PARAMETER ESTIMATION FOR THE BLIND
RESTORATION OF BLURRED IMAGERY

THESIS

Murat Cincioglu
Lieutenant, Turkish AF

AFIT/GE/ENG/86S-1
PARAMETER ESTIMATION FOR THE BLIND
RESTORATION OF BLURRED IMAGERY

THESIS

Murat Cincioğlu
Lieutenant, Turkish AF

AFIT/GE/ENG/86S-1

September 1986

Approved for public release; distribution unlimited.
PARAMETER ESTIMATION FOR THE BLIND
RESTORATION OF BLURRED IMAGERY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment for the
Master of Science in Electrical Engineering

Murat Cincioglu
Lieutenant, Turkish AF

September 1986

Approved for public release; distribution unlimited.
Acknowledgements

I wish to express my deepest gratitude and appreciation to my academic and thesis advisor, Major David A. King who kindly offered invaluable help and guidance throughout my course of study in the AFIT and preparation of this thesis.

I would also like to thank the professors and officers of the United States Air Force Institute of Technology for the consistent professionalism and the friendship which they have shown to me during my course of study in the United States of America.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vi</td>
</tr>
<tr>
<td>List of Symbols</td>
<td>vii</td>
</tr>
<tr>
<td>Abstract</td>
<td>viii</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>Statement of the Problem</td>
<td>2</td>
</tr>
<tr>
<td>Scope</td>
<td>2</td>
</tr>
<tr>
<td>Assumptions</td>
<td>3</td>
</tr>
<tr>
<td>Approach</td>
<td>4</td>
</tr>
<tr>
<td>Materials and Equipment</td>
<td>4</td>
</tr>
<tr>
<td>II. Image Restoration Theory</td>
<td>5</td>
</tr>
<tr>
<td>Linear Space Invariant Systems</td>
<td>5</td>
</tr>
<tr>
<td>Imaging System</td>
<td>5</td>
</tr>
<tr>
<td>Image Restoration System</td>
<td>12</td>
</tr>
<tr>
<td>OTF of Some Specific Degradations</td>
<td>12</td>
</tr>
<tr>
<td>OTF of Linear Camera Motion</td>
<td>12</td>
</tr>
<tr>
<td>OTF of Out-of-Focus Lens System</td>
<td>17</td>
</tr>
<tr>
<td>Noise Process</td>
<td>23</td>
</tr>
<tr>
<td>Restoration Methods</td>
<td>26</td>
</tr>
<tr>
<td>Inverse Filter</td>
<td>26</td>
</tr>
<tr>
<td>Wiener Filter</td>
<td>29</td>
</tr>
<tr>
<td>Power Spectrum Equalization Filer</td>
<td>33</td>
</tr>
<tr>
<td>III. Parameter Estimation Techniques</td>
<td>35</td>
</tr>
<tr>
<td>Estimation of Degrading OTF</td>
<td>35</td>
</tr>
<tr>
<td>Method of Frequency Domain Inspection</td>
<td>35</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>38</td>
</tr>
<tr>
<td>Line Estimation of the OTF</td>
<td>45</td>
</tr>
<tr>
<td>Logarithmic Estimate</td>
<td>50</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Estimation of the NSR</td>
<td>50</td>
</tr>
<tr>
<td>Estimation in the Spatial Domain</td>
<td>53</td>
</tr>
<tr>
<td>Estimation in the Frequency Domain</td>
<td>55</td>
</tr>
<tr>
<td>IV. Results</td>
<td>58</td>
</tr>
<tr>
<td>V. Conclusions and Recommendations</td>
<td>77</td>
</tr>
<tr>
<td>Conclusions</td>
<td>77</td>
</tr>
<tr>
<td>Recommendations</td>
<td>79</td>
</tr>
<tr>
<td>Appendix A - Pictures</td>
<td>A-1</td>
</tr>
<tr>
<td>Appendix B - Software Listing</td>
<td>B-1</td>
</tr>
<tr>
<td>Bibliography</td>
<td>80</td>
</tr>
<tr>
<td>Vita</td>
<td>83</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Generalized Imaging System</td>
</tr>
<tr>
<td>2</td>
<td>Image Restoration System</td>
</tr>
<tr>
<td>3</td>
<td>Imaging System Without Defect of Focus</td>
</tr>
<tr>
<td>4</td>
<td>Imaging System with Defect of Focus</td>
</tr>
<tr>
<td>5</td>
<td>The Degrading (Blurring) Process</td>
</tr>
<tr>
<td>6</td>
<td>Filtering Operation</td>
</tr>
<tr>
<td>7</td>
<td>The Zeros of $G(u,v)$</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Restored Pictures and R Values of Eq. (155)</td>
<td>64</td>
</tr>
<tr>
<td>Table 2</td>
<td>Restored Pictures and Noise Variances</td>
<td>69</td>
</tr>
</tbody>
</table>
List of Symbols

\[ f(x,y) \] Original (undegraded) image.

\[ F(u,v) \] Discrete Fourier transform of the original image.

\[ f(x,y) \] Estimated (restored) image.

\[ F(u,v) \] Discrete Fourier transform of the restored image.

\[ h_d(x,y) \] Point-spread-function (PSF) of the degrading system.

\[ H_d(u,v) \] Optical Transfer Function (OTF) of the degrading system.

\[ h_r(x,y) \] Point spread function of the restoration system.

\[ H_r(u,v) \] Optical Transfer Function of the restoration system.

\[ g(x,y) \] Degraded image.

\[ G(u,v) \] Discrete Fourier Transform of the degraded image.

\[ n(x,y) \] Noise.

\[ N(u,v) \] Discrete Fourier transform of \( n(x,y) \).
Abstract

Methods to estimate the optical transfer function (OTF) of the degradation and the noise-to-signal ratio in a degraded image are presented which use no prior information about the degradation and the noise. Two types of degradations are considered: uniform linear camera motion and out-of-focus camera with circular aperture. The OTF of the out-of-focus camera is estimated by inspection of the spectra of the degraded image and from average over the spectra of subsections of the degraded image. A line estimation method is described to estimate the degrading OTF of the linear camera motion in one direction. An additional method called the "logarithmic estimate" is also developed. Frequency and space domain estimation methods are developed for the noise-to-signal ratio for the case of additive gaussian noise. Inverse, Wiener, and power spectrum equalization filters based on these estimates are implemented on a digital computer to restore a variety of degraded images and examples of blind restoration of these degraded images are presented.
PARAMETER ESTIMATION FOR THE
BLIND RESTORATION OF
BLURRED IMAGERY

I. Introduction

Background

One of the major applications of digital image processing is image restoration. Image restoration is an estimation process that attempts to recover an original (ideal) image from a degraded image. These degradations are incurred while the image was being acquired and may include the blurring introduced by optical systems, atmospheric turbulence, camera/object motion, as well as distortion and noise due to electronic and photometric sources.

The overall performance of image restoration process greatly benefits from modelling of the transfer function of the degradation and characterization of the noise in the degraded image. If a prior knowledge of the image and the imaging system is available, it can be used to increase the performance of the restoration system. If little or nothing is known about the image, one can attempt to model and characterize the sources of degradation (blurring and noise) and subsequently remove or reduce their effects. When a prior knowledge about the image is not available, information about the degradation must be extracted from the observed (degraded) image. This task is called blind image restoration.
The Foreign Technology Division (FTD) at Wright-Patterson AFB has needed a blind image restoration method to deblur photographs taken under controlled conditions. Most of the restoration filters such as Wiener filter, inverse filter, power spectrum equalization filter require the degrading optical transfer function and/or the noise-to-signal ratio as their parameters. In order to restore a degraded image, one must know the degrading OTF (optical transfer function) and the NSR (noise-to-signal power ratio). Since no prior information about these two parameters is available, one has to estimate them from the degraded image itself. Once the degrading OTF and the NSR are estimated, any of the restoration filters (methods) can be employed to restore the degraded image. Therefore, the problem of the blind restoration reduces to estimation of the degrading OTF and the NSR.

Statement of the Problem

The problem that is addressed in this thesis may be stated as follows:

Given a degraded image with no information available concerning the degradation function and the noise, develop methods to estimate the degrading OTF and the NSR and use them to restore the degraded image.

Scope

This thesis is primarily concerned with the estimation of the degrading OTF and the NSR. Inverse filter, Wiener filter, and power
spectrum equalization filter are presented as restoration methods. Other restoration techniques and the investigation of their performance with the estimated degrading OTF and the NSR are considered out of the scope of this thesis. Two specific cases of the degradation will be considered: uniform camera motion and out-of-focus lens system with circular aperture. Restoration for other forms of degradation, such as blurring due to atmospheric turbulence, camera vibration, geometric distortion, and system nonlinearities will not be addressed. The primary objectives of this effort can be stated as:

1. Develop a method to estimate the degrading OTF.
2. Develop a method to estimate the noise-to-signal power ratio.
3. Implement the results by a general purpose restoration filter on a digital computer.

Assumptions

The imaging system and the degradation system will be assumed as linear space-invariant systems. We will assume that the original image and the noise are spatial stochastic random processes and that they are spatially stationary and statistically independent. The noise process is assumed to be additive gaussian process with mean value of zero.
Approach

Presentation of this research will proceed in Chapter 2 with a discussion of general theory concerning the linear space invariant system, image formation system, the degrading optical transfer functions, noise process and image restoration methods. This is followed in Chapter 3 with discussions of the methods to estimate the degrading OTF for the cases of uniform linear camera motion blur and out-of-focus camera blur and the noise-to-signal power ratio. In Chapter 4, we discuss the implementation of these estimation methods and their performance in actual degraded images. In Chapter 5, we summarize our findings and recommend topics for continuing research.

Materials and Equipment

All thesis research was accomplished in the Signal Processing Laboratory at AFIT utilizing the Data General Eclipse S/250 and Nova computers and the Octek Image Analyzer. The digital images used in this research were 256x256 monochrome images with four bits per pixel. The images were acquired by a variety of techniques, including optical scanning/digitization of photographic film, frame grabbing from a vidicon, and computer generation. Programming was accomplished using FORTRAN IV and FORTRAN 5. Existing software was used for image acquisition and display, two dimensional Fourier transformation and inversion, and image file input/output.
II. Image Restoration Theory

Linear Space Invariant Systems

We can think of a system as a mapping of a set of input functions into a set of output functions. For the case of electrical networks, the input and output functions are real functions (voltages or currents) of a one-dimensional independent variable (time): for the case of imaging systems, the inputs and outputs can be real-valued functions (intensity) or complex-valued functions of a two-dimensional independent variable (space).

A single-input single-output two-dimensional system can be represented by a mathematical operator "P", which operates on an input function \( f(x,y) \) to produce an output function; \( g(x,y) \), i.e.,

\[
g(x,y) = P\{f(x,y)\} \tag{1}
\]

The system is linear, if the superposition property is obeyed for all input functions \( f_1(x,y) \) and \( f_2(x,y) \), and all complex constants "a" and "b";

\[
P\{af_1(x,y) + bf_2(x,y)\} = aP\{f_1(x,y)\} + bP\{f_2(x,y)\} \tag{2}
\]

The great importance of linearity is the ability to express the response to a complicated input by first decomposing this input into a linear combination of elementary functions and then taking
the same combination of these elementary responses. Such a decomposition may be obtained by the sifting property of the delta function which states that

\[
f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\gamma, \beta) \delta(x-\gamma, y-\beta) \, d\gamma \, d\beta \quad (3)
\]
\(f(\gamma, \beta)\) is considered as weighting factor applied to \(\delta(x-\gamma, y-\beta)\).

Combining Equations (1) and (3), we obtain

\[
g(x, y) = P\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\gamma, \beta) \delta(x-\gamma, y-\beta) \, d\gamma \, d\beta \} \quad (4)
\]

If the system is a linear system, then applying linearity property to Equation (4), we can obtain

\[
g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\gamma, \beta) P\{ \delta(x-\gamma, y-\beta) \} \, d\gamma \, d\beta \quad (5)
\]

Now, let the \(h(x, y; \gamma, \beta)\) be the response of the system at point \((x, y)\) of the output space to a delta function input at coordinates \((\gamma, \beta)\) of the input space, thus,

\[
h(x, y; \gamma, \beta) = P\{ \delta(x-\gamma, y-\beta) \} \quad (6)
\]

This function is called the point spread function (PSF) of the system. The system input and output now can be related to each other by the following equation

\[
g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\gamma, \beta) h(x, y; \gamma, \beta) \, d\gamma \, d\beta \quad (7)
\]
This expression is known as the superposition integral.

A system is said to be space-invariant (shift-invariant or isoplanaric), if the behavior of the system is not a function of the independent variable (space). Thus if the linear system is also space-invariant, then the impulse response of the system will be

\[ h(x,y; \gamma, \beta) = h(x-\gamma, y-\beta) \quad (8) \]

and, we can express the input output relationship for a linear space invariant system as follows

\[ g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\gamma, \beta) h(x-\gamma, y-\beta) \, d\gamma d\beta \quad (9) \]

Equation (9) states that the output of a two-dimensional linear, shift-invariant system is the two-dimensional convolution of the input function with the point spread function of the system.

**Imaging System**

An imaging system consisting of lenses, prisms, mirrors and so on can be considered as a black box which provides a transformation or mapping of input light distribution or input energy radiated from the object to some output spatial light distribution or radiant energy.
Such an imaging system can be characterized by an entrance pupil, representing a finite aperture through which radiant energy (light) passes through to reach imaging elements, and an exit pupil representing again a finite aperture which radiant energy must pass as it leaves the imaging elements to reach the image plane.

In Figure 1, a point source in the object plane at coordinates \((x, y)\) of intensity \(f(x, y)\) radiates energy toward the imaging system entrance pupil. The energy coming from the system exist pupil produces an intensity distribution \(g(x_i, y_i)\) in the image plane. It is assumed that passage of the energy between the entrance and exit pupil can be described by geometrical optics.

In most image formation systems the energy radiation emitted by an object arises from transmitted or reflected light from an incoherent light source. The energy radiation can often be regarded
as quasimonochromatic in the sense that the spectral bandwidth of the energy radiation detected at the image plane is small with respect to the center of the radiation. Under these assumptions, the imaging system of Figure 1 will respond as a linear system in terms of the intensity of its input and output fields (Ref. 1). The relationship between the image intensity and the object intensity for the imaging system can then be expressed by the superposition integral; i.e.,

\[
g(x_i, y_i) = \int \int h(x_i, y_i; x, y) f(x, y) \, dx \, dy \quad (10)
\]

where \( h(x_i, y_i; x, y) \) is the image intensity response of the system at point \((x_i, y_i)\) of the image plane to a point source of light at coordinates \((x, y)\) of the object plane. If the intensity impulse response is space invariant, the input output relationship can be expressed by convolution equation

\[
g(x_i, y_i) = \int \int h(x_i-x, y_i-y) f(x, y) \, dx \, dy \quad (11)
\]

If we take Fourier transform of the both sides of Eq. (11)

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

where \( H(u, v) \) is called the optical transfer function (OTF), and the magnitude \(|H(u, v)|\) of the OTF is known as the modulation transfer function of the system (Ref. 1).
An image restoration system can be modeled as in Figure 2. The original (ideal) image \( f(x,y) \) is degraded by the operation \( h_d(x,y) \) and a noise \( n(x,y) \) added to form the degraded image \( g(x,y) \). This is convolved with the restoration filter point spread function \( h_r(x,y) \) to produce the restored image \( \hat{f}(x,y) \).

The degradation system models the source of degradation in the image. There are many sources of degradation. Some types do not involve blur, while effecting only the gray levels of the individual picture points. Other types which do involve blur are called spatial degradations. Such degradations can be introduced by atmospheric turbulence, aberrations of the camera and the object. We will only be concerned with the spatial degradations which are due to uniform linear camera motion and out-of-focus lens system. The degradation system and restoration system can be considered as the imaging system of Figure 1. Therefore, these systems will respond as a linear system in terms of the
intensity of their input and output fields. If the degradation system models an out-of-focus camera, then this system will introduce a defocusing at the output. A shift in the position of its input will not effect the defocusing introduced by the system. The only effect caused by a shift in the position of its input will be an equal shift in the position of its output, i.e., it will introduce the same amount of defocusing. This may be an idealization when applied to physical systems. But, we will assume that the degradation and restoration systems are linear space-invariant systems. Therefore, degraded and restored images can be expressed as

\[
g(x,y) = \int \int h_d(x-\gamma, y-\beta) f(\gamma, \beta) \, d\gamma d\beta + n(x,y) \tag{13}
\]

and

\[
f(x,y) = \int \int h_r(x-\gamma, y-\beta) g(\gamma, \beta) \, d\gamma d\beta \tag{14}
\]

By taking the Fourier transforms of both sides of these equations, we obtain

\[
G(u,v) = H_d(u,v) F(u,v) + N(u,v) \tag{15}
\]

\[
\hat{F}(u,v) = H_r(u,v) G(u,v) \tag{16}
\]

where \(G(u,v), F(u,v), \hat{F}(u,v), H_d(u,v), H_r(u,v), N(u,v)\) are the Fourier transforms of \(g(x,y), f(x,y), \hat{f}(x,y), h_d(x,y), h_r(x,y), n(x,y)\) respectively. The function \(H_d(u,v)\) is the transfer function of the
system that transforms ideal (original) image, \( f(x,y) \), into the degraded image \( g(x,y) \). The function \( H_r(u,v) \) is the transfer function of the restoration system which makes as good an estimate as possible of the original image \( f(x,y) \).

In order to determine such a transfer function, \( H_r(u,v) \) we need some form of information about the degrading transfer function. In some cases, the form of the degradation can be used for determining the \( H_d(u,v) \). Therefore, we will, first, restrict our attention to some specific forms of degradations and their transfer functions. We will later consider the case of blind restoration where we assume no prior knowledge of the degradation and must estimate the degrading transfer function from the degraded image.

**OTF of Some Specific Degradations**

Some common forms of degradations can be characterized by specific transfer functions. If we know the degradation form, we can obtain the parameters of that degrading transfer function from the degraded image, and use these parameters to obtain the same form of degrading transfer function. We will examine the degrading transfer functions which are produced by linear camera motion, and out-of-focus camera.

**OTF of Linear Camera Motion**

A relative motion between camera and object produces a blur in the image in the direction of the motion. We will assume that the image is invariant in time except for the motion. If a point \( K_i \) is
selected in the image plane, the instantaneous image intensity at this point will be denoted by \( g(K_i,t) \). The total exposure at any point of the recorded image can be described by the integrated image intensity at this position (Ref. 2).

\[
g(K_i) = \int_0^T g(K_i,t) \, dt \tag{17}
\]

where \( T \) is the total exposure time. The medium that actually performs the integration could be photographic film or the phosphor in the camera sensor. Assuming that energy is conserved by the imaging system at any instant during the exposure, the energy radiated and collected by a small area in the object plane around the object point \( K_o \) is described by

\[
g(x_i,y_i) \, dx_i \, dy_i = f(x_o,y_o) \, dx_o \, dy_o \tag{18}
\]

at a fixed time \( t \). And, we can say that

\[
g(K_i,t) = f(K_o,t) \tag{19}
\]

and time integral can be written as

\[
g(K_i) = \int_0^T f(K_o,t) \, dt \tag{20}
\]

This relation shows that a recorded image point \( g(K_i) \) over the exposure time \( T \) will be obtained by performing a summation over all object points that are imaged onto the image point. If there were no motion of either object or camera, this relationship would become
\[ g(K_j) = f(K_o, t) T \quad (21) \]

Since we have assumed that object is a two-dimensional entity and the image is time invariant except for the motion, we can rewrite the recorded image intensity as

\[ g(x_i, y_i) = \int_0^T f[x_o(t), y_o(t)] \, dt \quad (22) \]

Given a description of the object motion \( R(t) \), we can convert this time integral into a positional integral over an equivalent stationary object. Let the path of the object motion be described in two-dimensional object plane as a parametric function of the time

\[ R(t) = [x_o(t), y_o(t)] \quad (23) \]

The elemental length \( ds \) of the path \( R(t) \) is given by

\[ ds = \left[ \left( \frac{dx_o(t)}{dt} \right)^2 + \left( \frac{dy_o(t)}{dt} \right)^2 \right]^{\frac{1}{2}} dt \quad (24) \]

The time integral in equation (22) can then be changed into the line integral of the form

\[ g(x_i, y_i) = \int_{R(0)}^{R(T)} f[x_o(t), y_o(t)] ds \quad (25) \]

\[ \left[ \left( \frac{dx_o(t)}{dt} \right)^2 + \left( \frac{dy_o(t)}{dt} \right)^2 \right]^{-\frac{1}{2}} \]
which is the general expression for motion degradation with linear space invariant imaging system. The space invariant point spread function can be identified in Equation (25) as

$$h(x_i, y_i; x_0, y_0) = \left[ \left( \frac{dx_o(t)}{dt} \right)^2 + \left( \frac{dy_o(t)}{dt} \right)^2 \right]^{-\frac{1}{2}}$$

(26)

Where $x_o(t), y_o(t)$ are valid over the path of the object motion, that is,

$$x_o(0) \leq x_o(t) \leq x_o(T)$$

(27)

$$y_o(0) \leq y_o(t) \leq y_o(T)$$

and $h(x_i, y_i; x_0; y_0) = 0$ elsewhere.

