7-1-1985

Image reconstruction using Wiener filtering and unsharp masking: a computer model

Jay Berman

IMAGE RECONSTRUCTION USING WIENER FILTERING AND UNSHARP MASKING: A COMPUTER MODEL

By

Jay H. Berman

B.S. North Carolina State University

(1979)

A thesis presented in partial fulfillment of the requirements for the degree of Master of Science in the School of Photographic Arts and Science in the College of Graphic Arts and Photography of the Rochester Institute of Technology

July, 1985

Signature of the Author Jay H. Berman

Center for Imaging Science

Accepted by Ronald Francis

Coordinator, M.S. Degree Program

COLLEGE OF GRAPHIC ARTS AND PHOTOGRAPHY

Rochester Institute of Technology

Rochester, New York

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Jay H. Berman
has been examined and approved by
the thesis committee as satisfactory
for the thesis requirements for the
Master of Science Degree

Edward Granger
____
Dr. Edward M. Granger, Thesis Advisor

Willem Brouwer
____
Dr. W. Brouwer

Joseph J. Charles
____
Mr. Joseph J. Charles

Aug 16, 1985
____
Date

THESIS RELEASE PERMISSION FORM

ROCHESTER INSTITUTE OF TECHNOLOGY

COLLEGE OF GRAPHIC ARTS AND PHOTOGRAPHY

Title of Thesis: IMAGE RECONSTRUCTION USING WIENER FILTERING AND UNSHARP MASKING: A COMPUTER MODEL

I, Jay H. Berman, hereby grant permission to the Wallace Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

15 August 1985

Jay H. Berman

IMAGE RECONSTRUCTION USING WIENER FILTERING AND UNSHARP MASKING: A COMPUTER MODEL

By

Jay H. Berman

Submitted to the Center for Imaging Science in partial fulfillment of the requirements for the Master of Science degree at the Rochester Institute of Technology.

ABSTRACT

Research was conducted to computer model and compare the image reconstruction obtainable using Wiener filtering and unsharp masking. Wiener filtering and unsharp masking are techniques used to improve image quality and interpretation. It was demonstrated that far greater image restoration is obtained by Wiener filter than by unsharp masking because unsharp masking, unlike Wiener filtering, enhanced image noise along with the edges. A user friendly computer model, that may be used as a tutorial aid for Image Science students, incorporating Fast Fourier Transform (FFT) techniques was designed. Graphics allow the user to follow each stage of the image processing.

ACKNOWLEDGEMENTS

Assistance from the following sources is greatly appreciated. Without the help of these dedicated professionals this work would not have been done.

Dr. Edward M. Granger of the Eastman Kodak Co. and Professor at the Rochester Institute of Technology, for acting as my thesis advisor and primary instructor in this exciting new field of Digital Image Processing;

The Eastman Kodak Co. for allowing me the use of their facilities and for sharing their expertise;

Dr. W. Brouwer for acting as a member of the thesis committee, keeping me on my toes, and providing helpful assistance; and

The United States Air Force Institute of Technology (contract no. f33600-76-a-7299) for paying the bills and sending me to Graduate School.

LIST OF FIGURES

INTRODUCTION

The purpose of digital image processing and enhancement is to improve picture quality or image interpretation. More specifically, it is used to remove noise, to enhance object's edges, and to highlight specified features. (1). For the photointerpreter, scientist, surgeon,and others who analyze images, it is important to extract as much useful information from an image as possible. Often much of this information is contained in the fine detail of the image which is hard to resolve because of noise and blurring, Wiener filtering and unsharp masking are two image processing techniques which have been used to help the user get at this information and/or improve picture quality.

As is implied by the name Digital Image Processing, images are processed by computers using discrete values obtained by sampling an actual image. However, real images are continuous. Sampling implies that not every member of a population (in the case of film, not every silver crystal) is measured. Therefore, how an image is sampled is critical to the study of Digital Image Processing. Discrete values taken at points of equal spacing and proper rate insure a true

simulation of the image in the computer. This type of sampling is mathematically represented by the Comb function shown in figure 1.

