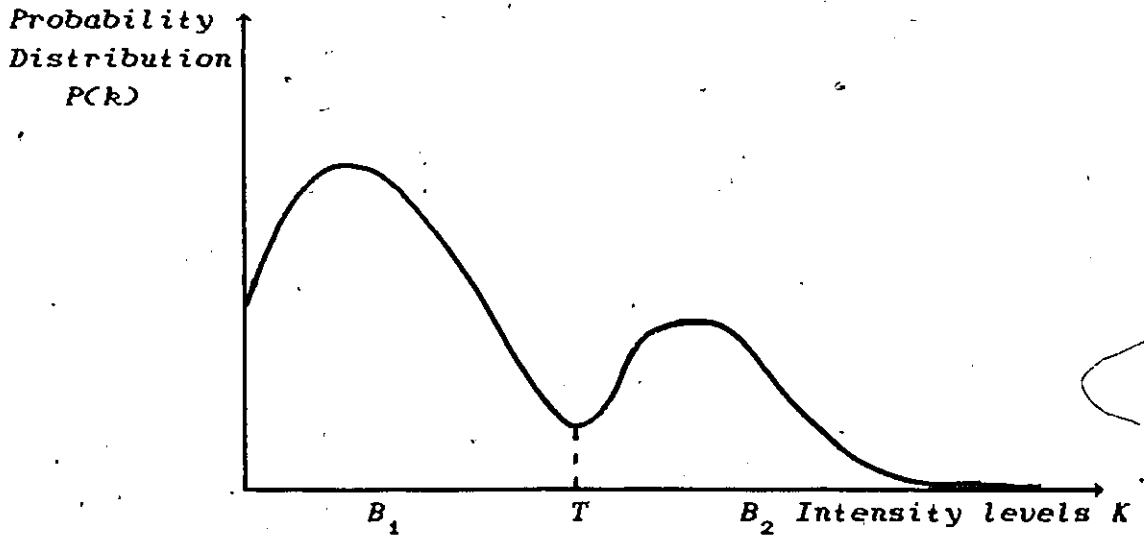


FIG. H.1 A SAMPLE HISTOGRAM ILLUSTRATING A BI-MODAL DISTRIBUTION.

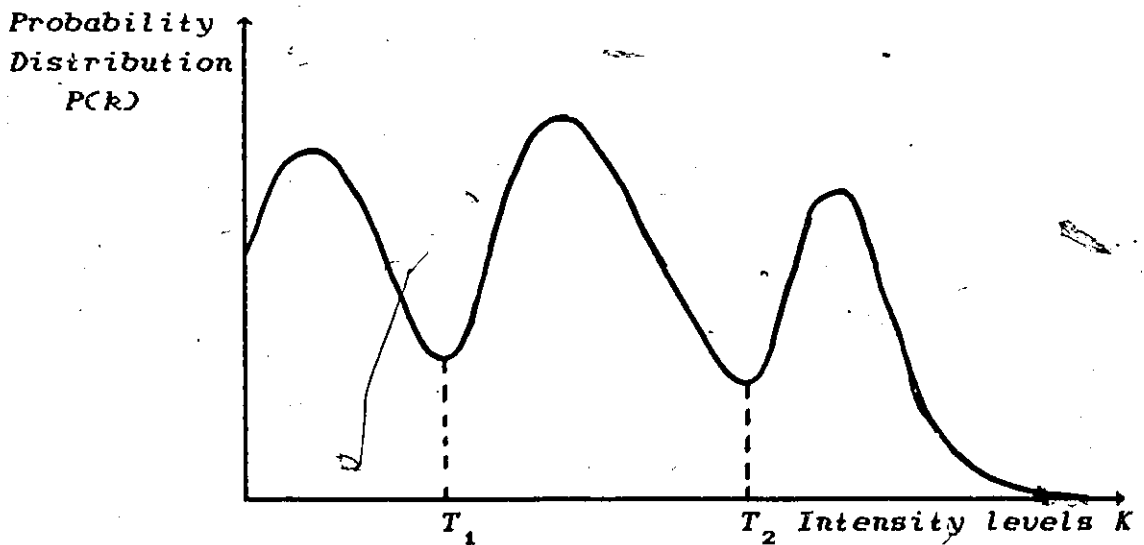


FIG. H.2 A SAMPLE HISTOGRAM ILLUSTRATING A MULTI-MODAL DISTRIBUTION.

VITA AUCTORIS

Mouloud M MAMERI

December	1960	Born in Pont Sainte Maxence (France)
June	1979	Completed High School (Baccalaureat) Lycee Abane Ramdane (Algiers) ALGERIA
January	1985	Graduated in Electronics (Ingenieur) Ecole Polytechnique d'Alger, ALGERIA
January	1986	Admitted in the Master's program, Department of Electrical Engineering University of Windsor
April	1988	Candidate for the degree of M.A.Sc. Electrical Engineering, University of Windsor Windsor, Ontario, Canada N9b 3P4