From above discussion, we can find a solution to determine the blurring produced by linear camera motion. We will again describe the image $g(x_i, y_i)$ which is integrated by the camera sensor, as follows

$$g(x, y) = \int_0^T f[x_0 - A(t), y_0 - B(t)] \, dt$$

(28)

Where $A(t)$ and $B(t)$ are the motion in $x$ and $y$ directions respectively. If we assume that camera is moved $\theta$ degrees off the horizontal axis and with a velocity of $V_c$, then the displacement distance in $x$ and $y$ directions at an instant time $t$ will be
\[ A(t) = V_c t \cos \theta \]
\[ B(t) = V_c t \sin \theta \]

and

\[ g(x,y) = \int_0^T f(x-V_c t \cos \theta, y-V_c t \sin \theta) \, dt \]  
(30)

Taking Fourier transform of the both sides of Eq. (30) we obtain

\[ G(u,v) = \int_0^T \int_0^\infty \int_0^\infty f(x-V_c t \cos \theta, y-V_c t \sin \theta) \, dx \, dy \, dt \]
\[ \exp\{-j2\pi(ux+vy)\} \, dx \, dy \, dt \]  
(31)

Making the change of variables

\[ \varepsilon = x-V_c t \cos \theta \]
\[ \eta = y-V_c t \sin \theta \]  
(32)

we then have

\[ G(u,v) = \int_0^T \int_0^{2\pi} \int_0^{2\pi} f(\varepsilon, \eta) \exp\{-j2\pi(\varepsilon u+\eta v)\} \, d\varepsilon \, d\eta \]
\[ \exp\{uV_c t \cos \theta + vV_c t \sin \theta\} \, dt \]  
(33)

If we let \( d \) be the total displacement made by the camera in the interval \((0,T)\), then
d = V_c T \quad \text{or,} \quad V_c = \frac{d}{T} \quad (34)

and if we define "f" as
\[ f = u \cos \theta + v \sin \theta \] (35)

then, Eq. (33) becomes
\[ G(u,v) = F(u,v) \int_0^T \exp\{-j2\pi f \frac{d}{T} t\} \, dt \quad (36) \]

as we can see, the OTF of the degradation equals to
\[ H_d(u,v) = \int_0^T \exp\{-j2\pi f \frac{d}{T} t\} \, dt \quad (37) \]

Performing the integration in Eq. (37), we have
\[ H_d(u,v) = \frac{T}{\pi fd} \sin \pi fd \, e^{-j\pi fd} \quad (38) \]
or
\[ H_d(u,v) = T \text{Sinc}(fd) \, e^{-j\pi fd} \quad (39) \]

where \( \text{Sinc}(fd) = \frac{\sin \pi fd}{\pi fd} \)

**OTF of Out-of-Focus Lens System**

An aberration-free (diffraction limited) imaging system converts a diverging spherical wave into a spherical wave, that converges toward a point in the image plane. An aberration, such as defocusing, causes the exit wave to depart from its ideal spherical shape. In Figure 3,
0 is an axial object point in the object plane and I is the geometrical image of that point in the image plane. The energy radiated from the point object 0 passes through the entrance and exit pupils and makes an angle A with the optical axis in the image plane. The references s. and s. are centered at 0 and I, respectively, and are of such radii that they intersect the optical axis at the entrance and exit pupils, respectively.

Figure 3 Imaging System Without Defect of Focus (Ref. 8)
In Figure 4, $s'$ is the converging wave front exiting the exit pupil with a different spherical shape from its ideal shape and forms the image at the out-of-focus point $P$. The wave front $s_2$ converges to the in-focus point $I$. We can specify the amount of defocusing by the out-of-focus distance $Z$ or the optical distance $W$ between $s_2$ and $s'$.

The pupil function which describes the defocusing effect on the ideal spherical wave front is of the form (Ref. 8)

$$
P(x,y) = \begin{cases} 
\exp\{jkw(x^2 + y^2)\} & \text{, if } x^2 + y^2 \leq 1 \\
0 & \text{, if } x^2 + y^2 > 1 
\end{cases} \quad (40)
$$
where \( k = 2\pi/\lambda \) and \( W \) is the optical distance between the emergent wave front and the ideal spherical wave front measured along the extreme ray, and a convenient measure of the defect of focus. \( \lambda \) is the wave length. The coordinates \((x,y)\) at exit pupil are normalized so that \( x^2 + y^2 = 1 \) at the edge of the pupil.

If the system has a circular aperture, then the OTF for such a system is a function of only one spatial frequency variable

\[
H(s) = \frac{\iint_{-\infty}^{\infty} P(x + \frac{s}{2}, y) P^*(x - \frac{s}{2}, y) \, dx \, dy}{\iint_{-\infty}^{\infty} |P(x,y)|^2 \, dx \, dy} \tag{41}
\]

The frequency \( s \) is a reduced frequency and related to normalized frequency \( f = \sqrt{u^2 + v^2} \) by

\[
s = \left( \frac{\lambda}{n \sin \alpha} \right) f \tag{42}
\]

where \( n \) is the refractive index and taken to be 1 for all calculations in this study and the angle \( \alpha \) is defined in Figure 4. The integrand in Equation (41) vanishes for \( s \) values outside the range \( 0 \leq |s| \leq 2 \). The OTF for a defocused system can be found by inserting the pupil function of Eq. (40) into Eq. (41).
The exact OTF obtained in Eq. (43) is difficult to calculate but, for small amounts of defocusing and small angles $\Delta$, this function can be approximated to the geometrical OTF. The geometrical PSF of a defocused aberration free system with a circular aperture is simply a circular path of light of radius $R$ (Ref. 14), that is
The fourier transform of Eq. (45) will give us the geometrical OTF of the defocused system.

\[ H_d(f) = \frac{J_1(2\pi rf)}{\pi rf} \] (46)

It is convenient to express the OTF in terms of the optical path length \( W \) and the normalized frequency \( s \), because, the OTF can then be calculated without knowledge of the optical parameters of the system. A relation between \( W \) and the out-of-focus distance \( Z \) is required to calculate the OTF for a specific system. From the triangle IKP in Figure 4, we can write

\[ (KP)^2 = (KI)^2 + (IP)^2 - 2(AI)(IP) \cos (\pi - A) \] (47)

This can be written as

\[ (r'+Z)^2 = (r'+w)^2 + Z^2 - 2(r'+w)Z \cos (\pi - A) \] (48)

where the image distance \( OI = r' \). Solving for \( W \) gives

\[ W = -r' - Z \cos A + (r'^2 + 2r'Z + Z^2 \cos A)^{\frac{1}{2}} \] (49)

This equation can be approximated for small angles and small amounts of defocusing by
\[ W = Z (1 - \cos A) \quad (50) \]

By using the trigonometric small angle approximation \( \sin \frac{A}{2} = \frac{1}{2} \sin A \), we find

\[ W = \frac{1}{2} Z \sin^2 A \quad (51) \]

We can again approximate the argument \( x \) of Bessel function in Eq. (46) by using Equations (41) and (51)

\[ x = 2\pi r f \]
\[ = \lambda \pi w s / \lambda \cos A \]
\[ = (4\pi / \lambda) w s \]
\[ x = a \quad (52) \]

Therefore,

\[ H_d(s) = 2 \frac{J_1(x)}{x} \quad (53) \]

This equation should be normalized so that \( H_d(0) = 1 \), and represents the approximate OTF for a defocused optical system with circular aperture.

**Noise Process**

Observations of an image field are subject to measurement errors, irregularities of the recording media, and atmospheric light fluctuations. These phenomena give rise to a noise in the image. Noise in images can be divided into three categories: film grain
noise, photoelectronic noise, and thermal noise.

Film-grain noise is produced by a photographic emulsion during the process of image recording and reproduction. Under microscopic examination, the smooth tones of a photographic image assume a random, grainy appearance. Randomness is further introduced by the variable number of photons which expose a particular film grain and the varying size of the grains themselves. The subjective appearance of these factors in the image is called film-grain noise. For most practical purposes, film-grain noise can be modeled a Gaussian process (Ref. 4).

Photoelectronic noise is due to the statistical nature of light and of the photoelectronic conversion process inherent in image sensors. A model of this noise is given in Ref. 5.

Electronic noise is due to random thermal motion of electrons in resistive circuit elements and can be modeled as white Gaussian noise process; i.e., it has a flat power spectrum (Ref. 5.).

To restore a picture in the presence of noise, in addition to a knowledge of the degrading optical transfer function, we need to know (at least in theory) both the statistical properties of the noise and how it is correlated with the original image. In practice, the most common assumptions (like in the electronic communications) about the noise are that it is white, i.e., its power spectrum density is constant, and
it is uncorrelated with the original image (signal). Both these assumptions may be questioned. The concept of white noise is a mathematical abstraction, however, it is a convenient and reasonably accurate model, if the power spectrum density of the original image falls off much faster than the power spectrum density of the noise. As far as the noise and the original image being uncorrelated is concerned, there are examples of the noise which is correlated with the original image, e.g., in the case of film-grain noise (Ref. 4, 5). We will be mainly concerned with the noise which is uncorrelated with the original image and additive in the spatial frequency domain. Concerning the power spectrum of the noise, we will consider both cases where the noise is white and where it is colored in the spatial frequency domain. The latter will be modeled as a gaussian random variable with zero mean.

Different restoration methods may require different amounts of information about the noise process. Constraint image restoration methods (Ref. 7, 6) require that only the variance of the noise be known. On the other hand, the Wiener filter and power spectrum equalization filters require the characterization of the process in terms of its power spectral density or the ratio of the noise power spectrum density to the original image power spectrum density which we refer to as the noise-to-signal ratio (NSR). Since we will use Wiener filter and power spectrum equalization filter for restoration, we will estimate the NSR in spatial domain and in the
spatial frequency domain by assuming that the noise is additive and it is spatially stationary stochastic random process with zero mean.

Restoration Methods

As we mentioned earlier, image restoration is an estimation process that attempts to recover an original image from its degradations that were incurred while the image was being obtained. Restoration techniques are methods of obtaining a PSF (point spread function) for the restoration filter, so that the output of the restoration filter is a good estimate of the original image. We will assume that the degraded image $g(x,y)$ and the original image $f(x,y)$ obey the model given in Eq. (13). That is,

$$g(x,y) = \int \int h(x-\gamma, y-\beta) f(\gamma, \beta) \, d\gamma \, d\beta + n(x,y) \quad (54)$$

and Fourier transform of degraded image and restored image are given by Equations (15) and (16) respectively.

$$G(u,v) = H_d(u,v) F(u,v) + N(u,v) \quad (55)$$

$$\hat{F}(u,v) = H_r(u,v) G(u,v) \quad (56)$$

Inverse Filter.

In the absence of noise, Equation (55) reduces to

$$G(u,v) = H_d(u,v) F(u,v) \quad (57)$$
or equivalently

\[ F(u,v) = \frac{G(u,v)}{H_d(u,v)} \]  

(58)

since we want \( \hat{F}(u,v) = F(u,v) \), Eq. (58) implies that

\[ H_r(u,v) = \frac{1}{H_d(u,v)} \]  

(59)

and

\[ \hat{F}(u,v) = H_r(u,v) G(u,v) = \frac{G(u,v)}{H_d(u,v)} \]  

(60)

This means that we can restore the degraded image \( g(x,y) \) by multiplying the Fourier transform \( G(u,v) \) of the degraded image by \( H_r(u,v) \) and then inverse Fourier transforming.

Some problems arise when one wants to use Eq. (60) in practice. There may be points or regions in the Fourier domain where \( H_d(u,v) = 0 \). In the absence of noise, as we can see from Eq. (57), the Fourier transform \( G(u,v) \) of the degraded image will also be zero, leading to undetermined ratios. Therefore, we can say that even in the absence of noise, it is, in general, impossible to restore \( f(x,y) \) exactly, if \( H_d(u,v) \) has zeros in the Fourier domain. In the presence of noise, the zeros of \( G(u,v) \) and \( H_d(u,v) \) will no longer coincide. When noise is present, we have

\[ G(u,v) = H_d(u,v) F(u,v) + N(u,v) \]  

(61)
applying the restoration filter gives

\[
\hat{F}(u,v) = G(u,v) H_r(u,v) = F(u,v) + \frac{N(u,v)}{H_d(u,v)}
\]  

(62)

If \(H_d(u,v)\) is small or zero in any region of the Fourier domain where \(N(u,v)\) is not correspondingly smaller or where \(N(u,v)\) is non zero, then the contribution of the noise will be amplified, thus the term \(N(u,v)H_d(u,v)\) will have much larger magnitude than \(F(u,v)\) and the inverse Fourier transform of \(G(u,v)H_r(u,v)\) will be strongly influenced by these large terms. Instead, it will contain many noiselike variations and will not be meaningful restoration of \(f(x,y)\).

Since the right-hand side of Eq. (60) cannot normally be evaluated numerically due to the zeros of \(H_d(u,v)\), the most we can do is to restore only the frequencies in the Fourier domain where the noise-to-signal ratio is low. Such an alteration in the restoration process means that the exact inverse filter is not being performed; instead, an approximation is being made. This approximation can be improved to some degree by examining the output of the restoration filter. Let us denote the approximated inverse filter transfer function by \(A(u,v)\). We can say that

\[
H_r(u,v) \approx A(u,v)
\]  

(63)

If \(H_d(u,v)\) does not have zeros and noise is not present, then the
restoration filter optical transfer function will be

\[ H_r(u,v) = A(u,v) = \frac{1}{H_d(u,v)} \]  \hspace{1cm} (64)

**Wiener Filter**

The inverse filter performs poorly in the presence of noise since the noise process is not taken into account in the filter design. Inverse filtering also requires tedious manipulation of the inverse of the degrading transfer function. It would be preferable, if the filter itself controlled the noise amplification automatically. One way to design such a filter is to find a restoration filter transfer function that minimizes some measure of the difference between estimated image \( f(x,y) \) and the original image \( f(x,y) \). We will design a restoration filter that minimizes (in a statistical sense) the mean-squared error between the original image \( f(x,y) \) and the estimated image \( f(x,y) \). This filter is called least squares filter or Wiener filter.

Let the original image, the degraded image and the noise be represented by the stochastic random processes \( f(r) \), \( g(r) \), and \( n(r) \) respectively. Where \( r \) is a position in the \( xy \)-plane. Then from the superposition integral in Eq. (9), we can write

\[ g(r) = \int h_d(r-r') f(r') \, dr' + n(r) \]  \hspace{1cm} (65)
where \( dr' \) represents the area element \( dx'dy' \) in the xy-plane and \( h_d(r) \) is the point spread function of the degradation. We need to solve the integral Equation (65) for the original image process \( f(r) \). However, with \( n(r) \) unknown, the Eq. (65) cannot be solved directly. The most we can do is to produce an estimate \( f(r) \) of the solution. We assume that \( f(r) \) and \( n(r) \) can be viewed as spatial stochastic processes, and assume that noise process \( n(r) \) is known. Then, the best estimate is to maximize the posterior probability density of \( f(r) \), given \( g(r) \), as defined Bayes's rule (Ref. 21). If we further assume that the estimate \( f(r) \) is a linear function of the gray levels in the degraded image process \( g(r) \), then the linear estimate \( f(r) \) can be expressed as

\[
\hat{f}(r) = \int h_r(r-r') g(r') \, dr'
\]  

(66)

Our aim is to find a function \( h_r(r) \), which is the point spread function of the restoration filter, such that mean-square error

\[
\epsilon^2 = E\{[f(r) - \hat{f}(r)]^2\}
\]

(67)

is minimized.

A function which satisfies

\[
E\{g(s)[f(r) - \int h_r(r-r') g(r') \, dr']\} = 0
\]

(68)

for all positions \( r \) and \( s \) in the xy-plane will also minimize Eq. (67). From Eq. (68), we can write
\[ \int h_r(r-r') E[g(s) \ g(r')]dr' = E[g(s) f(r)] \]  

(69)

By using the definitions of the autocorrelation and cross-correlation of a random process (Ref. 9), we can re-write Eq. (69) as

\[ \int h_r(r-r') R_g(s,r') \ dr' = R_{gf}(s,r) \]  

(70)

We will further assume that the stochastic processes representing the image \( f(r) \) and the noise \( n(r) \) are (spatially) stationary. Therefore, the autocorrelation function \( R_g(s,r') \) and cross-correlation function \( R_{gf}(s,r) \) can be represented as \( R_g(r'-s) \) and \( R_{gf}(r-s) \), respectively; i.e., eq. (70) becomes

\[ \int h_r(r-r') R_g(r'-s) \ dr' = R_{gf}(r-s) \]  

(71)

for every position \( r \) and \( s \) in the \( xy \)-plane. By making variable changes: \( r'-s = z' \) and \( r-s = z \), we obtain

\[ \int h_r(z-z') R_g(z') \ dz' = R_{gf}(z) \]  

(72)

for all positions \( z \) in the \( xy \)-plane. We denote the coordinates of \( z \) by \((\Delta x, \Delta y)\) and of \( z' \) by \((x,y)\) to obtain

\[ \int \int h_r(\Delta x-x, \Delta y-y) R_g(x,y) \ dx \ dy = R_{gf}(\Delta x,\Delta y) \]  

(73)

As we can see, the left-hand side of Eq. (73) is a convolution that is

\[ h_r(x,y) \ast R_g(x, x+\Delta x; y, y+\Delta y) = R_{gf}(\Delta x, \Delta y) \]  

(74)
or

\[ h_r(x, y) * R_g(\Delta x, \Delta y) = R_{gf}(\Delta x, \Delta y) \quad (75) \]

If we take the Fourier transform of both sides of Eq. (75) we have

\[ H_r(u, v) S_g(u, v) = S_{gf}(u, v) \quad (76) \]

or

\[ H_l(u, v) = \frac{S_{gf}(u, v)}{S_g(u, v)} \quad (77) \]

where \( S_{gf}(u, v) \) and \( S_g(u, v) \) are the Fourier transforms of \( R_{gf}(u, v) \) and \( R_g(u, v) \), respectively. \( R_{gf}(\Delta x, \Delta y) \) is the spatial cross-correlation of the degraded image \( g(x, y) \) and original image \( f(x, y) \) and it is given by

\[ R_{gf}(\Delta x, \Delta y) = E\{g(x, y) f(x+\Delta x, y+\Delta y)\} \quad (78) \]

\[ R_{gf}(\Delta x, \Delta y) = E\{[h_d(x, y) * f(x, y) + n(x, y)] f(x+\Delta x, y+\Delta y)\} \quad (79) \]

Since we assume that image process and noise process are uncorrelated and that the noise process has zero-mean, we have

\[ E\{n(x, y) f(x+\Delta x, y+\Delta y)\} = E\{n(x, y)\} E\{f(x+\Delta x, y+\Delta y)\} = 0 \quad (80) \]

and Equation (79) becomes

\[ R_{gf}(\Delta x, \Delta y) = h_d(x, y) * R_{f}(\Delta x, \Delta y) \quad (81) \]
Taking the Fourier transform of both sides of Eq. (81) gives us

$$S_g(u,v) = H^*(u,v) S_f(u,v)$$  \hspace{1cm} (82)

where $S_f(u,v)$ is the power spectrum of the original image. It can easily be seen from Figure 2 that $S_g(u,v)$ is given by

$$S_g(u,v) = \left| H_d(u,v) \right|^2 S_f(u,v) + S_n(u,v)$$  \hspace{1cm} (83)

where $S_n(u,v)$ is the power spectrum of the noise process. If we insert Equations (82) and (83) into Equation (77), we obtain

$$H_r(u,v) = \frac{H^*(u,v) S_f(u,v)}{\left| H_d(u,v) \right|^2 S_f(u,v) + S_n(u,v)}$$  \hspace{1cm} (84)

Eq. (84) can be written in a more convenient form as

$$H_r(u,v) = \frac{H^*(u,v)}{\left| H_d(u,v) \right|^2 + \frac{S_n(u,v)}{S_f(u,v)}}$$  \hspace{1cm} (85)

This result gives the transfer function of Wiener filter that we will use as the restoration filter transfer function.