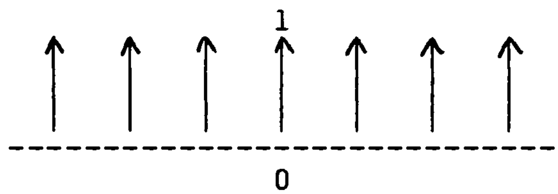


Figure 1. The Comb Function.

$$\mathrm{Comb}(x) = \sum_{n=-\infty}^{n=\infty} d(x-n) \qquad \text{(Eq.1)}$$

Thus the comb function is a series of unit area, equally spaced delta functions, d(x-n), that sifts out a series of single discrete values from the function. (2).

For a band limited signal, proper sampling insures a true and accurate simulation of the image will be made. The most important rule is that of sampling at a rate equal to or smaller than the interval ΔX calculated by the Nyquist criteria. The Nyquist criteria requires that the image be sampled at a rate ΔX such that the highest frequency of interest Fn is given by:

$$Fn = 1/(2*\Delta X) \qquad \text{Therefore: } \Delta X = 1/2Fn \qquad \text{(Eq.2,3)}$$

In any case where the signal is band limited within the region of bandwidth Fn and $\Delta X = 1/2Fn$ the function may be reconstructed exactly.

Incorrect sampling will cause aliasing, the overlapping of displaced, adjacent spectra, making exact image recovery impossible. (3) For this reason, when processing images using FFTs it is important to band limit and sample correctly. For example, given a function with N=256 points and a Nyquist frequency = 128.0 lines/mm, what sampling interval (ΔX) should be used? Figure 2 and the following mathematics provide a good guideline:

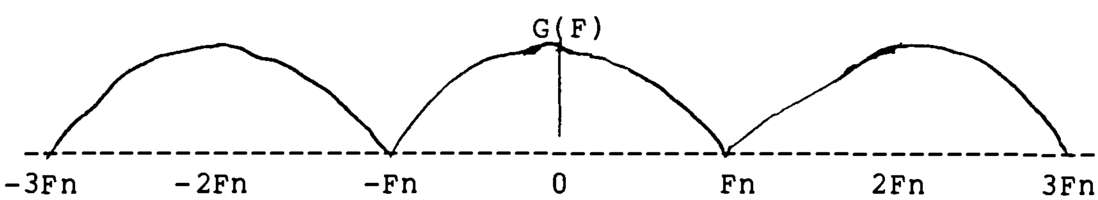


Figure 2. Illustration of the Nyquist frequency.

$$Fn = 1/(2*\Delta X) \text{ lines/mm} \qquad \text{(Eq.4)}$$

$$\text{Thus: } \Delta X = 1/(2* Fn) = 1/(2*128) = 1/256 \text{ mm} \qquad \text{(Eq.5)}$$

$$\text{Also: } \Delta F = Ft/256 = 256/256 = 1 \text{ line/mm} \qquad \text{(Eq.6)}$$

Figure 2 depicts the Discrete Fourier Transform G(F). The function is band limited from -Fn to Fn in accordance with the Nyquist criteria. The total frequency range, Ft, is from -Fn to Fn with N number of discrete points separated by F. However, because of sampling, Discrete Fourier Transforms are periodic. The periodic nature of these functions prevents the loss of information in performing the inverse transformation always associated with sharp cutoffs in the frequency domain.

The importance of sampling can not be stressed enough. Using the simple concepts just presented will insure the best reconstruction of an image. Just as important as sampling, an understanding of how light behaves and how optical systems respond to produce an image is primary to determining the type of image processing used. A short tutorial of these factors is presented here.

The irreducible object element is the mathematical point. A fundamental characteristic of any photographic material involves the way which a point or a line (an assemblage of points) is imaged within the material. Due to the nature of photographic emulsions, lenses, and other elements in an optical system, the irradiance energy falling on a film from a point or line is to some degree diffused and

spread within the emulsion. (4). For an image of a point or a line, this distribution of the irradiance by the emulsion is known as its point spread function, (p(x,y)), and line spread (l(x)) function respectively. Figures 3 and 4 are plots of the point and line spread functions.