**Power Spectrum Equalization Filter.**

We modeled the degraded image as the output of a linear system which has the degrading transfer optical function $H_d(u,v)$. The input to this system is taken to be original image. From linear systems theory,
we can establish a relationship between the original image power spectrum and the degraded image power spectrum. From Figure 2 the power spectrum of the degraded image can be written as

\[ S_g(u,v) = |H_d(u,v)|^2 S_f(u,v) + S_n(u,v) \quad (86) \]

Power spectrum equalization criterion states that the power spectrum of the estimated image be equal to the power spectrum of the original image. That is,

\[ S_f(u,v) = S_f^*(u,v) \quad (87) \]

where \( S_f^*(u,v) \) denotes estimated image power spectrum. It is possible to write \( S_f^*(u,v) \) in terms of \( S_g(u,v) \) and the transfer function of the restoration filter

\[ S_f^*(u,v) = |H_r(u,v)|^2 S_g(u,v) \]

\[ = |H_r(u,v)|^2 [ |H_d(u,v)|^2 S_f(u,v) + S_n(u,v) ] \quad (88) \]

by substituting Equation (87) into Equation (88) and solving for \( H_r(u,v) \) we obtain the transfer function of the restoration filter that performs the power spectral equalization as follows

\[ H_r(u,v) = \left[ \frac{1}{|H_d(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}} \right]^{1/2} \quad (88.1) \]
III. Parameter Estimation Techniques

Estimation of Degrating Transfer Function

We have seen several methods for restoring degraded images. These methods require, as a description of the degradation, the degrading point spread function (PSF) or its Fourier transform, the degrading optical transfer function (OTF). However, in many practical situations, a prior knowledge of the degrading PSF or OTF is not available. Therefore, in order to apply one of these restoration methods, one must first estimate the degrading OTF. Since we assume that we do not have a prior information about the degradation and that the original undegraded image is unavailable, we will consider the estimation of the degrading OTF from the degraded image itself; i.e., so called "blind" restoration.

Method of Frequency Domain Inspection.

The degraded image in the absence of noise can be expressed as in Eq. (57) in the frequency domain. It is given by point-by-point product of the degrading OTF $H_d(u,v)$ and the Fourier transform of the original image. Therefore, on the lines where $H_d(u,v)$ is zero, $G(u,v)$ will also be zero. If magnitude of $G(u,v)$ is displayed, these lines of zero values can be determined, given that $H_d(u,v)$ is not zero in the neighboring areas (Ref. 10). In the presence of noise, the displayed
values may not be exactly zero at these points, but the rough minima corresponding to ideal zeros can still be identified, if the noise is sufficiently small or white.

The identification of the zeros in the OTF may not be sufficient to determine the complete function. However, many of the common causes of the degradation such as the blurs due to a relative motion between camera and object or the out-of-focus camera have distinct zero patterns in their OTF's. We will ignore the unlikely possibility that the zeros of the Fourier transform of the original image have the same zero pattern, we will also assume that the degradation is due to either motion blur or out-of-focus camera. If the restored image looks reasonable, the confidence in the correctness of these assumptions increases. In the case that some prior knowledge of the OTF is available, this knowledge can be used in combination with the above method. For example, if the shape of the OTF is known, then only the frequency scale factor needs to be determined. In fact, in some degraded images, it is possible to determine the degradation form interactively. Once correct type of OTF is determined, the accuracy of its scale factor, in some cases, can be improved by averaging $|G(u,v)|$ or $|G(u,v)|^2$ along the lines parallel to the expected zeros and examining the resulting function.

In using this method, it is desirable to perform a simple linear spatial filtering operation such as differentiation on the degraded image before Fourier transforming it, in order to suppress the low
frequencies and to enhance the high frequencies. In most images, the amplitudes of the low frequency components are much greater than the amplitudes of the high frequency components. The above filtering operation may flatten the frequency spectrum of the degraded image and makes the identification of the zeros easier.

One such filter which meets the above objective is Laplacian operator \( \nabla^2 = \partial^2 / \partial x^2 + \partial^2 / \partial y^2 \). Since the Laplacian operator is linear and shift-invariant, applying it to the picture \( g(x,y) \) is equivalent to convolving the degraded image \( g(x,y) \) with the PSF \( h_d(x,y) \) defined by (Ref. 11).

\[
h_L(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (89)
\]

Visualization of the zeros in the spectrum can also be aided by taking the logarithm of the degraded image power spectrum. This operation may enhance the amplitude of dips due to the degrading transfer function. If the zeros are equally spaced, they produce a series of periodic spikes in the log power spectrum. The power spectrum of the log power spectrum, called the cepstrum, can be used for determining the exact spacing of the spikes and consequently the zeros of the degrading transfer function.

After visualization of the zeros, we need to recognize the zero pattern of the degrading OTF. In the case of linear camera motion, the zeros of the OTF requires a linear pattern. This pattern is the
zero crossing of the sinc function given by Eq. (39). That is, the zero pattern will be

$$f = \frac{1}{d}, \frac{2}{d}, \frac{3}{d}, \ldots$$  \hspace{1cm} (90)

The OTF of out-of-focus camera with circular aperture is given by Eq. (46) and its zeros will form a circular pattern about the origin which can be identified by (Ref. 10).

$$f = \frac{0.61}{r}, \frac{1.117}{r}, \frac{1.619}{r}, \frac{2.121}{r}, \ldots$$  \hspace{1cm} (91)

where $r = (u^2 + v^2)^{\frac{1}{2}}$.

If the ideal image consists of a very simple geometrical shape, then its Fourier transform may have a simple pattern of zeros that could be mistaken for the zeros in the OTF. In this case, the method described here may not work, unless a sufficient prior information to enable the separation of the zeros in the image from the zeros in the OTF is provided. But more-complex images would have more-random patterns of zeros in their Fourier transform, therefore this problem would be less likely to occur.

**Image Segmentation.**

Another approach to estimation of the degrading OTF, which has been studied by many researchers (Ref. 12, 13, 14, 15), is as follows: the degraded image is divided into $M$ regions (subimages) which are all identical in size and they may overlap. If we assume
that the extent of the degrading point spread function \( h_d(x,y) \) is small compared to that of each region of the degraded image, then approximately we have

\[
g_i(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_d(x-\gamma, y-\beta) f_i(\gamma, \beta) \, dy \, dB + n_i(x,y) \quad (92)
\]

where the index \( i \) denotes the \( i \)'th subimage. Since we can pre-process (low-pass filter) the degraded image to reduce the noise effects, we ignore the noise term in the right-hand side of Eq. (92). By taking the Fourier transform of the both sides of Eq. (92), we obtain

\[
G_i(u,v) = H_d(u,v) F_i(u,v) \quad , \quad i = 1, 2, 3, \ldots, M \quad (93)
\]

The functions \( G_i(u,v), F_i(u,v), H_d(u,v) \) are complex in the Fourier domain. Therefore, we may write Eq. (93) as

\[
|G_i(u,v)| e^{j\theta_j(u,v)} = |H_d(u,v)| \cdot |F_i(u,v)| \cdot e^{j[\theta_H(u,v) + \theta_{f_i}(u,v)]}
\]

\[
(94)
\]

By taking the product average of the magnitudes over \( M \) regions, we will have

\[
M \prod_{i=1}^{M} |G_i(u,v)| = \left[ \prod_{i=1}^{M} |F_i(u,v)| \right] \left[ H_d(u,v) \right] M \quad (95)
\]

it is possible to estimate the magnitude of the degrading OTF by (Ref. 13)
\[ |H(u,v)| = \left[ \prod_{i=1}^{M} |G_i(u,v)| \right]^{1/M} / \left[ \prod_{i=1}^{M} |F_i(u,v)| \right]^{1/M} \]  \hspace{1cm} (96)

On the right-hand side of the Eq. (96), the denominator term
\[ \prod_{i=1}^{M} |F_i(u,v)|, \] is not known. However, if the intensity distributions in the M regions of the original image vary sufficiently fast and are sufficiently different from one another, then we can expect that this term will be approximately a constant. We also do not have information about the phase of the estimated OTF. In most cases, it might be reasonable to assume that the phase of
\[ \prod_{i=1}^{M} |F_i(u,v)|^{1/M} \] is approximately constant.

Another way of estimating the OTF by segmentation is to treat the image as a (spatially stationary stochastic process (Ref. 14). From the stochastic linear system relationship, we can write the power spectrum of the degraded image in the i'th region as
\[ S_{g_i}(u,v) = S_{f_i}(u,v) |H_d(u,v)|^2 + S_{n_i}(u,v) \]  \hspace{1cm} (97)

After taking an average over the power spectrum of the subimages (the square of the magnitude of the Fourier transform of the subimages), we have an estimate of the power spectrum of subimages. Such an average will retain much of the flavor of \(|H_d(u,v)|\), since \(|H_d(u,v)|\) is included in each one of the averaged subimages, while each \(F_i(u,v)\) and \(N_i(u,v)\) are varying from subimage to subimage, thus, introducing a much more subtle contribution to the average (Ref. 14). We may now search for
the zeros of $H_d(u,v)$ more successfully than we did in the frequency domain inspection method. Since most degraded images have reduced amount of high frequency power, the search for the zeros of $|H_d(u,v)|$ should be carried out along the dc axes of estimated $S_g(u,v)$. The power cepstrum which is defined as the Fourier transform of the logarithm of the power spectra $S_g(u,v)$ may enable us to identify exact spacing of the zeros of $H_d(u,v)$. The power cepstrum of Eq. (97) is equal to

$$P(u,v) = F\{\log S_f(u,v)\} + 2 F\{\log |H_d(u,v)|\}$$  \hspace{1cm} (98)

Note that we have neglected the noise term. Since the right hand side of Eq. (98) results from an averaging process, $F\{\log S_f(u,v)\}$ retains little of the flavor of the original image while $2 F\{\log |H_d(u,v)|\}$ remains strongly characteristic of the degrading OTF. The rings of spikes, as previously mentioned, results from focus blur and the radius of these rings will be the parameter that we need to know to estimate the degrading OTF. In the case of motion blur, these spikes will be periodic in the direction of the motion and their pattern is given by Eq. (90).

Another approach which is similar to the method which is defined in Ref. 12 may be used for estimating the degrading OTF by segmentation method. By taking the complex logarithm of the both sides of Eq. (93), we have

$$\text{CLOG}[G_i(u,v)] = \text{CLOG}[F_i(u,v)] + \text{CLOG}[H_d(u,v)]$$ \hspace{1cm} (99)
Averaging over $M$ subimages,

\[
\frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[G_i(u,v)] = \frac{1}{M} \sum_{i=1}^{M} \text{CLOG} [F_i(u,v)] + \frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[H_d(u,v)] \quad (100)
\]

We assumed that $H_d(u,v)$ is included in each subimage. Therefore we can write Eq. (100) as

\[
\text{CLOG}[H_d(u,v)] = \frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[G_i(u,v)] - \frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[F_i(u,v)] \quad (101)
\]

Since the right hand side of Eq. (100) results from an averaging process and the term $\text{CLOG}[H_d(u,v)]$ is included by each subimage, the averaged term $\frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[F_i(u,v)]$ retains little of the flavor of $\text{CLOG}[F_i(u,v)]$. If the term $\frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[F_i(u,v)]$ was available, one could estimate the complex logarithm of the degrading OTF by Eq. (101). However, we must estimate the degrading OTF without a prior knowledge of the original image and the degradation. If the intensity distributions in the $M$ regions of the original image are sufficiently different from one another, and if $M$ is large enough ($M > 100$), then it may be assumed that the term $\frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[F_i(u,v)]$ converges to zero. Therefore, we have

\[
\text{CLOG}[H_d(u,v)] = \frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[G_i(u,v)] \quad (102)
\]

$H_d(u,v)$ is then obtained by taking the antilogarithm i.e.,
\[ H_d(u,v) = \exp \left\{ \frac{1}{M} \sum_{i=1}^{M} \text{CLOG}[G_i(u,v)] \right\} \] (103)

We can estimate the magnitude of the degrading OTF by one of the above methods. We still need to estimate the phase of the degrading OTF. A method for this estimation is given in Ref. 15 and will be summarized here. After segmenting the degraded image and taking the Fourier transform of each segment, we have (Ref. Eq. 93)

\[ \theta_{G_i}(u,v) = \theta_H(u,v) + \theta_{f_i}(u,v) \] (104)

By averaging over \( M \) regions,

\[ \frac{1}{M} \sum_{i=1}^{M} \theta_{G_i}(u,v) = \theta_H(u,v) + \frac{1}{M} \sum_{i=1}^{M} \theta_{f_i}(u,v) \] (105)

Our hope was that the second term in the right-hand side of Eq. (105) would converge to a constant value. However, we cannot make sure that this term converges to anything meaningful. Another method for estimating the phase (Ref. 15) is to form the product of subimages

\[ G_i(u,v) G_i^\ast(u+\Delta u, v+\Delta v) = H_d(u,v) H_d^\ast(u+\Delta u, v+\Delta v) F_i(u,v) \]

and taking average over \( M \) regions, we obtain
\[ \frac{1}{M} \sum_{i=1}^{M} G_i(u,v) G_i^*(u+\Delta u, v+\Delta v) = H_d(u,v)H_d^*(u+\Delta u, v+\Delta v) \]

\[ \frac{1}{M} \sum_{i=1}^{M} F_i(u,v) F_i^*(u+\Delta u, v+\Delta v) \]  

Equation (107)

In Equation (107), we can calculate \( \frac{1}{M} \sum_{i=1}^{M} G_i(u,v) G_i^*(u+\Delta u, v+\Delta v) \) from the degraded image by dividing the degraded image into \( M \) regions (subimages), and Fourier transforming each subimage to form the auto-correlation product \( G_i(u+v) G_i^*(u+\Delta u, v+\Delta v) \) (where \( \Delta u \) and \( \Delta v \) are constants for \( M \) subimages, and then averaging the resulting products by \( M \). If a prototype image which is similar to the original image is available, then we can obtain a rough estimate of \( \frac{1}{M} \sum_{i=1}^{M} F_i(u,v) F_i^*(u+\Delta u, v+\Delta v) \) from the prototype image. An estimation of the product \( H_d(u,v) H_d^*(u+\Delta u, v+\Delta v) \) can then be made by Eq. (107). For a lossless degradation, which means the volume under \( f(x,y) \) is equal to the volume under \( g(x,y) \), we have \( H_d(0,0) = 1 \). Assuming that the products can be estimated and given \( H_d(0,0) = 1 \), we can obtain a recursive relationship where \( H_d^*(u+\Delta u, v+\Delta v) \) is expressed in terms of \( H_d(u,v) \)

\[
H_d^*(u+\Delta u, v+\Delta v) = \frac{\frac{1}{M} \sum_{i=1}^{M} G_i(u,v) G_i^*(u+\Delta u, v+\Delta v)}{H_d(u,v) \frac{1}{M} \sum_{i=1}^{M} F_i(u,v) F_i^*(u+\Delta u, v+\Delta v)}
\]

(108)

However, the stability of this expression is quite poor. For the phase of OTF, we have
\[
\theta_{G_i}(u,v) - \theta_{G_i}(u+\Delta u, v+\Delta v) = \theta_H(u,v) - \theta_H(u+\Delta u, v+\Delta v)
\]

\[
+ \theta_{F_i}(u,v) - \theta_{F_i}(u+\Delta u, v+\Delta v)
\]

(109)

If Eq. (109) is averaged in some sense over \(i\), and if the average value of \([\theta_{F_i}(u,v) - \theta_{F_i}(u+\Delta u, v+\Delta v)]\) can be estimated, then noting that \(\theta_H(0,0) = 0\), we may estimate \(\theta_H(u,v)\) as (Ref. 15).

\[
\theta_H(u,v) = \theta_H(u,v) - [\theta_{G_i}(u,v) - \theta_{G_i}(u+\Delta u, v+\Delta v)] \text{ av}
\]

\[
- [\theta_{F_i}(u,v) - \theta_{F_i}(u+\Delta u, v+\Delta v)] \text{ av}
\]

(110)

**Line Estimation of the OTF.**

Some degrading optical transfer functions such as motion blur in one direction can be described by a single line which may be estimated from the Fourier transform of the degraded image itself. To support this statement, consider the degrading OTF of the motion blur given by Eq. (39).

\[
H_d(u,v) = T \text{Sinc}(fd) e^{-j\pi fd}
\]

where \(f = u \cos\theta + v \sin\theta\). If we assume that the relative motion between camera and object is in the horizontal direction (\(\theta=0\)), the Equation (39) becomes
\[ H_d(u,v) = T \text{Sinc}(ud) e^{-j\pi ud} \quad (111) \]

or for the motion in vertical direction \((\theta = \frac{\pi}{2})\)

\[ H_d(u,v) = T \text{Sinc}(vd) e^{-j\pi vd} \quad (112) \]

Since the right-hand side of the Eq. (111) does not depend on the vertical spatial frequency variable \(v\), we can say that

\[ H_d(u,v) = H_d(u) = T \text{Sinc}(ud) e^{-j\pi ud} \quad (113) \]

Recall that the degraded image can be manipulated by a digital computer in \(N\) by \(N\) matrix form where the rows and columns of this matrix represent the spatial frequency variable \(v\) and \(u\), respectively. In the absence of noise, the Fourier transform of the degraded image is equal to the point-by-point multiplication of the Fourier transform of the original image and the degrading OTF. Therefore, the degrading OTF and the original image matrices has to be identical in size. We can modify, Eq. (113) to write it in matrix form

\[ H_d(k,\ell) = T \text{Sinc}(ud) e^{-j\pi ud} \quad (114) \]

where \(k,\ell\) denotes \(k'th\) column and \(\ell'th\) row of the degrading OTF. And if the degraded image is represented by an \(NxN\) matrix, \(k\) and \(\ell\) take values from 1 to \(N\). If the function given by Eq. (114) is centered at \(N/2\), then we can say

\[ u = k - \frac{N}{2} \quad (115) \]
Equations (113) and (114) show that the degrading OTF, in this particular case, consists of identical rows (lines). Therefore, it will be sufficient to obtain a row of the degrading OTF.

In order to estimate a row of the degrading OTF, we will follow a procedure similar to the segmentation method. By taking the complex logarithm of the Fourier transform of the degraded image, we obtain

\[ \text{CLOG}[G(k, \ell)] = \text{CLOG}[H_d(k, \ell)] + \text{CLOG}[F(k, \ell)] \]  

(116)

where \( k, \ell = 1, 2, 3, \ldots, N \). Averaging over \( N \) rows, we have

\[
\frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[G(k, \ell)] = \frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[H_d(k, \ell)] \\
+ \frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[F(k, \ell)] 
\]  

(117)

We can write from Eq. (117)

\[
\frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[G(k, \ell)] = \text{CLOG}[H_d(k)] + \frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[F(k, \ell)] 
\]  

(118)

In order to obtain an estimate of the first term, \( \text{CLOG}[H_d(k)] \), in the right-hand side of Eq. (118), we need to know the second term, \( \frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[F(k, \ell)] \). Since right hand-side of Eq. (118) results from an averaging process, and the term \( \text{CLOG}[H_d(k)] \) is constant while the term \( \text{CLOG}[F(k, \ell)] \) is changing randomly from pixel to pixel on the same line (\( \ell \)'th row), the resultant average term \( \frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[G(k, \ell)] \) retains

47
little of the flavor of \( \text{CLOG}[F(k, \lambda)] \). If the term \( \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[F(k, \lambda)] \)
can be neglected, then we have
\[
\text{CLOG}[H_d(k)] = \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[G(k, \lambda)]
\]  
(119)

and
\[
H_d(k) = \exp \left( \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[G(k, \lambda)] \right)
\]  
(120)

If the light distribution in the degraded image varies sufficiently
fast, we can expect that the second term in the right-hand side of
Eq. (118) can be approximated by a constant. In this case, we
obtain
\[
H_d(k) = \exp \left( \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[G(k, \lambda)] - c \right)
\]  
(121)

where
\[
C \approx \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[F(k, \lambda)]
\]  
(122)

A prototype image which is similar to degraded image can also be used
for estimating \( \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[F(k, \lambda)] \). Let \( P(k, \lambda) \) be the Fourier
transform of the prototype image, then we may assume that
\[
\frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[F(k, \lambda)] \approx \frac{1}{N} \sum_{\lambda=1}^{N} \text{CLOG}[P(k, \lambda)]
\]  
(123)

Since the degrading OTF ought to be identical in size to the degraded
image, we need to obtain a degrading OTF which is in the form of \( N \times N \) matrix from the estimated row, \( H_d(k) \). This can be done by duplicating
\( H_d(k) \) \( N \) times.
We have discussed another approach called "Line estimation of the OTF" to estimating the degrading OTF of the motion blur in one direction. It seems that it is possible to estimate the OTF of such a degradation from the degraded image itself by this method. The only problem in this approach seems to be the estimation of the term \[ \frac{1}{N} \sum_{\ell=1}^{N} \text{CLOG}[F(k,\ell)]. \]

We consider two cases to overcome this problem. We first assumed that this term would converge to zero or a constant, and secondly we estimated it from a prototype image which is assumed to be similar to that of original image. However, we can say that both cases may contribute an error to estimated degrading OTF due to the estimation of \[ \frac{1}{N} \sum_{i=1}^{N} \text{CLOG}[F(k,\ell)]. \]

In order to avoid the estimation of the term \[ \frac{1}{N} \sum_{i=1}^{N} \text{CLOG}[F(k,\ell)], \] we will employ another approach which is similar to the frequency domain inspection method. Since the first term in the right-hand side of Eq. (118) converges to the complex logarithm of the degrading OTF, it may be possible to recognize the zero pattern of the degrading OTF correctly. The only parameter we need to know then is the total displacement distance \( d \) in Eq. (114). If the first zero of the pattern is "A" pixels (columns) away from the origin, then the total displacement distance \( d \) will be

\[ d = \frac{1}{A} \]  

(124)

After determining the parameter \( d \) of the Eq. (124), we can easily
calculate Eq. (114) for a given exposure time T.

**Logarithmic Estimate.**

In cases where the previous methods are not successful in estimating the degrading OTF, we can obtain an approximation by the following method. In the absence of noise, the complex logarithm of the Fourier transform of the degraded image is given by Eq. (116). For an N\times N degraded image, we divide both sides of Equations (116) by N to obtain

\[
\frac{1}{N} \text{CLOG}[G(k,\lambda)] = \frac{1}{N} \text{CLOG}[H_d(k,\lambda)] + \frac{1}{N} \text{CLOG}[F(k,\lambda)] \quad (125)
\]

Since \(H_d(k,\lambda)\) is spatially invariant, the first term in the right-hand side of Eq. (125) will be the logarithm of the degrading OTF scaled by \(1/N\), while the second term, \(\frac{1}{N} \text{CLOG}[F(k,\lambda)]\), converges to a small number. Therefore, we may say that the right-hand side of Eq. (125) resembles the degrading OTF rather than the original image; i.e., a rough approximation of the degrading OTF is

\[
H_d(k,\lambda) = \frac{1}{N} \text{CLOG}[G(k,\lambda)] \quad (126)
\]

**Estimation of Noise-to-Signal Ratio.**

The least-squares and the power spectrum equalization methods require the noise-to-signal power density ratio in the degraded image as a parameter. We can also say that these methods respond only to the
ratio of the noise-to-signal power density. Since we do not have a prior knowledge about the original image and the noise, we must estimate the NSR (noise-to-signal power ratio) from the degraded image itself.

Most of the researchers assume that the NSR is constant for all spatial frequencies. We may encounter different definitions of constant NSR, such as,

\[ \text{NSR} = \frac{\text{STANDARD DEVIATION OF THE NOISE}}{\text{STANDARD DEVIATION OF THE SIGNAL}} \]  \hspace{1cm} (127)

where the original image is assumed to be corrupted by white Gaussian noise (Ref. 16). Another definition of the NSR is given by (Ref. 17).

\[ \text{NSR} = 2x \frac{\text{STANDARD DEVIATION OF THE NOISE}}{\text{PEAK-TO-PEAK VARIATION OF THE IMAGE}} \]  \hspace{1cm} (128)

or it is defined (Ref. 22)

\[ \text{NSR} = \frac{\text{STANDARD DEVIATION OF THE NOISE}}{\text{PEAK TO PEAK VARIATION OF THE IMAGE}} \]  \hspace{1cm} (129)

It is also possible to define the NSR as the variance of the noise process (Ref. 18), i.e.

\[ \text{NSR} = 2\sigma_n^2 \]  \hspace{1cm} (130)

We can apply one of above equations to estimate the NSR; however, we first need to estimate the variance of the noise process and the original image. Consider a relatively noisy region in the degraded image. This region consists of the blurred image and additive noise.
Since the region is relatively noisy; i.e., it includes more noise components than the blurred image components, the variance of this region may resemble the variance of the noise rather than the variance of the original image. Therefore, it is reasonable to assume that the variance of a relatively noisy region in the degraded image be approximately equal to the variance of the noise process. If we pre-process (high pass filter) the degrading to enhance the high frequency components than the original image does, and if the variance is calculated over several noisy regions in the degraded image and the resulting variances are averaged to obtain the noise variance, then the confidence in above assumption increases. The Laplacian operator may satisfy the objective of above pre-processing. Peak-to-peak variation of the original image can be determined from the degraded image interactively.

In our discussion of the various methods of estimating the NSR from the degraded image, we have assumed that the noise process is wide-sense spatially stationary stochastic process with zero mean and is spectrally white, i.e., \( S_n(u,v) = \text{a constant} \). If \( F(u,v) \) falls off much faster in the spatial frequency domain than \( S_n(u,v) \), correctness of the white noise assumption increases. We will define the NSR as

\[
\text{NSR} = \frac{\text{Average noise power}}{\text{Average signal power}}
\]

(131)

Where average signal power refers to the average power contributed by the original image.
Estimation in the Spatial Domain.

Since the noise and the original image are wide-sense spatially
stationary processes, their power density spectrum is defined by the
Fourier transform of their autocorrelation function (Ref. 9), that
is

\[ S_n(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_n(\Delta x, \Delta y) \{ \exp(-j2\pi(u\Delta x + v\Delta y)) \} \, d\Delta x \, d\Delta y \]  \hspace{1cm} (132)

and

\[ S_f(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_f(\Delta x, \Delta y) \{ \exp(-j2\pi(u\Delta x + v\Delta y)) \} \, d\Delta x \, d\Delta y \]  \hspace{1cm} (133)

It follows that the autocorrelation function is the inverse
Fourier transform of the power density spectrum.

\[ R_n(\Delta x, \Delta y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_n(u,v) \exp(j2\pi(u\Delta x + v\Delta y)) \, du \, dv \]  \hspace{1cm} (134)

and

\[ R_f(\Delta x, \Delta y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_f(u,v) \exp(j2\pi(u\Delta x + v\Delta y)) \, du \, dv \]  \hspace{1cm} (135)

From Equations (134) and (135), we can write

\[ E\{ |h(x,y)|^2 \} = R_n(0,0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_n(u,v) \, du \, dv \]  \hspace{1cm} (136)

and

\[ E\{ |f(x,y)|^2 \} = R_f(0,0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_f(u,v) \, du \, dv \]  \hspace{1cm} (137)
This means that the average noise power and original image power
can be identified by their mean square value. Therefore, we
may write Eq. (131) as

\[
\text{NSR} \quad = \quad \frac{E\{|n(x,y)|^2\}}{E\{|f(x,y)|^2\}}
\]  

(138)

since the mean value of the noise is zero, the numerator of the
Eq. (138) is equal to

\[
E\{|n(x,y)|^2\} \quad = \quad \sigma_n^2
\]  

(139)

where \(\sigma_n^2\), variance of the noise, can be estimated from a relatively
noisy region in the degraded image. If a prototype image which is
similar to that of original image is available, then we may estimate
the denominator term in Eq. (138) from the prototype image \(p(x,y)\).

\[
E\{|f(x,y)|^2\} \quad = \quad E\{|p(x,y)|^2\}
\]  

(140)

However, our primary aim is to estimate all necessary parameters of
restoration filter from the degraded image itself. The degraded
image may be expressed as

\[
g(x,y) \quad = \quad f_b(x,y) + n(x,y)
\]  

(141)

where

\[
f_b(x,y) \quad = \quad h_d(x,y) * f(x,y)
\]  

(142)

If we assume that the average powers contained by \(f_b(x,y)\) and \(f(x,y)\) are
approximately equal, then we may say that
We can now estimate the NSR from the degraded image itself as follows
\[
E\{ |f(x,y)|^2 \} = E\{ |f(x,y)|^2 \}
\]
(143)

where the noise and the original image are assumed to be statistically independent. By using equations (141), (139), (143), and (144) we may obtain,
\[
\text{NSR} = \frac{\sigma_n^2}{\text{E}\{ |g(x,y)|^2 \} - \sigma_n^2}
\]
(145)

where the numerator and denominator denote the noise and original image average power, respectively. If \( \sigma_n^2 \) is small compared to \( \text{E}\{ |g(x,y)|^2 \} \), then Eq. (140) can be approximated as
\[
\text{NSR} = \frac{n^2}{\text{E}\{ |g(x,y)|^2 \}}
\]
(146)

**Estimation in the Frequency Domain.**

It is also possible to estimate NSR in the spatial frequency domain. The power density spectrum of the degraded image is given by
\[
S_g(u,v) = |H_d(u,v)|^2 S_f(u,v) + S_n(u,v)
\]
(147)

where \( S_n(u,v) \) is constant for all spatial frequencies \( u \) and \( v \) and is approximately equal to the variance, i.e.,
By substituting Eq. (148) into Eq. (147) and solving for \( S_n(u,v)/S_f(u,v) \), we obtain

\[
\text{NSR} = \frac{S_n(u,v)}{S_f(u,v)} = \frac{|H_d(u,v)|^2}{S_g(u,v) - \sigma_n^2} \sigma_n^2
\]  

(149)

Since the only constant in above equation is \( \sigma_n^2 \), the NSR will depend on the spatial frequencies \( u \) and \( v \). If the degrading OTF is zero at any spatial frequency, then the NSR becomes zero and leads the restoration filter to undetermined ratio. In order to see this effect, we may insert Eq. (149) into the Eq. (85) which the OTF of the Wiener filter.

\[
H_r(u,v) = \frac{H_d^*(u,v)}{|H_d(u,v)|^2 + \frac{|H_d(u,v)|^2 \sigma_n^2}{S_g(u,v) - \sigma_n^2}}
\]  

(150)

If \( H_d(u,v) = 0 \), then \( H_r(u,v) = 0/0 \) which is undetermined ratio that we encountered in the inverse filtering method. We can avoid this by modifying Eq. (150) so that

\[
H_r(u,v) = \begin{cases} 
\frac{H_d^*(u,v)}{|H_d(u,v)|^2 + \frac{|H_d(u,v)|^2 \sigma_n^2}{S_g(u,v) - \sigma_n^2}}, & \text{if } |H_d(u,v)| > R \\
1, & \text{if } |H_d(u,v)| < R 
\end{cases}
\]  

(151)
where \( R = 0 \). Since we cannot recover the zeros of the \( G(u,v) \), Eq. (161) can be considered as a reasonable approximation to the Wiener filter. In the absence of noise, it performs as an inverse filter. Otherwise, it deemphasizes the noise component in the degraded image.
IV. Results

In order to be able to compare the degrading OTF estimated by one of the methods we discussed with the exact OTF, we may generate a degrading OTF and convolve it with an image to produce a degraded image and then estimate the degrading OTF assuming a little or no prior knowledge about the degradation.

To accomplish the above task, we need to define the spatial domain and the spatial frequency domain. We used 256x256x4-bit images represented by a matrix of 256 rows and 256 columns with each element of this matrix corresponding to a pixel in the picture and each pixel taking on gray-scale values from 0 (black) to 15 (white). In the frequency domain, the discrete Fourier transform of the image is represented by a 256x256 matrix of complex values. Both the spatial (space) domain and the spatial frequency domain representations are centered at the 128'th row and 128'th column of the matrices, i.e., it corresponds to (x=0, y=0) in the spatial domain and (u=0, v=0) in the spatial frequency domain.

A simple computer graphics program was then implemented in FORTRAN 5 to generate an image. This program enables us to draw lines from one point to another and circles centered at a point in the spatial domain. For example, the instructions below will draw a line from (x=-50, y=-50) to (x=50, y=50) with the intensity level 14, and a circle
at the center \((x=0, y=0)\) with radius of 10 and the intensity level 7, respectively,

call LINE( IBUF, 14, -50, -50, 50, 50)

call CIRCLE( IBUF, 7, 0, 0, 10)

where IBUF is an array used to hold the picture file (see Ref. 19). A test image was generated using this program and is shown in Picture 1.

To characterize the degrading system which we described in Chapter II, "Image Restoration System Model", we need to generate a point spread function or an optical transfer function. We implemented Eq. (112) to obtain the OTF of the motion blur in one direction. As we can see from Eq. (112), it only depends on the frequency variable "u" which corresponds to the vertical direction and takes values from -127 to 127 in the frequency domain that we described above. Picture 2 shows the OTF of the degrading system obtained from Eq. (112) for \(T=1.0\), and \(d=0.021\).

The degraded system is then used to obtain a degraded image. This degrading system simply convoloves its point spread function with the input (original image) in the spatial domain. Since the convolution in the spatial domain is equivalent to point-by-point multiplication in the spatial frequency domain, the process of convolving the input (original image) picture with the PSF (point spared function) to produce the output (blurred image) is equivalent to first computing the Fourier transform of the input (original image) then multiply it with the OTF (optical transfer function) point-by-point.
and then computing the inverse Fourier transform of the result to produce the output (blurred) image. This process is summarized by Figure 6 where 2DFFT, and 2DFFT\(^{-1}\) denote forward and inverse two-dimensional Fast Fourier transform, respectively and "X" denotes point-by-point multiplication. Picture 3 shows the output of this degradation process for the input image of Picture 1.

![Diagram](attachment:Figure_5.png)

**Figure 5 The Degrading (Blurring) Process**

We applied all filtering techniques in the Fourier domain. The input and output picture files of the two-dimensional Fast Fourier Transform software that we used were required to be complex picture files, while the input/output picture files of the Octek Image Analyzer system which we used for display of the pictures on a monochrome video monitor were required to be 4-bit integer picture files. Therefore, the process of any of the filtering operations that we will discuss next require the conversion of pictures from integer to complex form and vice versa. A general purpose filtering operation is summarized in Figure 6 where VTC and CTV refer to the video-to-complex and
complex-to-video conversion processes, respectively. The filter parameters are the NSR and the degrading OTF.

![Diagram of filtering operation](#)

**Figure 6 Filtering Operation**

With our implementation on the AFIT Signal Processing Lab's Data General Eclipse S/250 16-bit minicomputer: it takes approximately 36 minutes (except for the estimation of the parameters of the filter) to process a 256x256 input picture by a Wiener filter using above algorithm described in Figure 6. We will now attempt to estimate the degrading OTF from the degraded image by means of the estimation methods described in previous sections.

As we mentioned in the method of frequency domain inspection, we may pre-process the degraded image to reduce the noise effects. However, we can see by looking at Picture 3 that there is no apparent noise present. Therefore, we simply compute the Fourier transform \( G(u,v) \) of the degraded image without pre-processing. Picture 4 shows the plot of \( G(u,v) \). The dark (black) points corresponds to the zeros of the \( G(u,v) \). Before examining the zeros of the \( G(u,v) \), we will turn our attention back to the degraded image (Picture 3). We see that this image is blurred only in vertical direction, i.e., it contains a spread
in one direction only. Based upon this observation, we will assume that
the degradation is due to motion blur in one direction which we described
earlier. If this assumption is true, then we must be able to find a zero
pattern in the Fourier transform of Picture 3 which corresponds to the zeros
of the sinc function given by Eq. (90). Since the blur is in vertical
direction, we should search for the zero pattern in the vertical
direction in the plot of $G(u,v)$ shown by picture 4, and determine the
first zero of the degrading OTF. There are first zeros located 34, 36,
42, 46, 48, and 54 pixels away (see Picture 4) from the origin which obey
the zero pattern given by Eq. (90). However, only one of these zeros
should be the first zero of the degrading OTF given by Eq. (112). In
order to determine the correct first zero, we may pick one of those
zeros, obtain the degrading OTF by Eq. (112), and restore the degraded
image. If the first zero that we chose is not the correct one, the
output of the restoration system (e.g., inverse filter) will not be a
meaningful picture and we should repeat the process with another
choice until we find the zero that produces the "best" restoration.

The parameter $d$ of the degrading OTF given by Eq. (112) is found by

$$d = \frac{1}{|\text{the first zero distance}|} \quad (152)$$

A value of $d = 1/4b$ gives the best restoration of Picture 3 by
inverse filtering. The degrading OTF obtained by Eq. (112) for the
parameter $d = 1/4b$ is shown by Picture 5, and the image restored
by the inverse filter is shown in Picture 6. When we examine the
restored image (Picture 6), we see noise-like amplification artifacts, in addition to restored degradations. These artifacts are due to the ill-conditioned behavior of the inverse filter. We will consider this problem later. When the possible zero patterns become too many, it takes too much time and effort to find the correct zero pattern by this trial-and-error.

Since we concluded from the degraded image (Picture 3) that the degradation is due to the motion in vertical direction, we can employ the line estimation method described in Chapter III, section entitled "Line Estimation of the OTF." If Equation (118) is modified for motion in the vertical direction as

\[
\frac{1}{N} \sum_{k=1}^{N} \text{CLOG}[G(k,\ell)] = \text{CLOG}[H_d(\ell)] + \frac{1}{N} \sum_{k=1}^{N} \text{CLOG}[F(k,\ell)]
\]  

(153)

If the term \( \frac{1}{N} \sum_{k=1}^{N} \text{CLOG}[F(k,\ell)] \) is assumed to converge to zero, then \( H_d(\ell) \) is given by

\[
H_d(\ell) = \exp\left\{ \frac{1}{N} \text{CLOG}[G(k,\ell)] \right\} \quad k = 1, 2, \ldots, N.
\]  

(154)

As we can see from the Eq. (154) \( H_d(\ell) \) is a column obtained by averaging the columns of the matrix which represents the complex logarithm of the \( G(u,v) \), and calculating the antilogarithm of the average. Since the degraded image is 256x256 in size, \( N \) is, in this case, equal to 256 and \( H_d(\ell) \) is an array (column) with 256 elements. To estimate the first element of the \( H_d(\ell) \) corresponding elements of the 256 columns of this matrix should be summed and averaged and then
antilogarithm of this average should be calculated. This method was implemented to estimate $H_d(\ell)$ from the complex logarithm of $G(u,v)$. $H_d(\ell)$ is then duplicated $N$ times to obtain the $N \times N$ matrix of the degrading OTF. Picture 7 shows the estimated degrading OTF. It seems like a sinc function, but the intensity variations do not obey a sinc function variations due to the term, $\frac{1}{N} \sum_{k=1}^{N} \text{CLOG}[F(k,\ell)]$, which we assumed to be zero. However, we may obtain an exact sinc function which has its first zero identical to that of Picture 7. This exact sinc function is shown by picture 8 and it is assumed to be the degrading OTF. The degraded image is restored by inverse filter and it is shown by Picture 9. We can see the ill-conditioned behavior of exact inverse filter in the Picture 9. We then modified the inverse filter as follows

$$
H_r(u,v) = A(u,v) = \begin{cases} 
\frac{1}{H_d(u,v)}, & \text{if } H_d(u,v) \geq R \\
1, & \text{if } H_d(u,v) < R 
\end{cases}
$$

(155)

Restored pictures using Eq. (155) for different $R$ values are listed below:

<table>
<thead>
<tr>
<th>Picture</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picture 10</td>
<td>$0.000999$</td>
</tr>
<tr>
<td>Picture 11</td>
<td>$0.009999$</td>
</tr>
<tr>
<td>Picture 12</td>
<td>$0.099999$</td>
</tr>
<tr>
<td>Picture 13</td>
<td>$0.02$</td>
</tr>
</tbody>
</table>

Table 1 Restored Pictures and $R$ Values of Eq. (155)
As we can see from the these restored pictures, it is possible to overcome the ill-conditioned behavior of the inverse filter. For small values of \( R \), the ill-conditioned behavior still exists, while the degradation is no longer restored for the large values of \( R \). We could, therefore, find an "optimum" value of \( R \) interactively.

The degraded image (Picture 3) that we restored in the preceding experiment was a binary (two-intensity level) picture generated by computer. We next applied the methods developed in Chapter III to the degraded image shown in Picture 14. This picture was digitized from photographic film taken a hand held 35 mm camera. The picture is a blurred image of the masking of an aircraft fuselage. It includes more than two intensity levels with maximum and minimum intensity levels, 14, 0, respectively. Its mean, meansquare, and variance are calculated to be 5.11, 34.43, 9.84, respectively. Since no detailed information was available on how the image was acquired, we assume no prior information about the degradation. Therefore, we must estimate all essential parameters from the degraded image itself. It is not possible to determine or assume the form of the degradation by looking at the degraded image (Picture 14). But, we can see that it contains noise. Therefore, we will first apply the Laplacian operator to the degraded image. Picture 15 shows the degraded image after the Laplacian operation. We next take the Fourier transform of Picture 15. The magnitude of the Fourier transform presented in Figure 7 shows circles of zeros of \(|G(u,v)|\) centered about the origin. The first and second zero
are located \( R_1 = 3 \) and \( R_2 = 6 \) pixels away from the origin, respectively. The ratio of \( R_1 \) to \( R_2 \) becomes

\[
\frac{R_1}{R_2} = \frac{3}{6} = 0.5
\] (156)

The degrading OTF of out-of-focus camera with circular aperture forms approximately periodic circular zero pattern and it is given by Eq. (91). The ratio of the first zero of this pattern to its second zero is equal to

\[
\frac{\text{THE FIRST ZERO}}{\text{THE SECOND ZERO}} = \frac{0.61}{0.117} = 0.546106
\] (157)

which is approximately equal to the ratio \( R_1/R_2 \). Therefore, we will assume that the degradation is due to out-of-focus camera with circular aperture and its degraded OTF is of the form of Eq. (46). The parameter \( r \) of Eq. (46) is obtained from the relationship

\[
\frac{0.61}{r} = 3
\] (158)

and then

\[
r = \frac{0.61}{3} = 0.2033
\] (159)

Implementation of Eq. (46) requires us to calculate the Bessel function of first kind of order \( n \). These series expansion of \( J_n(x) \) is given by (Ref. 20)
<table>
<thead>
<tr>
<th></th>
<th>123</th>
<th>124</th>
<th>125</th>
<th>126</th>
<th>127</th>
<th>128</th>
<th>129</th>
<th>130</th>
<th>131</th>
<th>132</th>
<th>133</th>
<th>134</th>
<th>135</th>
<th>136</th>
<th>137</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7 The Zeros of $|G(u,v)|$
\[ J_n(x) = \sum_{k=0}^{N} \frac{(-1)^k (x/2)^{n+2k}}{k! \Gamma(n+k+1)} \]  

where \( x = 2\pi r f \), and \( f = \sqrt{u^2 + v^2} \). The spatial frequency domain that we defined at the beginning of this chapter is normalized by 127 so that the spatial frequency variable can take values from -1.0 to 1.0. This enables the Bessel function to converge to zero. Eq. (46) is then calculated for \( N=50 \), \( n=1 \), and \( r = 0.2033 \) to obtain the degrading OTF shown by Picture 16.