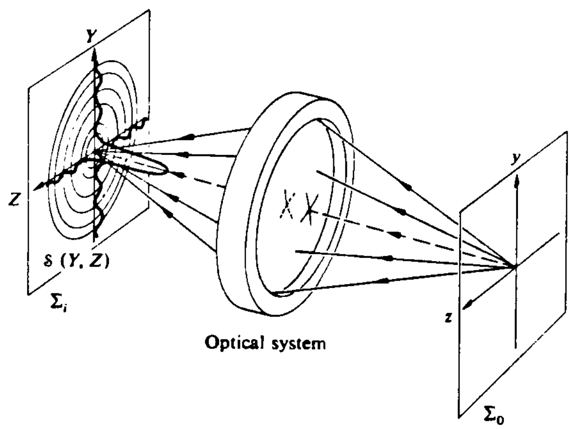


Figure 3. The Point Spread Function.

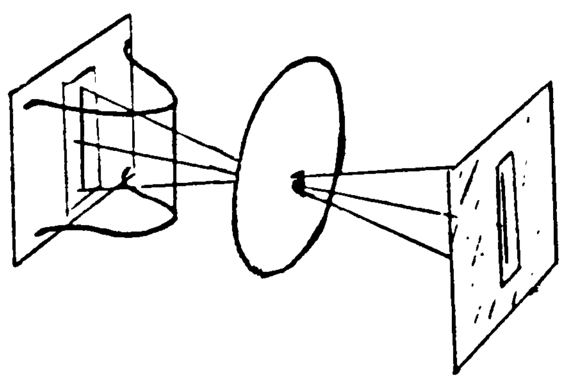


Figure 4. The Line Spread Function.

The Fourier transform of l(x) is (L(f)), commonly known as the Optical Transfer Function (OTF). The modulus of the OTF is known as the Modulation Transfer Function (MTF). The OTF and the MTF are presently the most widely used criteria for evaluating optical systems. The following mathematics illustrate the relationships of the functions just discussed: (5,6)

$$l(x) = \int_{-\infty}^{\infty} p(x,y) \quad dy \qquad \text{(Eq.7)}$$

$$L(f) = \int_{-\infty}^{\infty} l(x) \; e^{-i2\pi fx} \quad dx \quad = \quad OTF(f) \qquad \text{(Eq.8)}$$

$$MTF(f) = \left| OTF(f) \right| \qquad \text{(Eq.9)}$$

NOTE: Throughout the rest of this paper, $\Leftrightarrow$ will represent the Fourier Transformation. Therefore, $l(x) \Leftrightarrow L(f)$.

Another limitation associated with all films is graininess. Due to the discrete nature of film grains and because it is impossible to to make a truly monodispersed emulsion (an emulsion in which all of the imaging particles are the same size and are evenly dispersed), noise results in

the image. Noise is a general term used to refer to any undesired signal recorded on a recording medium or picked up by an instrument designed to respond to a certain signal; for example, static on a television or a radio. Photosensitive grains or small integral areas on an emulsion receiving equal exposures do not have equal densities when developed. These fluctuations in density are commonly referred to as grain and produce image noise. (7).

Image noise (G) on a film is defined as the product of the measured mean-squared density fluctuations $\sigma(a)**2$ and the area of the scanning aperture (A) used to make the measurement. (8).

$$G = A \ \sigma(a)**2 \qquad \text{(Eq.10)}$$

Image noise is related to granularity by the following:

$$S = \sigma(a)\sqrt{2A} \qquad \text{(Eq.11)}$$

Thus: $$S**2 = (\sqrt{2A}\ \sigma(a))**2 = 2G \qquad \text{(Eq.12)}$$

Where ** indicates exponentiation, S is the Selwyn granularity coefficient, and $\sigma(a)$ is the root-mean-square (rms) fluctuation in the measured density. (9,10).

The combination of the optics and film, and the grain of the film usually degrade the object being photographed. These undesirable effects are easily seen in aerial reconnaissance where the objects of interest often appear as very blurred and grainy images on the film. To improve the image, or in this case, make it more like the original scene, enhancements filters are used to reduce image noise and to sharpen faint or barely recognizable details from blurred edges. This report will compare two enhancement techniques, Wiener filtering and unsharp masking.

The Wiener filter is a mathematical algorithm designed to enhance images by removing noise and sharpening edges. Its design is governed by a prior knowledge of the noise in the system. The following is the mathematical description of the Wiener filter (11,12)

$$H(f) = L^{*}(f) \,/\, \left[L(f)^{2} + \left(N^{2}(f) \,/\, O^{2}(f) \right) \right] \qquad \text{(Eq.13)}$$

Where $L^{*}(f)$ denotes the complex conjugate of $L(f)$, normalized to $L(0)=1$, and $N^{2}(f)$ and $O^{2}(f)$ are the noise and input signal power spectra.