After estimating the degrading OTF, we next need to estimate the NSR in the degraded image. If the NSR is defined by Eq. (146), i.e.,

\[ \text{NSR} = \frac{\sigma_n^2}{E(|g(x,y)|^2)} \]

where \( E(|g(x,y)|^2) \) is the mean square value of the degraded image (Picture 14) and is computed to be equal to 34.43, and the variance of the noise is calculated from a relatively noisy region shown in Picture 17 to be 5.91. Then, the NSR becomes 0.172. Picture 18 shows the image restored by the Wiener filter with degrading OTF of Picture 16 and the NSR of 0.172. Another way to obtain the degrading OTF is to obtain the degrading point spread function. The Fourier transform of the degrading point spread function gives us the degrading OTF. The degrading point spread function given by Eq. (45) is calculated for \( R = 0.203 \) (shown in Picture 19), then Fourier
transformed to obtain the degrading OTF shown in Picture 20.

Picture 21 shows the restored image obtained by Wiener filter with this degrading OTF (Picture 20) and the same NSR (0.172).

We also tested another approach to estimating the degrading OTF presented in Eq. (126). Since the degraded image is 256x256, the parameter N of Eq. (126) is equal to 256. With the degrading OTF estimated by Eq. (126) and the NSR of 0.172, the degraded image is restored by Wiener filter. Picture 22 shows the restored image. As we can see, it is possible to restore some degradation, but most of the degradation is unrestored. We estimated the NSR by Eq. (149) and used the same OTF to obtain restoration filter OTF given by Wiener filter. Restored images with different NSR's are listed below:

| Picture 23 | $S_n(u,v) = 5.91$ |
| Picture 24 | $S_n(u,v) = 2.4$ |
| Picture 25 | $S_n(u,v) = 1.55$ |

Table 2 Restored Pictures and Noise Variances

Where $S_n(u,v)$ is the noise power required by Eq. (149) to estimate the NSR. We assumed that the noise is Gaussian random process with zero mean and the variance obtained from the noisy region (see Picture 17) equal to 5.91. The autocorrelation of the noise $R_n$ in the degraded image is estimated by

$$R_n = 0.164 \exp \left\{ -\frac{1}{11.8} \left( x^2 + y^2 \right) \right\}$$

(161)
noise power is obtained by

$$S_n(u,v) = | \mathcal{F}(R_n)|$$  \hspace{1cm} (162)

The NSR is then estimated by Eq. (149). The restored image is shown in Picture 26. In order to avoid unwanted artifacts in the restored images with the NSR obtained from Eq. (149), we used modified Wiener filter OTF given by Eq. (150) and obtained restored images using different values of $R$ values with the same NSR value (see Pictures 27, 28). The NSR was estimated by Eq. (149) for $S_n(u,v) = \sigma_n^2 = 5.91$.

Another approach to estimation of the degrading OTF is the line estimation method described in the previous chapter. Although it was developed for linear motion blur in one direction, we will use it for estimating the degrading OTF of the degraded picture. This degrading OTF consists of only a line either in vertical direction or horizontal direction. A line in vertical direction can be estimated from the degraded image itself by Eq. (120). In this case, we have

$$H_d(k) = \exp\left\{ \frac{1}{256} \sum_{k=1}^{256} \text{CLOG}[G(k,Z)] \right\}$$  \hspace{1cm} (163)

where $k = 1, 2, 3, \ldots, 256$. The degrading OTF obtained by Eq. (120) is shown in Picture 29. The degraded image is restored by Wiener filter with the NSR of 0.172, and shown in Picture 30. Secondly, A line in horizontal direction is estimated by

$$H_d(\ell) = \exp\left\{ \frac{1}{256} \sum_{k=1}^{256} \text{CLOG}[G(k,Z)] \right\}$$  \hspace{1cm} (164)
and is shown in Picture 31. The degraded image is restored by Wiener filter with the same NSR(0.172) and it is shown in Picture 32. As we can see, it is possible to restore most of the degradation in the degraded image. Therefore, we generated a sinc function (Eq. (112)) with a first zero \((d=1/32, T=1.0)\) is taken to the equal to that of Picture 31. Pictures 33 and 34 show the sinc function and the restored image by Wiener filter with the same NSR(0.172), respectively. Some degradation still is present while most of the high frequencies are emphasized in the restored image (Picture 34).

We finally tried to estimate the degrading OTF by segmentation method which we described in Chapter III. The degraded image is divided into 169 subimages which are all 64x64 in size. This is done by forming a window which is 64x64 in size, then its left-hand corner placed at the left hand-corner (first row and first column) of the degraded image, and subimages are obtained shifting this window by a shifting factor 16 pixels in this particular case toward right and down on the degraded image. Each subimage is Fourier transformed to form the Eq. (93). Eq. (96) is calculated by multiplying the magnitude of the Fourier transform of each subimage point-by-point, then taking \(1/M\) power of each resultant point and dividing by the term \(\left[ \pi F_i(u,v) \right]^{1/M}\) which is assumed to be constant and equal to 1.0. We were not able to restore the degraded image by using the resulting OTF. The same algorithm was implemented for different constant values, but, it was still not possible to obtain a meaningful degrading OTF.
We next estimated the power spectra given by Eq. (97) by averaging the magnitude square of the Fourier transform of each subimage over 169 subimages. Then the power cepstrum of this estimate is obtained. The spikes occur periodically as in motion blur and they are along the horizontal direction (the degraded image (Picture 14) seems to be blurred in both horizontal and vertical directions through). The first spike is 4 pixels away from the origin. We assumed the degradation was due to motion in horizontal direction and calculated the degrading OTF given by Eq. (111) for $d=0.25$ it was not possible to obtain a meaningful restoration.

Another attempt to estimate the degrading OTF was made by using Eq. (103). The degraded image was divided into 49 subimages by shifting the $64\times64$ window for 32 pixels in horizontal and vertical directions. Each subimage was Fourier transformed and the complex logarithm of the Fourier transform of each subimage taken to form Eq. (99). The degrading OTF was then estimated by taking the average over 49 subimages and taking anti-logarithm to calculate Eq. (103). The resultant degrading OTF seemed like a sinc function with its minimum at 18 pixel away from the center. We again generated a sinc function by Eq. (112) and used this as the degrading OTF (shown in Picture 35), and with the same NSR(0.172). The restored image is shown in Picture 36.

To be able to test the methods that we described in Chapter III for the estimation of the NSR, we used a Vidicon camera and the blurred image shown by Picture 1. The vidicon camera was oriented to a flat surface and
a picture was acquired. The light distribution on the flat surface was made as uniform as possible. Our aim was to obtain a noise source which might reflect the characteristics of the noise process that we discussed in Chapter II section entitled "Noise Process". The noise picture was then processed (Laplacian operator applied) to emphasize its high frequency components and subtracted the mean. The processed noise image is shown in Picture 37. Then the blurred image (Picture 3) and the noise image was added together to obtain the noisy, degraded image shown in Picture 38. Then we used the estimation methods that we describe in the spatial domain and the spatial frequency domain. A noisy region is selected as in labeled in picture. The variance over this region was calculated as 14.7376 and the mean square calculated over the picture as 65.8043. Then the degraded image (picture 38) was restored by Wiener filter using the degraded OTF shown in Picture 5, and the NSR estimated by Eq. (103) as 0.2. The restored picture is shown in Picture 40. As we can see most of the noise in the degraded image (Picture 38) is reduced in the restored image (Picture 40) while some blur is left unrestored. This is because we assumed that the power spectrum density is white, i.e., constant and equal to the variance of the noise to derive a method to estimate the NSR in Chapter III. This is not true in reality in this case and the restored image is not optimal. Picture 41 shows the image restored by power reduction filter using the same degrading OTF and the reduced noise in the restored image. There is no effect in the restoration.
Two main approaches to estimating the degrading OTF have been presented in this thesis. First, we determined the form of the degrading OTF and whether it is due to linear camera motion or out-of-focus camera, then determined the necessary parameters to estimate the degrading OTF. Secondly, the degrading OTF was estimated directly from the degraded image. Methods to estimate the NSR have been defined in spatial domain and in spatial frequency domain.

A trivial solution to the first approach was the method of frequency domain inspection. However, this method required that the degrading OTF must be oscillatory and the image must contain appreciable high frequency components, well distributed, so that the zeros in the degrading OTF can be identified. Furthermore, the NSR must be small enough not to obscure the zeros; otherwise, it may become impossible to determine its parameter (i.e., to recognize the zero pattern) unless a prior knowledge about the original image is available. The degraded image shown in Picture 3 satisfied these requirements. There were six possibilities for the first zero of the degrading OTF that we were searching for. We had to try each one of them and find out the one which restores the degraded image best. The line estimation method was then implemented to estimate the degrading OTF. We could not estimate the exact degrading OTF due to the term contributed by the original image. However, we were able to determine the first zero of the degrading OTF. This method resulted in a considerable saving in time and effort over that required by the frequency domain inspection method. It takes six months for the time of the entire.
which implements line estimation method while it takes more than an hour for the frequency domain inspection to analyze and determine the parameters of the degrading OFF. The degraded image (Picture 3) was then restored by an inverse filter. We modified the inverse filter to avoid the ill-conditioned behavior. It was possible to restore most of the information which was lost due to the blur. If we compare the degraded image Picture 3 with the original image (Picture 1), most of the information in the circles and the upper parallel lines of the squares are lost. One can no longer say that they are parallel lines. In the restored pictures, these lines and circles, including tiny details, are recovered. This includes the highest possible frequency data, the upper parallel lines of the inside square which are one pixel away from each other. However, the restored images do not look as pleasant and clear as the original image does. They include noise like variations. We had to compromise at this point, since we could not implement the exact inverse filter. On the other hand, it may be considered more important to recover the information which is lost than to obtain a more pleasant picture. Once the lost information is recovered, it then can be processed with the constraints of the human eye to improve the quality of the restored image.
hand side of the Eq. (160) is theoretically defined for infinite number of terms which we calculated it for 50 terms. But this did not contribute a significant error to the restored pictures. The same degrading OTF is calculated by the aid of the Fourier transform. The degrading point spread function was calculated and then Fourier transformed to obtain the degrading OTF. Some improvement was obtained in the restoration. The segmentation methods which are successfully used to restore the degraded images by related references were applied to estimate the degrading OTF from the degraded image. A limited improvement in the restorations was obtained due to the nature of the degraded image. It was possible to restore some of the degradation by the logarithmic estimate method. This method performed poorly, since it was a rough approximation the degrading OTF.

The estimation of the NSR in spatial domain performed well while the estimation in frequency domain introduced an ill-conditioned behavior. The ill-conditioned behavior of the frequency domain estimation can be avoided by modifying the restoration filter OTF like we did. Except the ill-conditioned behavior, the frequency domain estimation of the NSR performed well.
V. Conclusions and Recommendations

Conclusions

The method of spatial frequency domain inspection produces good results, if the degraded image has high frequency components, well distributed in the spatial frequency domain and a relatively small NSR so that the zeros of the degrading transfer function can be identified. The degraded image in Picture 3 met these requirements and the frequency domain method was successful. However, one must analyze the zeros and consider many possible patterns for the zero pattern of the degrading OTF. This can be done for the pictures 64x64 in size, if the degraded picture is 256x256, 512x512, or 1024x1024, recognition of the possible zero patterns may become a very difficult and time consuming task.

Image segmentation methods required that we segment the degraded image into 169 subimages. It takes about 3 hours of run-time for the segmentation method to estimate the degrading OTF using the Signal Processing Lab's Eclipse S/250 computer and the program SECEST.FR given in Appendix B. We were not able to obtain a good estimate of the degrading OTF by the segmentation methods.

Line estimation method takes less run-time (approximately 10 minutes) and effort to estimate the degrading OTF. It was developed for the degradation due to uniform camera motion in a sphere.
may be applied to other pictures with unknown degradations. If an improvement is obtained in the restored picture, then further steps can be taken, such as computing an exact sinc function, or Bessel function with the zeros of the estimated line. This method performed well in the restoration of the degraded images that we used in this thesis.

The logarithmic estimate method performed poorly, since it was a rough approximate to the degrading OTF.

The estimation method of the NSR in the spatial domain performed well while the estimation in the spatial frequency introduced an ill-conditioned behavior.

As a final statement of this discussion we can say that there is a chance to restore images degraded by linear camera motion or out-of-focus lens system with circular aperture with no information about the degradation or the noise. The success in restoring the degraded images depends upon a good estimate of the degrading OTF and the NSR. The success in estimating these two depends upon a good modeling. The results indicated that the logarithmic estimate could be used in the case that other methods are not successful, since it makes a rough approximation to the degrading OTF. The line estimate method for determining the degrading OTF of uniform linear camera motion and the spatial domain and frequency domain estimation methods of the NSR, if the restrictions are sufficient to estimate the degradation OTF and the NSR properly.
Recommendations

The depth of this research was necessarily limited. Only limited analysis and testing could be accomplished due to time constraints. Therefore, recommendations for follow-on study consists of further analysis and testing of the estimation methods.

A further study may be made on estimating the magnitude and the phase of the original image more carefully to obtain a better estimate of the degrading OTF.

In the stages of estimating the degrading OTF by segmentation method or frequency domain inspection method, the form of the degradation and necessary parameters to estimate the degrading OTF has to be obtained interactively. A study can be made to reduce the human interaction.

The case that the degradation due to both linear camera motion and out-of-focus camera was not considered. The methods described here can be tested for this type of degradation.

If noise is not present it can be neglected, and in case the degradation is the Fourier transform of the degraded image, the

would then be estimated by the method of cross correlation.

The results of a study of this nature could be used as a basis for a more complete study of degradation in these cases.
(Ref. 23, 24). Therefore, investigation and the implementation of them may be considered as a further study.
## Appendix A

<table>
<thead>
<tr>
<th>Pictures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picture 1</td>
<td>A-3</td>
</tr>
<tr>
<td>Picture 2</td>
<td>A-4</td>
</tr>
<tr>
<td>Picture 3</td>
<td>A-5</td>
</tr>
<tr>
<td>Picture 4</td>
<td>A-6</td>
</tr>
<tr>
<td>Picture 5</td>
<td>A-7</td>
</tr>
<tr>
<td>Picture 6</td>
<td>A-8</td>
</tr>
<tr>
<td>Picture 7</td>
<td>A-9</td>
</tr>
<tr>
<td>Picture 8</td>
<td>A-10</td>
</tr>
<tr>
<td>Picture 9</td>
<td>A-11</td>
</tr>
<tr>
<td>Picture 10</td>
<td>A-12</td>
</tr>
<tr>
<td>Picture 11</td>
<td>A-13</td>
</tr>
<tr>
<td>Picture 12</td>
<td>A-14</td>
</tr>
<tr>
<td>Picture 13</td>
<td>A-15</td>
</tr>
<tr>
<td>Picture 14</td>
<td>A-16</td>
</tr>
<tr>
<td>Picture 15</td>
<td>A-17</td>
</tr>
<tr>
<td>Picture 16</td>
<td>A-18</td>
</tr>
<tr>
<td>Picture 17</td>
<td>A-19</td>
</tr>
<tr>
<td>Picture 18</td>
<td>A-20</td>
</tr>
<tr>
<td>Picture 19</td>
<td>A-21</td>
</tr>
<tr>
<td>Picture 20</td>
<td>A-22</td>
</tr>
<tr>
<td>Picture 21</td>
<td>A-23</td>
</tr>
</tbody>
</table>
Picture 22 ........................................ A-24
Picture 23 ........................................ A-25
Picture 24 ........................................ A-26
Picture 25 ........................................ A-27
Picture 26 ........................................ A-28
Picture 27 ........................................ A-29
Picture 28 ........................................ A-30
Picture 29 ........................................ A-31
Picture 30 ........................................ A-32
Picture 31 ........................................ A-33
Picture 32 ........................................ A-34
Picture 33 ........................................ A-35
Picture 34 ........................................ A-36
Picture 35 ........................................ A-37
Picture 36 ........................................ A-38
Picture 37 ........................................ A-39
Picture 38 ........................................ A-40
Picture 39 ........................................ A-41
Picture 40 ........................................ A-42
Picture 41 ........................................ A-43
Picture 23 Output of Wiener Filter OTF obtained by Eq. (126) and NSR obtained by Eq. (149) for $S_n(u,v) = 5.91$
Picture 24  Output of Wiener filter. OTF obtained by Eq. (126) and NSR by Eq. (149) for $S_n(u,v) = 2.4$
Picture 25  Output of Wiener filter. OTF obtained by Eq. (126) and NSR by Eq. (149) for $S_n(u,v)$
Picture 28 Obtained by Modified Wiener Filter Eq. (151) for Different R Value
### Appendix B

#### Software Listings

<table>
<thead>
<tr>
<th>Software Listings</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVFLT.FR</td>
<td>B-2</td>
</tr>
<tr>
<td>WIENER.FR</td>
<td>B-4</td>
</tr>
<tr>
<td>PEQFLT.FR</td>
<td>B-6</td>
</tr>
<tr>
<td>CONST.FR</td>
<td>B-8</td>
</tr>
<tr>
<td>SEGEST.FR</td>
<td>B-10</td>
</tr>
<tr>
<td>SEGEST2.FR</td>
<td>B-13</td>
</tr>
<tr>
<td>SEGEST3.FR</td>
<td>B-16</td>
</tr>
<tr>
<td>MTRF.FR</td>
<td>B-19</td>
</tr>
<tr>
<td>DPSF.FR</td>
<td>B-21</td>
</tr>
<tr>
<td>BESSEL.FR</td>
<td>B-23</td>
</tr>
<tr>
<td>LEST.FR</td>
<td>B-25</td>
</tr>
<tr>
<td>LOGTRF.FR</td>
<td>B-27</td>
</tr>
<tr>
<td>CEPST.FR</td>
<td>B-29</td>
</tr>
<tr>
<td>LAP.FR</td>
<td>B-31</td>
</tr>
<tr>
<td>ADV.FR</td>
<td>B-33</td>
</tr>
<tr>
<td>ADC.FR</td>
<td>B-35</td>
</tr>
<tr>
<td>STAT.FR</td>
<td>B-37</td>
</tr>
<tr>
<td>IMAGE.FR</td>
<td>B-40</td>
</tr>
<tr>
<td>CIRCLE.FR</td>
<td>B-42</td>
</tr>
<tr>
<td>LINE.FR</td>
<td>B-43</td>
</tr>
<tr>
<td>SUBIMG.FR</td>
<td>B-45</td>
</tr>
<tr>
<td>SVTC.FR</td>
<td>B-46</td>
</tr>
<tr>
<td>SFFT2.FR</td>
<td>B-47</td>
</tr>
<tr>
<td>UNP.FR</td>
<td>B-47</td>
</tr>
</tbody>
</table>
This program implements the inverse filter. Inverse filter is modified to avoid the ill-conditioned behaviour due to the zeros of the degrading OTF.

The input files will be 2DFFT of the degraded image and the degrading OTF. The minimum value of the degrading OTF to avoid the zeros will also be asked. The output file will be 2DFFT of the restored image.

RELOAD LINE.
RLDR INVFLT @FLIB@

COMMAND LINE
INVFLT

*****************************************************************************************************************************************

COMPLEX IN1(1024), IN2(1024), OUT(1024)
INTEGER INFLN1(7), INFLN2(7), OFLN1(7)
REAL R1, R2

C ACCEPT INPUT

ACCEPT "ENTER BLURRED IMAGE FILE NAME ---:"
READ(11, 1) INFLN1(1)
ACCEPT "ENTER TRANSFER FUNCTION FILE NAME---:"
READ(11, 1) INFLN2(1)
1 FORMAT(S13)
ACCEPT "ENTER OUTPUT FILENAME ---:"
READ(11, 1) OFLN1(1)

TYPE " 
TYPE "ENTER MINIMUM VALUE TO AVOID ZEROS OF THE OTF "
ACCEPT " 
CALL OPEN(1, INFLN1, 1, IER)
CALL CHECK(IER)
CALL OPEN(2, INFLN2, 1, IER)
CALL CHECK(IER)

CALL DFILW (OFLN1, IER)
CALL CFILW (OFLN1, 2, KER)
CALL CHECK(KER)
OPEN 3, OFLN1, ATT="OR", ERR=10

CONTINUE

C TYPE " 
TYPE "ENTER SIZE OF FILES "
ACCEPT "256, 128, 64, OR 32 ---: " , ISIZE
C
C TEST FOR ONLY THE ALLOWABLE SIZES
C
IF(ISIZE.EQ.256) GO TO 6
IF(ISIZE.EQ.128) GO TO 6
IF(ISIZE.EQ.64) GO TO 6
IF(ISIZE.EQ.32) GO TO 6
GO TO 5
6 CONTINUE ; size is okay

IFACTOR=256/ISIZE ; compute scaling factor
ITIMES=64/IFACTOR**2-1
A=0
DO 3 I=0,ITIMES
   CALL RDBLK(1,16*I,IN1,16,IER)
   CALL CHECK(IER)
   CALL RDBLK(2,16*I,IN2,16,IER)
   CALL CHECK(IER)
   DO 2 J=1,1024
      R1=REAL(IN2(J)**2 + AIMAG(IN2(J))**2
      R1=R1**0.5
      IF(R1.LE.R2) OUT(J)=IN1(J)
      IF (R1.GT.R2 ) OUT(J)=IN1(J)/IN2(J)
2 CONTINUE
   CALL WRBLK(3,16*I,OUT,16,IER)
   CALL CHECK(IER)
3 CONTINUE
CALL RESET
STOP"<7><7><7>INVFLT FR"
END
This program implements the Wiener filter in the Fourier domain. The inputs are the degrading OTF and the Fourier transform of the degraded image, and the NSR. The output will be the Fourier transform of the estimated (restored) image.

**RELOAD LINE:**

```
RLDR WIENER @FLIBC
```

**COMMAND LINE:**

```
WIENER
```

Program will ask for the degrading OTF, degraded image, and the output file names and the NSR, and the size of the files.

---

```c
COMPLEX IN1(1024), IN2(1024), OUT(1024)
COMPLEX H1, H2, WIENER, CSN

INTEGER INFLNMI(7), INFLNM2(7), OFLNM1(7)
```

---

```c
C ACCEPT INPUT

ACCEPT "ENTER BLURRED IMAGE FILE NAME -->"" READ(11,1)INFLNMI(1)
ACCEPT "ENTER TRANSFER FUNCTION FILE NAME -->" READ(11,1)INFLNM2(1)
1 FORMAT(5F13)
ACCEPT "ENTER OUTPUT FILENAME -->" READ(11,1)OFLNM1(1)
TYPE " " ACCEPT "ENTER NOISE TO SIGNAL RATIO = ", SN CALL OPEN(1,INFLNMI,1,IER)
CALL CHECK(IER)
CALL OPEN(2,INFLNM2,1,IER)
CALL CHECK(IER)
CALL DFLIW (OFLNM1, IER)
CALL CFILW (OFLNM1, 2, KFR)
CALL CHECK(KER)
OPEN 3, OFLNM1, ATT="OR", ERR=100
100 CONTINUE
5 TYPE " " TYPE " ENTER SIZE OF FILES " ACCEPT 256, 128, 64, OR 32 -->". ISIZE
```
TEST FOR ONLY THE ALLOWABLE SIZES

IF(ISIZE.EQ.256) GO TO 6
IF(ISIZE.EQ.128) GO TO 6
IF(ISIZE.EQ.64) GO TO 6
IF(ISIZE.EQ.32) GO TO 6
GO TO 5
END

CONTINUE ; size is okay

IFACTOR=256/ISIZE ; compute scaling factor
ITIMES=64/IFACTOR**2-1

DO 3 I=0, ITIMES

CALL RDBLK(1, 16*I, IN1, 16, IER)
CALL CHECK(IER)
CALL RDBLK(2, 16*I, IN2, 16, IER)
CALL CHECK(IER)

DO 2 J=1, 1024
   H1=CONJG(IN2(J))
   H2= IN2(J)*H1
   CSN=CMPLX(SN, 0.0)
   WIENER= H1/( H2 + CSN ) ; Wiener Filter OTF
   OUT(J)=IN1(J)*WIENER ; 2dfft of restored image.
2 CONTINUE

CALL WRBLK(3, 16*I, OUT, 16, IER)
CALL CHECK(IER)

3 CONTINUE

CALL RESET
STOP "<7><7><7>WIENER"
END
This program implements the power spectrum equalization filter. Input files should be the degrading OTF and the Fourier transform of the degraded image. Output file will be the estimated (restored) image.

RELOAD LINE:
RLDR PEGFLT @FLIBd

COMMAND LINE:
PEGFLT
Program will ask for input and output file name, and the NSR.

COMPLEX IN1(1024), IN2(1024), OUT(1024)
COMPLEX H1, H2, PEGFLT, CSN

INTEGER INFLNM1(7), INFLNM2(7), OFLNM1(7)

ACCEPT INPUT

ACCEPT "ENTER DEGRADED IMAGE FILE NAME -->" READ(11,1)INFLNM1(1)
ACCEPT "ENTER OTF FILE NAME -->" READ(11,1)INFLNM2(1)
1 FORMAT(S13)
ACCEPT "ENTER OUTPUT FILE NAME -->" READ(11,1)OFLNM1(1)

TYPE""
ACCEPT "ENTER NOISE TO SIGNAL RATIO = ", SN
CALL OPEN (1, INFLNM1, 1, IER)
CALL CHECK (IER)
CALL OPEN (2, INFLNM2, 1, IER)
CALL CHECK (IER)

CALL DFILW (OFLNM1, IER)
CALL CFILW (OFLNM1, 2, KER)
CALL CHECK (KER)
OPEN 3, OFLNM1, ATT="OR", ERR=100

100 CONTINUE
5 TYPE""

TYPE "ENTER SIZE OF FILES "
ACCEPT 256, 128, 64, OR 32 -->", ISIZE
C TEST FOR ONLY THE ALLOWABLE SIZES

IF(ISIZE. EQ. 256) GO TO 6
IF(ISIZE EQ 128) GO TO 6
IF(ISIZE EQ 64) GO TO 6
IF(ISIZE EQ. 32) GO TO 6
GO TO 5
6 CONTINUE ; size is okay

IFACTOR=256/ISIZE ; compute scaling factor
ITIMES=64/IFACTOR**2-1

DO 3 I=0, ITIMES
    CALL RDBLK(1, 16*I, IN1, 16, IER)
    CALL CHECK(IER)
    CALL RDBLK(2, 16*I, IN2, 16, IER)
    CALL CHECK(IER)
    DO 2 J=1, 1024
        H1=CONJG(IN2(J))
        H2=IN2(J)*H1
        CSN=CMPLX(SN, 0.0)
        PEQFLT=((1.0/(H2+CSN)))*0.5
        OUT(J)=IN1(J)*PEQFLT
    2 CONTINUE
    CALL WRBLK(3, 16*I, OUT, 16, IER)
    CALL CHECK(IER)
3 CONTINUE

CALL RESET
STOP"<7><7><7>:PEQFLT.FR"
END
The NSR is estimated in the spatial frequency domain. Wiener filter is then used to restore the degraded image. Inputs are 2DFFT of the degraded image, and the noise power. Output will be the restored image.

**RELOAD LINE:**
RLDR CONST @FLIB@

**COMMAND LINE:**
CONST
Program will ask for input and output file name, create the output file and then ask for file size.

```c
COMPLEX G(1024), F(1024), NW, HBPS, NSR, DENUM, GPS, MPS, SIZE
INTEGER INFLNM1(7), OFLNM1(7), INFLNM2(7)
COMPLEX HB(1024)
REAL NLEV, R, M

C ACCEPT INPUT

ACCEPT "ENTER DEGRADED IMAGE FILE NAME -->"
READ(11, 1) INFLNM1(1)
ACCEPT "ENTER TRANSFER FUNCTION FILE NAME -->"
READ(11, 1) INFLNM2(1)

1 FORMAT(S13)
ACCEPT "ENTER OUTPUT FILENAME -->"
READ(11, 1) OFLNM1(1)

CALL OPEN(1, INFLNM1, 1, IER)
CALL CHECK (IER)

CALL OPEN(2, INFLNM2, 1, IER)
CALL CHECK (IER)

CALL DFILW (OFLNM1, IER)
CALL CFILW (OFLNM1, 2, KER)
CALL CHECK (KER)
OPEN 3, OFLNM1, ATT="OR", ERR=100

100 CONTINUE

5 TYPE "C
TYPE "ENTER SIZE OF FILES: 
ACCEPT " 256, 128, 64, OR 32 --> ", ISIZE
TYPE "C
ACCEPT " ENTER MINIMUM OF DEG. OTF ....> ", R
ACCEPT " ENTER NOISE POWER ....> ", NLEV
```
TEST FOR ONLY THE ALLOWABLE SIZES

IF(ISIZE.EQ.256) GO TO 6
IF(ISIZE.EQ.128) GO TO 6
IF(ISIZE.EQ.64) GO TO 6
IF(ISIZE.EQ.32) GO TO 6
GO TO 5
CONTINUE ; size is okay

IFACTOR=256/ISIZE ; compute scaling factor
ITIMES=64/IFACTOR**2-1

SIZE=CMPLX(ISIZE**2, 0.0)
NPS = CMPLX(NLEV,0.0)

DO 3 I=0, ITIMES
    CALL RDBLK(1, 16*I, G, 16, IER)
    CALL CHECK(IER)
    CALL RDBLK(2, 16*I, HB, 16, IER)
    CALL CHECK(IER)

DO 2 J=1, 1024
    HBP$ = HB(J)*CONJG(HB(J)) * NPS
    GPS = G(J)*CONJG(G(J))
    DENUM = GPS - NPS
    NSR = HBP$/DENUM

CALCULATE RESTORATION FILTER TRF
M = HB(J)*CONJG(HB(J))
M = SQRT(M)
HW = CONJG(HB(J))/ ( CONJG(HB(J))*HB(J) + NSR )
IF ( M.LT.R ) HW = CMPLX(1.0,0.0)

CALCULATE RESTORED IMAGE
F(J) = G(J) * HW
2 CONTINUE

CALL WRBLK(3, 16*I, F, 16, IER)
CALL CHECK(IER)

3 CONTINUE

CALL RESET
STOP"<7><7><7>CONST.FR"
END
This program implements the segmentation method. Its input is the degraded image. It divides the image into square regions (subimages), each subimage is then converted to complex representation, and Fourier transformed. Finally, the magnitude squares of the Fourier transform of each subimage are summed and averaged by the number of subimages.

LOAD LINE:
RLDR SEGEST SUBIMG SVTC SFFT2 FFT2 F5PICBUF. LD @FLIB@

COMMAND LINE:

SEGEST
Program will ask for the I/O file names and two temporary file names, size of the subimages, and the shifting factor of the window which subimages are obtained by shifting this window.
The output file name will be the estimated power spectra.

INTEGER DIMGNM(7), TMPNM1(7), TMPNM2(7), OTFN(7)
INTEGER R, C, DELTHA, M, ISTR, ISTP, OPTION
COMPLEX IN1(1024), IN2(1024), OUT(1024)
COMPLEX CM, CMAC

I/O CONSTANTS
PARAMETER KTTYIN=11
PARAMETER KTTYOUT=10

ACCEPT"ENTER DEGRADED IMAGE FILE NAME:.......>
READ(KTTYIN, 40) DIMGNM(1)
ACCEPT"ENTER OUTPUT FILE NAME:.......>
READ(KTTYIN, 40) OTFN(1)
ACCEPT"ENTER TEMPORARY FILE NAME #1:.......>
READ(KTTYIN, 40) TMPNM1(1)
ACCEPT"ENTER TEMPORARY FILE NAME #2:.......>
READ(KTTYIN, 40) TMPNM2(1)

40 FORMAT(S40)

C

10 TYPE"ENTER SIZE OF THE SUBIMAGES"
ACCEPT"128, 64, 32:.......>" , ISIZE
ACCEPT"ENTER SHIFING FACTOR:.......>" , DELTHA
IF(ISIZE.EQ.128) GO TO 20
IF(ISIZE.EQ.64) GO TO 20
IF(ISIZE.EQ.32) GO TO 20
GO TO 10
20 CONTINUE
CALL DFILW(OTFNMI, IER)
CALL CFILW(OTFNMI, 2, IER) ; 2 random file
CALL CHECK(IER)
OPEN 7, OTFNMI, ATT="OR", ERR=120
CONTINUE

CALL CFILW(TMPNM1, 2, IER) ; initialize
CALL CHECK(IER)

CALL CFILW(TMPNM2, 2, IER)
CALL CHECK(IER)
ISTR=1 ; starting point(1,1) for first segment
ISTP=256-ISIZE
DELTHA=DELTHA-1
M=0 ; number of segments
OPTION=1 ; 2D forward Fourier transform
IFACTOR=256/ISIZE
ITIMES=64/IFACTOR**2-1

DO 2 I=0, ITIMES
DO 3 J=1, 1024
OUT(J)=CMPLX(0.0,0.0)
3 CONTINUE
2 CONTINUE

DO 25 R=1, ISTP, DELTHA
DO 35 C=1, ISTP, DELTHA
M=M+1
99 CALL SUBIMG(DIMCNM, TMPNM1, ISIZE, C, R) ; segmentation
CALL SVTC(ISIZE, TMPNM1, TMPNM2) ; convert to complex
CALL SSFT2(TMPNM2, OPTION) ; 2D-forward FFT
OPEN 2, TMPNM2, ATT="OR", ERR=101
OPEN 7, OTFNM, ATT="ORR", ERR=101
101 CONTINUE

DO 55 I=0, ITIMES
CALL RDBLK(2, 16*I, IN1, 16, IER)
CALL CHECK(IER)

CALL RDBLK(7, 16*I, IN2, 16, IER)
CALL CHECK(IER)
DO 50 J=1, 1024
OUT(J)=IN2(J) + IN1(J)*CONJG(IN1(J))
50 CONTINUE
CALL WRBLK(7, 16*I, OUT, IER)
CALL CHECK(IER)
55 CONTINUE
TYPE"SEGMENT #", M
35 CONTINUE

25 CONTINUE
CM=CMPLX(FLOAT(M), 0.0)
DO 77 I=0, ITIMES
CALL RDBLK(7, 16*I, OUT, IER)
CALL CHECK(IER)
DO 87 J=1, 1024
OUT(J)=OUT(J)/CM ; average by M
87 CONTINUE
CALL WRBLK(7, 16*I, OUT, IER)
CALL CHECK(IER)
77 CONTINUE
TYPE"NUMBER OF SEGMENTS=", M
CALL RESET
STOP "<7><7><7>SEDEST.FR"
END
This program implements the segmentation method. Its input is the degraded image. It divides the degraded image into M subimages. The magnitude of the Fourier transform of each subimage is then multiplied and 1/M power of the result is calculated to estimate the magnitude of the degrading OTF.

**RELOAD LINE:**

RLDR SEGEST2 SVTC SFFT2 FFT2 UNP F5PICBUF.LB @FLIB@

**COMMAND LINE:**

SEGEST2

Program will ask for the I/O and two temporary file names, the size of the subimages and the shifting factor. The output will be the magnitude of the estimated degrading OTF.

```
INTEGER DIMGNM(7), TMPNM1(7), TMPNM2(7), OTFN(7)
INTEGER R, C, DELTHA, M, ISTR, ISTP, OPTION
COMPLEX IN1(1024), IN2(1024), OUT(1024)
REAL R1, R2, RM
I/O CONSTANTS
PARAMETER KTTYIN=11
PARAMETER KTTYOUT=10

ACCEPT"ENTER DEGRADED IMAGE FILE NAME :.... >"
READ(KTTYIN, 40) DIMGNM(1)
ACCEPT"ENTER OUTPUT FILE NAME:...... >"
READ(KTTYIN, 40) OTFN(1)
ACCEPT"ENTER TEMPORARY FILE NAME #1..... >"
READ(KTTYIN, 40) TMPNM1(1)
ACCEPT"ENTER TEMPORARY FILE NAME #2..... >"
READ(KTTYIN, 40) TMPNM2(1)
FORMAT(S40)

10 TYPE"ENTER SIZE OF THE SUBIMAGES"
ACCEPT"  128, 64, 32 ....... >", ISIZE
ACCEPT"ENTER SHIFTING FACTOR ....... >", DELTHA
IF(ISIZE.EQ.128) GO TO 20
IF(ISIZE.EQ.64) GO TO 20
IF(ISIZE.EQ.32) GO TO 20
GO TO 10
GO TO 10
CONTINUE
```
CALL DFILW(OTFNM, IER)
CALL CFILW(OTFNM, 2, IER) ; 2 random file
CALL CHECK(IER)
OPEN 7, OTFNM, ATT="OR", ERR=120
CONTINUE

CALL CFILW(TMPNM1, 2, IER) ; initialize
CALL CHECK(IER)

CALL CFILW(TMPNM2, 2, IER)
CALL CHECK(IER)
ISTR=1 ; starting point(1,1) for first segment
ISTP=256-ISIZE
DELTHA=DELTHA-1
M=0 ; number of segments
OPTION=1 ; 2D forward Fourier transform
IFACTOR=256/ISIZE
ITIMES=64/IFACTOR**2-1

DO 2 I=0, ITIMES
   DO 3 J=1, 1024
      OUT(J)=CMPLX(1.0,0.0) ; initialize
   CONTINUE

CALL WRBLK(7, 16*I, OUT, 16, IER)
CALL CHECK(IER)
CONTINUE

DO 25 R=1, ISTP, DELTHA
   DO 35 C=1, ISTP, DELTHA
      M=M+1
      CALL SUBIMG(DIMGNM, TMPNM1, ISIZE, C, R) ; segmentation
      CALL SVTC(ISIZE, TMPNM1, TMPNM2) ; convert to complex
      CALL SFFT2(TMPNM2, OPTION) ; 2D-forward FFT
      OPEN 2, TMPNM2, ATT="OR", ERR=101
      OPEN 7, OTFNM, ATT="OR", ERR=101
      CONTINUE

   DO 55 I=0, ITIMES
      CALL RDBLK(2, 16*I, IN1, 16, IER)
      CALL CHECK(IER)
      CALL RDBLK(7, 16*I, IN2, 16, IER)
      CALL CHECK(IER)

      25 CONTINUE
      35 CONTINUE
      55 CONTINUE
DO 50 J=1, 1024

R1 = REAL(IN1(J))**2 + AIMAG(IN1(J))**2
R2 = REAL(IN2(J))**2 + AIMAG(IN2(J))**2
R1 = R1**0.5
R2 = R2**0.5
RR = R1*R2
OUT(J) = CMPLX(RR, 0.0)

50 CONTINUE
CALL WRBLK(7, 16*I, OUT, IER)
CALL CHECK(IER)
55 CONTINUE
TYPE"SEGMENT #", M
35 CONTINUE

25 CONTINUE

109 RM = 1.0/(FLOAT(M))

DO 77 I=0, ITIMES

CALL RDBLK(7, 16*I, OUT, IER)
CALL CHECK(IER)
DO 87 J=1, 1024
RR = REAL(OUT(J))** RM
OUT(J) = CMPLX(RR, 0.0)
87 CONTINUE
CALL WRBLK(7, 16*I, OUT, IER)
CALL CHECK(IER)
77 CONTINUE

TYPE"NUMBER OF SEGMENTS=", M
CALL RESET
STOP "<7><7><7>SEGEST2.FR"
END
SEGEST3.FR  DG  FORTRAN 5 - LT MURAT CINCIOLU, AUG 1986

This program obtains implements the segmentation method. Obtains an estimate of the complex logarithm of degrading OTF by averaging the complex logarithm of the Fourier transform of subimages.

RELOAD LINE:

RLDR SEGEST3 SUBIMG SVTC UNP SFFT2 F5PICBUF LB @FLIB@

COMMAND LINE:

SEGEST3

Program will ask for I/O file and two temporary file names and for the shifting factor, and the size of the subimages.

INTEGER DIMGNM(7), TMPNM1(7), TMPNM2(7), OTFNM(7)
INTEGER R, C, DELTHA, M, ISTR, ISTP, OPTION
COMPLEX IN1(1024), IN2(1024), OUT(1024)
COMPLEX CM
REAL ROUT, IDUT, RR, II

I/O CONSTANTS
PARAMETER KTTYIN=11
PARAMETER KTTYOUT=10

ACCEPT"ENTER DEGRADED IMAGE FILE NAME: .... >"
READ(KTTYIN, 40) DIMGNM(1)
ACCEPT"ENTER OUTPUT FILE NAME: .... >"
READ(KTTYIN, 40) OTFNM(1)
ACCEPT"ENTER TEMPORARY FILE NAME #1: .... >"
READ(KTTYIN, 40) TMPNM1(1)
ACCEPT"ENTER TEMPORARY FILE NAME #2: .... >"
READ(KTTYIN, 40) TMPNM2(1)
FORMAT(S40)

Type"ENTER SIZE OF THE SUBIMAGES"
ACCEPT"  128, 64, 32: .... >", ISIZE
ACCEPT"ENTER SHIFTING FACTOR: .... >", DELTHA
IF(ISIZE.EQ.128) GO TO 20
IF(ISIZE.EQ.64) GO TO 20
IF(ISIZE.EQ.32) GO TO 20
GO TO 10
CONTINUE
CALL DFILE(OTFNM,IER)
CALL CFILW(OTFNM,1,IER) ; 2 random file
CALL CHECK(IER)
OPEN 7, OTFNM, ATT="ORA", ERR=120
CONTINUE

CALL CFILW(TMPNM1,1,IER) ; initial
CALL CHECK(IER)

CALL CFILW(TMPNM2,2,IER)
CALL CHECK(IER)
ISTR=1 ; starting point(1,1) for first segment
ISTP=256-ISIZE
DELTHA=DELTHA-1
M=0 ; number of segments
OPTION=1 ; 2D forward Fourier transform
IFACTOR=256/ISIZE
ITIMES=64/IFACTOR**2-1
DO 2 I=0, ITIMES
  DO 3 J=1, 1024
       OUT(J)=CMPLX(0.0,0.0)
  CONTINUE
  CALL WRBLK(7, 16*I, OUT, 16, IER)
  CALL CHECK(IER)
  CONTINUE
  DO 25 R=1, ISTP, DELTHA
    DO 35 C=1, ISTP, DELTHA
      M=M+1
      CALL SUBIMG(DIMGNM, TMPNM1, ISIZE, C, R) ; segmentation
      CALL SVTC(ISIZE, TMPNM1, TMPNM2) ; convert to complex
      CALL SFFT2(TMPNM2, OPTION) ; 2D-forward FFT
      OPEN 2, TMPNM2, ATT="ORA", ERR=101
      OPEN 7, OTFNM, ATT="ORA", ERR=101
      CONTINUE
      DO 55 I=0, ITIMES
        CALL RDQBLK(2, 16*I, IN1, 16, IER)
        CALL CHECK(IER)
        CALL RDQBLK(7, 16*I, IN2, 16, IER)
        CALL CHECK(IER)
      CONTINUE

B-17
DO 50 J=1, 1024
OUT(J)=IN2(J) + CLOG(IN1(J))

CONTINUE
CALL WRBLK(7, 16*IOUT, OUT, IER)
CALL CHECK(IER)

55 CONTINUE
TYPE "SEGMENT #", M

35 CONTINUE

DO 77 I=1, ITIMES
CALL RDBLK(7, 16*IOUT, OUT, IER)
CALL CHECK(IER)

DO 87 J=1, 1024
OUT(J)=OUT(J)/CM ; average by M

87 CONTINUE
CALL WRBLK(7, 16*IOUT, OUT, IER)
CALL CHECK(IER)

77 CONTINUE
TYPE "NUMBER OF SEGMENTS=", M
CALL RESET
STOP "<7><7><7><7> SEGES13.FR"
END
This program obtains a degrading OTF for the motion blur in the horizontal direction. The degrading OTF of the motion blur in vertical direction can be obtained by transforming the output file.

**RELOAD LINE:**

RLDR MTRF @FLIB@

**COMMAND LINE:**

MTRF

Program will ask for the degrading OTF file name, total exposure time (T), and total displacement distance (d), and the size of the output file.

```fortran
COMPLEX TRF(256)
PARAMETER KTTYIN=11, KTTYOUT=10, PI=3.1415927
REAL R1, R2, T, d, R, RO, X
INTEGER OFLNM(7)
WRITE(KTTYOUT, 12)
FORMAT('ENTER FILE NAME ........'), Z)
READ(KTTYIN, 14) OFLNM(1)
FORMAT( S39 )

TYPE" 
ACCEPT"ENTER EXPOSURE TIME ........"", T
TYPE" 
ACCEPT"ENTER TOTAL DIPLACEMENT ("d")..."", d

CALL DFILW(OFLNM, IER)
CALL CFILW(OFLNM, 2, KER)
CALL CHECK( KER )
CALL FOPEN(2, OFLNM)

CALCULATE TRANSFER FUNCTION

DO 20 U=1,256
  X= U-128.0
  RO= PI*X*d
  IF( X EQ. 0.0 ) R=T
  IF( X. NE. 0.0 ) R=( T*SIN( RO ) )/( RO )
  R1= R*COS( RO )
  R2= -R*SIN( RO )
  TRF(U) = CMPLX( R1, R2 )
CONTINUE
```
TYPE "WRITING TRANSFER FUNCTION"
DO 30 I=1,256
    DO 40 J=1,256
        WRITE(2) TRF(J)
        CONTINUE
    IF( MOD(I,50).EQ.0 ) TYPE "DONE", I
40 CONTINUE
30 CONTINUE
CLOSE 2
STOP "<7><7><7>MTRF.FR"
END
This program calculates two-dimensional point spread function for an out-of-focus camera with circular aperture in the normalized coordinates.

**RELOAD LINE:**
RLDR DPSF @FLIB@

**COMMAND LINE:**
DPSF

Program will ask for an output file name, create the output file and then ask for file size, the radius of the defocusing error.

```c
COMPLEX OUT(1024)
INTEGER OFLNMI(7)
REAL R,RD,VAL,VAL2
REAL X,Y,RSIZE,OR
PARAMETER PI=3.1415927

C ACCEPT INPUT
C
1 FORMAT(S13)
   ACCEPT"ENTER OUTPUT FILENAME ->"
   READ(11,1)OFLNMI(1)
   CALL DFILW (OFLNMI,IER)
   CALL CFILW (OFLNMI,2,KER)
   CALL CHECK(KER)
   OPEN 3, OFLNMI,ATT="OR",ERR=100

100 CONTINUE
5 TYPE""
   TYPE"ENTER SIZE OF FILES."
   ACCEPT"256, 128, 64, OR 32 -->",ISIZE
   ACCEPT"RADIUS OF DEFOCUSING -->",R

C TEST FOR ONLY THE ALLOWABLE SIZES
C
IF(ISIZE.EQ.256) GO TO 6
IF(ISIZE.EQ.128) GO TO 6
IF(ISIZE.EQ.64) GO TO 6
IF(ISIZE.EQ.32) GO TO 6
GO TO 5
6 CONTINUE ; size is okay
   RSIZE = ISIZE ; type conversion
```
OR = KSIZE/2.0

DO 3 I=1,ISIZE

DO 2 J=1,ISIZE
   X = J
   X = (X - OR)/OR; translate to center
   Y = I
   Y = (Y - OR)/OR
   RD = X**2 + Y**2
   VAL2 = PI*RD
   RD = SQRT(RD)
   VAL = 0.0
   IF (RD.LE.R) VAL = 1.0/VAL2
   OUT(J) = CMPLX(VAL,0.0)
   WRITE(3) OUT(J)
2    CONTINUE
   IF (MOD(I,50).EQ.0) TYPE" DONE ... >",I
3    CONTINUE

CALL RESET
STOP"<7><7><7>DPSF.FR"
END
This program obtains a degrading OTF for defocused camera with circular aperture.

**RELOAD LINE:**

**RLDR BESSEL @FLIB@**

**COMMAND LINE:**

**BESSEL**

Program will ask for the degrading OTF file name,
radii of the first minima, number of terms to approximate
the Bessel function of first kind with order one, and the
size of the output file.

```fortran
COMPLEX TRF(256)
PARAMETER KTTYIN=11, KTTYOUT=10
REAL C(256), X, R, F, U, V, OR, C2(256), WX, WY, SIZE, APRX, J1, J, N, K
INTEGER OFLNM(7)
REAL K2, A

WRITE(KTTYOUT, 12)
12 FORMAT("ENTER FILE NAME FOR THE TRANSFER FUNCTION >", Z)
READ(KTTYIN, 14) OFLNM(1)
14 FORMAT( S39 )

TY E" 
TYPE"ENTER DEFOCUSING ERROR( Radii of the first minima)" 
ACCEPT" 
TYPE" 
TYPE"HOW MANY TERMS SHOULD BE CALCULATED FOR THE BESSEL FUNC." 
ACCEPT" 
ACCEPT"ENTER SIZE OF THE OUTPUT FILE" 
CALL DFILW(OFLNM, IER)
CALL CFILW(OFLNM, 2, KER)
CALL CHECK(KER)
CALL FOPEN(2, OFLNM)
```
C
CALCULATE TRANSFER FUNCTION
C2(I)=SIZE/2.0
DO 50 I=1,APRX
C(I)=1.0
C2(I)=1.0
50
CONTINUE
DO 100 I=1,APRX
X=I
DO 200 J=1.0,X
C(I)=C(I)*J
C2(I)=C2(I)*J
200
CONTINUE
C2(I)=C2(I)* (X+1.0)
100
CONTINUE
N=APRX
DO 20 WY=1.0,SIZE
DO 30 WX=1.0,SIZE

V=(WY-OR)/OR; normalized coord.
U=(WX-OR)/OR; translate org. to the cntr. of the file
F=(U**2+V**2)**0.5
X=2.0*R*F
IF(X.EQ.0.0) GO TO 310
J1=X/2.0
A=1.0
DO 300 I=1,APRX
K=I
K2=1.0+2.0*K
IF(MOD(I,2).EQ.0) A=-1.0
J1=J1-A* ((X/2.0)**2*K2 )/(C(I)*C2(I))
300
CONTINUE
310
IF(X.EQ.0.0) TRF(WX)=CMPLX(1.0,0.0)
IF(X.EQ.0.0) GO TO 19
J1=J1/(R*F)
IF(X.NE.0.0) TRF(WX)=CMPLX(J1,0.0)
19
WRITE(2) TRF(WX)
30
CONTINUE
IF(MOD(WY,OR).EQ.0.0) TYPE"DONE....", WY
20
CONTINUE
CLOSE 2
STOP"<7><7><7>BESSEL FR"
END
This program implements the line estimation method. It estimates a line in the vertical direction and duplicates this line to obtain an output file which is identical in size to the input file.

RELOAD LINE.

COMMAND LINE:

LEST
Program will ask for the input file size, and input/output file names

COMPLEX IN1(256), OUT1(256), OUT(256)
COMPLEX NB, EST
INTEGER INFLNM(7), OFLNM(7), JMP, K, M, N, STEP, WINDOW, KSIZE

ACCEPT INPUT
TYPE "INPUT FILE SHOULD BE FOURIER TRANSFORM OF "
TYPE "THE DEGRADED IMAGE..."
TYPE ""
WRITE(10, 12)
12 FORMAT("ENTER INPUT FILE NAME = -->", Z)
READ(11, 14) INFLNM(1)
14 FORMAT(S39)

WRITE(10, 16)
16 FORMAT("ENTER OUTPUT FILE NAME = -->", Z)
READ(11, 14) OFLNM(1)

TYPE " ENTER SIZE OF INPUT FILE :
ACCEPT "32, 64, 128, 256 -->", ISIZE
TYPE ""
TYPE " ENTER WINDOW SIZE "
ACCEPT "1, 2, 4, 8, 16, 32, 64, 128 -->", WINDOW

WINDOW = 1
CALL OPEN(1, INFLNM, 1, IER)
CALL CHECK(IER)
CALL DFILW(OFLNM, IER)
CALL CFILW(OFLNM, 2, KER)
CALL CHECK(KER)
CALL FOPEN(2, OFLNM)
CONTINUE

REWIND 1
REWIND 2

C

KSIZE = ISIZE - 3

STEP = ISIZE / WINDOW ; number of windows
JNP = STEP * 1
NB = CMPLX(STEP, 0.0)

DO 100 I = 1, ISIZE
   DO 200 J = 1, ISIZE
      READ(1) IN1(J) ; READ ONE ROW AT A TIME
   200 CONTINUE
   DO 300 K = 1, WINDOW
      EST = CMPLX(0.0, 0.0)
      DO 400 J = 0, JNP
         KW = K + J * WINDOW
         EST = EST + CMPLX(IN1(KW))
   400 CONTINUE
   OUT1(K) = EST / NB
300 CONTINUE

C

ENLARGE THE PROCESSED WINDOW INTO "ISIZE"
M = 0
DO 500 J = 1, KSIZE, STEP
   M = M + 1
   DO 600 N = 0, JNP
      OUT(J + N) = OUT1(M)
600 CONTINUE
500 CONTINUE
DO 700 J = 1, ISIZE
   WRITE(2) OUT(J)
700 CONTINUE
IF(MOD(I, 64).EQ. 0) TYPE' DONE ", I
100 CONTINUE
CLOSE1
CLOSE2
STOP<7><7><7><7><7>LEST FR"
END
This program estimates degrading OTF from degraded image and calculates Wiener filter transfer function for the restoration filter.

LOGTRF estimates the OTF by taking complex logarithm of Fourier transform of the degraded image and dividing by total number of pixels in a line.

RELOAD LINE:
  RLDR LOGTRF @FLIB@

COMMAND LINE:
  LOGTRF

Program will ask for input and output file name, create the output file and then ask for file size. Input file name should be Fourier transform of the degraded image. Noise-to-Signal ratio is also asked for Wiener filter. Noise-to-signal ratio is assumed to be constant.

COMPLEX HB(1024), G(1024), HW(1024), OUT(1024)
INTEGER INFLNM1(7), OFLNM1(7)
REAL NS

ACCEPT INPUT

ACCEPT"ENTER DEGRADED IMAGE FILE NAME ....>
READ(11,1)INFLNM1(1)
1 FORMAT(S13)
ACCEPT"ENTER OUTPUT FILENAME -->
READ(11,1)OFLNM1(1)

CALL OPEN(1, INFLNM1, 1, IER)
CALL CHECK(IER)

CALL DFILW (OFLNM1, IER)
CALL CFILW (OFLNM1,2, KER)
CALL CHECK(KER)
OPEN 3, OFLNM1, ATT="OR", ERR=100

100 CONTINUE

5 TYPE" "
   TYPE"ENTER SIZE OF FILES 
   ACCEPT" 256, 128, 64, OR 32 -->", ISIZE
C TEST FOR ONLY THE ALLOWABLE SIZES

IF(ISIZE.EQ.256) GO TO 6
IF(ISIZE.EQ.128) GO TO 6
IF(ISIZE.EQ.64) GO TO 6
IF(ISIZE.EQ.32) GO TO 6
GO TO 5

6 CONTINUE ; size is okay

IFACTOR=256/ISIZE ; compute scaling factor
ITIMES=64/IFACTOR**2-1
SIZE = CMPLX(ISIZE**2,0.0)
CNS = CMPLX(NS,0.0)

DO 3 I=0,ITIMES
    CALL RDBLK(1,16*I,G,16,IER)
    CALL CHECK(IER)
    DO 2 J=1,1024
        HB(J) = CMPLX(0(J),0.0) / SIZE ; estimated OTF of degrad
        HW(J) = CONJG(HB(J)) / (HB(J)*CONJG(HB(J)) + CNS) ; W
        OUT(J) = HW(J)*G(J)
    2 CONTINUE
    CALL WRBLK(3,16*I,OUT,16,IER)
    CALL CHECK(IER)
    3 CONTINUE
    CALL RESET
STOP "<7><7><7>LOGTRF"
END
This program computes the cepstrum of the degraded image. The input file should be the Fourier transform of the degraded image.

RELOAD LINE:
RLDR CEPST SFFT 2 @FLIB@

COMMAND LINE:
CEPST.FR

Program will ask for input and output file name, create the output file and then ask for file size.

COMPLEX IN1(1024), OUT(1024)
COMPLEX CEP
REAL AMP
INTEGER INFLNM1(7), OFLNM1(7)

ACCEPT INPUT

ACCEPT "ENTER INPUT FILENAME #1 -> "
READ(11, 1) INFLNM1(1)
1 FORMAT (S13)
ACCEPT "ENTER OUTPUT FILENAME -> "
READ(11, 1) OFLNM1(1)

CALL OPEN (1, INFLNM1, 1, IER)
CALL CHECK (IER)
CALL DFILW (OFLNM1, IER)
CALL CFILW (OFLNM1, 2, KER)
CALL CHECK (KER)
OPEN 8, OFLNM1, ATT="OR", ERR=100

100 CONTINUE
5 TYPE " "
TYPE "ENTER SIZE OF FILES . "
ACCEPT " 256, 128, 64, OR 32 -> " , ISIZE

TEST FOR ONLY THE ALLOWABLE SIZES

IF (ISIZE EQ. 256) GO TO 6
IF (ISIZE EQ. 128) GO TO 6
IF (ISIZE EQ. 64) GO TO 6
IF (ISIZE EQ. 32) GO TO 6
GO TO 5
6 CONTINUE ; size is okay
DO 2 J=1, ISIZE

DO 4 K=1, ISIZE
READ(1) IN1(K)
AMP= REAL(IN1(K)) **2 + AIMAG(IN1(K))**2
AMP= SQRT(AMP)
AMP= LOG(AMP)
C IF(AMP. NE. 0.0) AMP = LOG(AMP)
C IF(AMP. EQ. 0.0) AMP = 0.0
CEP = CMPLX(AMP, 0.0)
OUT(K) = CEP
WRITE(6) OUT(K)
CONTINUE
4 CONTINUE
2 CONTINUE
CLOSE 8
CALL SFFT2(OFNM1, 1)

STOP"<7><7><7>CEPST.FR"
END
This program applies the Laplacian operator to the input image.

RELOAD LINE:

RLDR LAP F5PICBUF.LB @FLIB@

COMMAND LINE:

LAP

PARAMETER NBUFSZ=300, KTTYOUT=10, KTTYIN=11
PARAMETER NCOLSMAX=256
INTEGER IBUF(NBUFSZ)
INTEGER IARRAY(NCOLSMAX,3), IARRAY2(NCOLSMAX), INAME(20)

WRITE (KTTYOUT,996)
996 FORMAT("INPUT PICTURE FILE NAME = ",Z)
READ (KTTYIN,997) INAME(1)
997 FORMAT(S39)
CALL PICFMT(IBUF,INAME,IFLC)
CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)
IF (IMODE EQ.3) GO TO 10
IF ((NROWS LT.3).OR.(NCOLS LT.3)) GO TO 10
IF (NCOLS GT.NCOLSMAX) GO TO 10
IF (NBITS GT.11) GO TO 10 , to preclude integer overflow
CALL MAKB(IBUF)
CALL PICIN(IBUF,INAME,IFLC)

CALL GROW(IBUF,1,1,NCOLS,IARRAY(1,1))
CALL GROW(IBUF,2,1,NCOLS,IARRAY(1,2))
I1=0
I2=1
I3=2
NR1=NROWS-1
NC1=NCOLS-1
IARRAY2(1)=0
IARRAY2(NCOLS)=0
PERFORM CONVOLUTION A ROW AT A TIME

DO 200 I=2, NR1
IF (MOD(I, 10) EQ 0) WRITE(KTTYOUT, 998) I
998 FORMAT("ROW = ", I3)

I1 = MOD(I1, 3) + 1
I2 = MOD(I2, 3) + 1
I3 = MOD(I3, 3) + 1

CALL GROW(IBUF, I+1, 1, NCOLS, IARRAY(1, I3))

DO 100 J=2, NCOLS
IVAL = IARRAY(J-1, 11)*0 - IARRAY(J, 11) + IARRAY(J+1, 11)*0
IVAL = IVAL - IARRAY(J-1, 12) + IARRAY(J, 12)*5 - IARRAY(J+1, 12)
IVAL = IVAL + IARRAY(J-1, 13)*0 - IARRAY(J, 13) + IARRAY(J+1, 13)*0
if (IVAL LT 0) IVAL = (-1)*IVAL
if (IVAL GT 15) IVAL = 15
IARRAY2(J) = IVAL  (laplacian of image)
100 CONTINUE

CALL PROW(IBUF, I, 1, NCOLS, IARRAY2)
200 CONTINUE

C OUTPUT RESULT AND RELEASE BUFFERS

WRITE (KTTYOUT, 999)
999 FORMAT("OUTPUT PICTURE FILE NAME = ", Z)
READ (KTTYIN, 997) INAME(1)
CALL PICOUT(IBUF, INAME, IFLC)
CALL RELB(IBUF)
STOP"<7><7><7>LAP.FR"
END
PARAMETER KTTYIN=11, KTTYOUT=10, NBUFSZ=300
PARAMETER MAXCOL=256, MAXROW=256
INTEGER IBUF1(NBUFSZ), IBUF2(NBUFSZ), IBUF(NBUFSZ)
INTEGER INAME1(7), INAME2(7), ONAME(7)
INTEGER IARRAY1(MAXCOL), IARRAY2(MAXCOL), IARRAY(MAXCOL)

WRITE(KTTYOUT, 990)
FORMAT(1X, "ENTER PICTURE FILE NAME #1 : ...>", Z)
READ(KTTYIN, 991) INAME1(1)
FORMAT(S39)

WRITE(KTTYOUT, 993)
FORMAT(1X, "ENTER PICTURE FILENAME #2 : ...>", Z)
READ(KTTYIN, 991) INAME2(1)

WRITE(KTTYOUT, 992)
FORMAT(1X, "ENTER OUTPUT FILE NAME -->", Z)
READ(KTTYIN, 991) ONAME(1)

CALL PICFMT(IBUF1, INAME1, IHDR)
CALL MAKB(IBUF1)
CALL PICIN(IBUF1, INAME1, IHDR)
CALL GFMT(IBUF1, NROWS, NCOLS, NBITS, IMODE)

CALL PICFMT(IBUF2, INAME2, IHDR)
CALL MAKB(IBUF2)
CALL PICIN(IBUF2, INAME2, IHDR)
CALL GFMT(IBUF2, NROWS, NCOLS, NBITS, IMODE)
CALL PFMT(IBUF1, NROWS, NCOLS, NBITS, IMODE)
CALL MAKB(IBUF1)
DO 500 I=1, NRDWS
CALL GROW(IBUF1, 1, 1, NCOLS, IARRAY1) ; get rows
CALL GROW(IBUF2, 1, 1, NCOLS, IARRAY2)
DO 600 J=1, 256
   IVAL=IARRAY1(J) + IARRAY2(J) ; add rows
   IF(IVAL.LT.0) IVAL=-1*IVAL
   IF(IVAL.GT.15) IVAL=15
   IARRAY(J)=IVAL
600 CONTINUE
IF(MOD(I,64).EQ.0) TYPE"DONE ...", I ; report to keyboard
CALL PROW(IBUF, 1, 1, NCOLS, IARRAY)
500 CONTINUE

CALL PICOUT(IBUF, ONAME, 0)
CALL RELB(IBUF1)
CALL RELB(IBUF2)
CALL RELB(IBUF)
CALL RESET
STOP "<7><7><7>ADV"
END
This program adds two complex image to obtain the output file.

RELOAD LINE:
   RLDR VCM @FLIB@

COMMAND LINE:
   ADC

Program will ask for input and output file name, create the output file and then ask for file size.

COMPLEX IN1(1024), IN2(1024), OUT(1024)
INTEGER INFLNM1(7), INFLNM2(7), OFLNM1(7)

ACCEPT INPUT

ACCEPT"ENTER INPUT FILENAME #1 ->"
READ(11,1)INFLNM1(1)
ACCEPT"ENTER INPUT FILENAME #2 ->"
READ(11,1)INFLNM2(1)
1 FORMAT(S13)
ACCEPT"ENTER OUTPUT FILENAME ->"
READ(11,1)OFLNM1(1)

CALL OPEN(1, INFLNM1, 1, IER)
CALL CHECK(IER)
CALL OPEN(2, INFLNM2, 1, IER)
CALL CHECK(IER)

CALL DFILW (OFLNM1, IER)
CALL CFILEW (OFLNM1, 2, KER)
CALL CHECK(KER)
OPEN 3, OFLNM1, ATT="OR", ERR=100

100 CONTINUE

5 TYPE" "
   TYPE"ENTER SIZE OF FILES:"
ACCEPT" 256, 128, 64, OR 32 -->", ISIZE
C TEST FOR ONLY THE ALLOWABLE SIZES
C
IF(ISIZE.EQ.256) GO TO 6
IF(ISIZE.EQ.128) GO TO 6
IF(ISIZE.EQ.64) GO TO 6
IF(ISIZE.EQ.32) GO TO 6
GO TO 5
6 CONTINUE ; size is okay
IFACTOR=256/ISIZE ; compute scaling factor
ITIMES=64/IFACTOR**2-1
DO 3 I=0, ITIMES
   CALL RDBLK(1, 16*I, IN1, 16, IER)
   CALL CHECK(IER)
   CALL RDBLK(2, 16*I, IN2, 16, IER)
   CALL CHECK(IER)
   DO 2 J=1, 1024
      OUT(J)=IN1(J)+IN2(J)
2 CONTINUE
   CALL WRBLK(3, 16*I, OUT, 16, IER)
   CALL CHECK(IER)
3 CONTINUE
   CALL RESET
STOP "<7><7><7>ADC"
END

B-36
This program obtains the mean, variance, mean square, maximum, and minimum values of a picture or a given region in the picture.

**RELOAD LINE:**

```
RLDR STAT F5PICBUF.LB @FLIB@
```

**COMMAND LINE:**

```
STAT
```

Program will ask for the picture file name and the coordinates(row, col) of the region where the mean, variance, mean square to be calculated.

```c
REAL MEAN, MEANSQ, VAR, STDEV, PR, SN, L(16), SV, N
INTEGER K, V, Y, M, MIN, MAX
PARAMETER NBUFSZ=300, KTTYOUT=10, KTTYIN=11
PARAMETER NCOLSMAX=256
INTEGER IBUF(NBUFSZ)
INTEGER INAME(20)
```

**INITIALIZE INPUT BUFFER AND READ IN PICTURE FILE**

```
10 WRITE(KTTYOUT,996)
996 FORMAT("ENTER INPUT PICTURE FILE NAME =",Z)
READ(KTTYIN,997) INAME(1)
997 FORMAT(S39)
```

```
CALL PICFMT(IBUF, INAME, IFLG)
CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)
IF(IMODE.EQ.3) GO TO 10
IF((NROWS.LT.3) OR (NCOLS.LT.3)) GO TO 10
IF(NCOLS.GT.NCOLSMAX) GO TO 10
IF(NBITS.GT.11) GO TO 10
CALL MAKB(IBUF)
CALL PICIN(IBUF, INAME, IFLG)
```

```
TYPE" ENTER REGION BOUNDARIES"
ACCEPT"ROW BOUNDARY FROM-->": K
ACCEPT" TO -->": M
ACCEPT"COLUMN BOUNDARY FROM-->": V
```
ACCEPT " TO -->", Y  

MAX=0.0
MIN=15.0
N=0.0
DO 2 I=0,15 ; initialize.
   L(I)=0.0
   CONTINUE

DO 20 I=K,M ; count possible intensity levels.
   CALL GPNT(IBUF, I, J, IVAL)
   IF (IVAL .LT. MIN) MIN=IVAL ; max, min search.
   IF (IVAL .GT. MAX) MAX=IVAL
   IF (IVAL .EQ. 0) L(0)=L(0)+1.0
   IF (IVAL .EQ. 1) L(1)=L(1)+1.0
   IF (IVAL .EQ. 2) L(2)=L(2)+1.0
   IF (IVAL .EQ. 3) L(3)=L(3)+1.0
   IF (IVAL .EQ. 4) L(4)=L(4)+1.0
   IF (IVAL .EQ. 5) L(5)=L(5)+1.0
   IF (IVAL .EQ. 6) L(6)=L(6)+1.0
   IF (IVAL .EQ. 7) L(7)=L(7)+1.0
   IF (IVAL .EQ. 8) L(8)=L(8)+1.0
   IF (IVAL .EQ. 9) L(9)=L(9)+1.0
   IF (IVAL .EQ. 10) L(10)=L(10)+1.0
   IF (IVAL .EQ. 11) L(11)=L(11)+1.0
   IF (IVAL .EQ. 12) L(12)=L(12)+1.0
   IF (IVAL .EQ. 13) L(13)=L(13)+1.0
   IF (IVAL .EQ. 14) L(14)=L(14)+1.0
   IF (IVAL .EQ. 15) L(15)=L(15)+1.0
   N=N+1.0 ; number of elements
   CONTINUE

20 CONTINUE

MEAN=0.0
VAR=0.0
MEANSQ=0.0

DO 88 I=0.0,15.0
   PR =L(I)/N
   MEAN= MEAN+ I*PR
   VAR= VAR+ (I-MEAN)**2 *PR
   MEANSQ= MEANSQ+ ( I**2 )*PR
   CONTINUE
STDEV=VAR**0.5
SN=VAR/MEANsq

C
REPORT MEAN, MEAN SQUARE, VARIANCE, MAX., MIN.
TYPE" "
TYPE" MEAN =", MEAN
TYPE" "
TYPE" VARIANCE =", VAR

TYPE" "
TYPE" STANDARD DEVIATION =", STDEV
TYPE" "
TYPE" MEAN SQUARE =", MEANsq
TYPE" "
TYPE" (VARIANCE)/(MEAN SQUARE) =", SN
TYPE" "
TYPE" MINIMUM =", MIN
TYPE" MAXIMUM =", MAX

STOP"<7><7><7>STAT. FR"
END
This program is used to create the test image (Picture 1).

RELOAD LINE:

RLDR IMAGE LINE CIRCLE F5PICBUF.LB @FLIB@

COMMAND LINE:

IMAGE

Program will ask for the output file name and the background intensity level.

PARAMETER KTTYIN=11, KTTYOUT=10, NBUFSZ=300
INTEGER IBUF(NBUFSZ), IARRAY(256)
INTEGER X1, X2, Y1, Y2, INAME(7), IBGL, IVAL

CALLS SUBROUTINE LINE(IBUF,IVAL,X1,Y1,X2,Y2)
CALLS SUBROUTINE CIRCLE(IBUF,IVAL,X,Y,RADIUS)

X1<X2, Y1<Y2

write(KTTYOUT,12)
format("ENTER FILE NAME.......>",Z)
read(KTTYIN,14) INAME(1)
format(539)
type"The background and image levels should be an integer,"
type"between 0 and 15......."
accept"ENTER BACKGROUND LEVEL.......">", IBGL
accept"ENTER IMAGE LEVEL.......">", IVAL
type"

program running

CREATE BUFFER

CREATE BUFFER

call pfmt(IBUF,256,256,4,1)
call makb(IBUF)

FILL BUFFER WITH BACKGROUND

do 20 i= 1,256
       IARRAY(i)= IBGL ,
     continue

do 30 i= 1,256
       call prow(IBUF,i,1,256,IARRAY)
     continue
PRIMITIVES TO CREATE THE IMAGE

call CIRCLE(IBUF,IVAL,0,0,10)
call CIRCLE(IBUF,IVAL,0,0,20)
call CIRCLE(IBUF,IVAL,0,0,25)
call CIRCLE(IBUF,IVAL,0,0,26)
call CIRCLE(IBUF,IVAL,0,0,27)
call LINE(IBUF,IVAL,-110,110,110,110)
call LINE(IBUF,IVAL,-105,105,105,105)
call LINE(IBUF,IVAL,-100,100,100,100)
call LINE(IBUF,IVAL,-110,-110,110,-110)
call LINE(IBUF,IVAL,-105,-105,105,-105)
call LINE(IBUF,IVAL,-100,-100,100,-100)
call LINE(IBUF,IVAL,110,-110,110,-110)
call LINE(IBUF,IVAL,105,-105,105,-105)
call LINE(IBUF,IVAL,100,-100,100,-100)
call LINE(IBUF,IVAL,89,86,89,86)
call LINE(IBUF,IVAL,86,83,86,83)
call LINE(IBUF,IVAL,83,80,83,80)
call LINE(IBUF,IVAL,89,86,89,86)
call LINE(IBUF,IVAL,86,83,86,83)
call LINE(IBUF,IVAL,83,80,83,80)
call LINE(IBUF,IVAL,84,-71,84,71)
call LINE(IBUF,IVAL,87,-74,87,74)
call LINE(IBUF,IVAL,90,-77,90,77)
call LINE(IBUF,IVAL,-84,-71,-84,71)
call LINE(IBUF,IVAL,-87,-74,-87,74)
call LINE(IBUF,IVAL,-90,-77,-90,77)
call LINE(IBUF,IVAL,-50,50,50,50)
call LINE(IBUF,IVAL,50,-50,50,50)
call LINE(IBUF,IVAL,-50,-50,50,-50)
call LINE(IBUF,IVAL,-50,-50,-50,-50)
call LINE(IBUF,IVAL,-50,50,50,-50)
call LINE(IBUF,IVAL,-50,50,50,50)
call picout(IBUF,1NAME,1)
call relb(IBUF)
stop"<7><7><7>image.fr"
end

B-41
This subroutine is used to draw circles in the spatial domain (Ref. Chapter 4).

```fortran
subroutine circle(IBM,IVAL,X1,Y1,IR)
    integer IBM(300), IVAL, X1, Y1, IR
    integer X, Y, D, IROW, ICOL

    X = 0
    Y = IR
    D = 3 - 2*IR

    ICOL = (X+X1) + 128
    IROW = (-1)*(-Y+Y1)+128
    call ppnt(IBM, IROW, ICOL, IVAL)

    ICOL = (Y+X1) + 128
    IROW = (-1)*(-X+Y1)+128
    call ppnt(IBM, IROW, ICOL, IVAL)

    ICOL = (Y+X1) + 128
    IROW = (-1)*(-X+Y1)+128
    call ppnt(IBM, IROW, ICOL, IVAL)

    ICOL = (X+X1) + 128
    IROW = (-1)*(-X+Y1)+128
    call ppnt(IBM, IROW, ICOL, IVAL)

    if (X .eq. Y) call ppnt(IBM, IROW, ICOL, IVAL)
    if (X .ge. Y) go to 15

    if (D .ge. 0) go to 12
        D = D + 4*X + 6
        X = X + 1
    go to 10

    D = D + (X-Y) + 10
    Y = X + 1
    X = X + 1
    go to 10

    return
end
```

B-42
This Subroutine is used to draw line in the spatial domain (ref. Chapter 4)

Subroutine LINE(ibuf, ival, x1, y1, x2, y2)
integer ibuf, ival, x1, y1, x2, y2
integer lx, ly, incr1, incr2, sx, sy, ld, xend
integer intx, inty, dm
real dx, dy, x, y, m
if(x1. eq. x2) go to 300

dy=y2-y1
dx=x2-x1
m=dy/dx
if(m. lt. 0) go to 150
lx=x2-x1
ly=y2-y1
if(lx. lt. 0) lx=(-1)*lx
if(ly. lt. 0) ly=(-1)*ly
ld=2*ly-1x
incr1=2*ly
incr2=2*(ly-1x)

if (x1. lt. x2) go to 25
sx=x2
sy=y2
xend=x1
go to 30

25
sx=x1
sy=y1
xend=x2
irow=(-1)*sy+128
icol=sx+128
call ppnt(ibuf, irow, icol, ival)

30
if (sx. ge. xend) go to 50
sx=sx+1
if(ld. ge. 0) go to 40
ld=ld+incr1
go to 45

40
sy=sy+1
ld=ld+incr2

45
irow=(-1)*sy+128
icol=sx+128
call ppnt(ibuf, irow, icol, ival)
go to 35
150  if(dx.lt.dy) go to 500
    b=y1-m*x1
    do 160 i=x1,x2
       x=i
       y=m*x+b
       inty=y
       intx=x
       irow=(-1)*inty+128
       icol=intx+128
       call ppnt(ibuf,irow,icol,ival)
    160    continue
    go to 50
    x=x1
    do 200 i=y1,y2
       intx=x
       irow=(-1)*i+128
       icol=intx+128
       call ppnt(ibuf,irow,icol,ival)
       x=x+(1/m)
    200    continue
    go to 50
    300  if(y1.lt.y2) go to 350
        dm=y1
        y1=y2
        y2=dm
    350    do 250 i=y1,y2
           irow=(-1)*i+129
           icol=x1+128
           call ppnt(ibuf,irow,icol,ival)
    250    continue
    50    return
    end
This subroutine creates a subimage from an image

INAME : input file name
IDUTNAME : output file name
ISIZE : output file size
C, R : starting column and row in the input image

SUBROUTINE SUBIMG(INAME, IDUTNM, ISIZE, C, R)

PARAMETER NBUFSZ=300
PARAMETER NCOLSMAX=256
INTEGER IBUF2(NBUFSZ), IBUF(NBUFSZ)
INTEGER IARRAY(NCOLSMAX)
INTEGER C, R

PICBUF FOR INPUT FILE
CALL PICFMT(IBUF, INAME, IHDR)
CALL GFMT(IBUF, NROWS, NCOLS, NBITS, IMODE)
CALL MAKB(IBUF)
CALL PICIN(IBUF, INAME, IHDR)

DELETE OUTPUT FILE IF IT EXISTS
CALL DFILW(IDUTNM, IER)
CALL CHECK(IER)
IF(IER.NE.1) GO TO 15
CONTINUE
IFACTOR=256/ISIZE

PICBUF FOR OUTPUT FILE
CALL GFMT(IBUF, NROWS, NCOLS, NBITS, IMODE)
NROWS2=NROWS/IFACTOR
NCOLS2=NCOLS/IFACTOR
CALL PFMT(IBUF2, NROWS2, NCOLS2, NBITS, IMODE)
CALL MAKB(IBUF2)

ISTART=R
ISTOP=R+ISIZE-1
ICOL=C
IROW2=1
ICOL2=1

DO 10 IROW=ISTART, ISTOP
CALL GROW(IBUF, IROW, ICOL, ISIZE, IARRAY)
CALL PROW(IBUF2, IROW2, ICOL2, ISIZE, IARRAY)
IROW2=IROW2+1
 CONTINUE

CALL PICOUT(IBUF2, IDUTNM, IHDR)
CALL RELB(IBUF2)
CALL RELB(IBUF)
RETURN
END
This subroutine converts an integer file to a complex form.

ISIZE : size of the files
INNAME: integer file name
OUTNAME : complex (output) file name

SUBROUTINE SVTC(ISIZE, INNAME, OUTNAME)
INTEGER TMP(256), IN(1024)
INTEGER INNAME(7), OUTNAME(7)
COMPLEX OUT(1024)

CALL OPEN(8, INNAME, 1, IER)
CALL CHECK(IER)
CALL DFILW(OUTNAME, IER)
CALL CFILW(OUTNAME, 2, KER)
CALL CHECK(KER)
CALL OPEN(9, OUTNAME, 2, IER)
CALL CHECK(IER)

OPEN 9, OUTNAME, ATT="OR", ERR=100
CONTINUE
IFACTOR=256/ISIZE
ITIMES=64/IFACTOR**2-1

DO 30 I=0, ITIMES
CALL RDBLK(B, I, TMP, I, IER)
CALL CHECK(IER)
CALL UNP(256, TMP, IN)
DO 40 J=1, 1024
   OUT(J)=CMPLX(FLOAT(IN(J)), 0.0)
40 CONTINUE
CALL WRLBK(9, 16*I, OUT, 16, IER)
CALL CHECK(IER)

CONTINUE
CALL RESET
RETURN
END
OPTIONS:
1: Forward transform (Origin at CENTER)
2: Inverse transform (Origin at CENTER)
3: Forward transform (Origin at CORNER)
4: Inverse transform (Origin at CORNER)
5: Transform 2D complex file

SUBROUTINE SFFT2(INAME, OPTION)

COMPLEX X(512), Y(512)
INTEGER OPTION

ICHAN = 1
CALL OPEN(ICHAN, INAME, 2, IER)
IF(IER.EQ.1) GO TO 30
TYPE"in SFFT2. error-openinin file", IER
CALL CLOSE(ICHAN, IER)
GO TO 99

C COMPUTE 2D-FFT

IOPT = OPTION
CALL FFT2(ICHAN, X, Y, IOPT, IER)
IF(IER.EQ.1) TYPE"NORMAL COMPLETION"
IF(IER.EQ.4) TYPE"INVALID FILE SIZE"

RETURN
END

This subroutine will unpack four 4-bit integers from a 16-bit integer word. The pixels in a video file have to be unpacked if each pixel is to be operated on separately (ref 2)

SUBROUTINE UNP(N, PIXWORD, PIXELS)
INTEGER PIXWORD(N), PIXELS(4, N) , four pixels per word
DO 1 I=1, N
   'N' allows higher order arrays to be passed
   DO 1 J=1, 4
      PIXELS((5-J), I) = 15 AND PIXWORD(I) , pick off right pixel
      PIXWORD(I) = ISHFT(PIXWORD(I), -4) , shift word 4 bits right
      IF(MOD(N, 12) EQ 0) TYPE"UNP DONE .N . to pick off next pixel"
1 RETURN
END
Bibliography


Vita

First Lieutenant Murat Cincioglu was born on 10 August 1961, in Sungurlu, Corum, Turkey. He graduated from Sungurlu High School in Sungurlu in 1978, and attended Turkish Air Force Academy graduating in 1982 as a distinguished graduate with a Bachelor of Science in Electrical Engineering. He was then assigned to Turkish Armed Forces Staff Academy, Istanbul as a project officer on electronic communications systems until entering the School of Engineering, the United States Air Force Institute of Technology to work on his master's degree in Communications and RADAR in 1984.

Permanent Address:
Egitim Daire Baskanligi
Dis Kurslar Subesi
Bakanliklar/ANKARA-TURKEY
Title: Parameter Estimation for the Blind Restoration of Blurred Imagery

Dissertation Chairman: David A. Haig, Major, USAF
Instructor of Electrical Engineering

Approved for Public Release: 15W APR 1986
Abstract

Methods to estimate the optical transfer function (OTF) of the degradation and the noise-to-signal ratio in a degraded image are presented which use no prior information about the degradation and the noise. Two types of degradations are considered: uniform linear camera motion and out-of-focus camera with circular aperture. The OTF of the out-of-focus camera is estimated by inspection of the spectra of the degraded image and from average over the spectra of subsections of the degraded image. A line estimation method is described to estimate the degrading OTF of the linear camera motion in one direction. An additional method called the "logarithmic estimate" is also developed. Frequency and space domain estimation methods are developed for the noise-to-signal ratio for the case of additive gaussian noise. Inverse, Wiener, and power spectrum equalization filters based on these estimates are implemented on a digital computer to restore a variety of degraded images and examples of blind restoration of these degraded images are presented.
END
4-87
DTIC