By examining equation 13 it is easy to see how the Wiener filter works. When the noise power greatly exceeds the input signal power, which happens at very high frequencies, $H(f) \rightarrow 0$; the filter allows very little signal to pass through it. If the signal is much greater than noise, as happens to be the case for the lower frequencies, $H(f) \rightarrow L^{*}(f)/L(f)^{2} = 1/L(f)$ thus allowing most signals to pass through the filter. For noiseless conditions the Wiener filter becomes an inverse filter. For signal/noise ratios in between these extremes the Wiener filter responds accordingly. (13,14). Even though some of the noise will be passed by the filter and some of the signal lost, the Wiener filter is considered the optimal restoration filter in a least squares sense and the standard to which other filtering techniques are compared to because it minimizes the mean-square differences between the input and the restored target. (15,16).

The noise power (Wiener) spectrum of an image is the Fourier decomposition of the noise. It can be used in the frequency domain as a statistical evaluation of image noise. (17,18,19,20). Once an estimate of the noise Wiener spectrum has been calculated, the optimal Wiener filter can be

designed. The several steps which must be taken to estimate the Wiener spectrum are outlined below:

First calculate the mean density or signal in the system and subtract this mean value from each data point. These values fluctuating about the mean are the noise. The next step is to perform the technique of noise windowing. To better describe this technique the triangular or Bartlet window is described below. Figure 5 illustrates the technique using the Bartlet window (21).

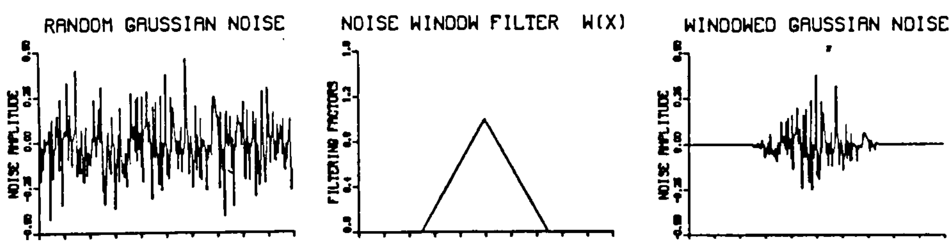


Figure 5: The Bartlet Window.

$$w(x) = \begin{cases} 1 - |2x/C| , & |x| \leq C/2 \\ 0 \quad \text{elsewhere} & \end{cases} \qquad \text{(Eq.14)}$$

$$w(x) \Leftrightarrow W(f) = \frac{C}{2}\left[\frac{\sin(Cf/2)}{f/2}\right]^2 = \frac{C}{2}\,\mathrm{sinc}^2\,(Cf/2) \qquad \text{(Eq.15)}$$

The windowing is performed by multiplying the noise data (n(x)) with the window w(x). As is apparent the window acts to limit the extent of the noise, and it also eliminates end transitions that cause Wiener spectrum estimate error that could occur when performing the third step, FFT the windowed noise. These three steps yield the following advantages:

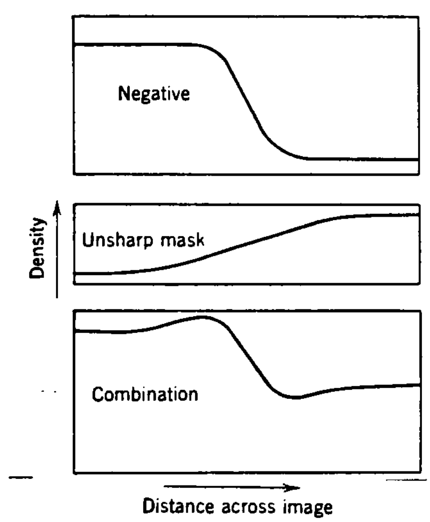
1) Eliminates discontinuities at the end of data files.
2) Prevents wrap around when estimating Wiener Spectrum by using an FFT algorithm.
3) Smooths the power spectrum. (Note: Multiplication in the space domain is a convolution in the frequency domain.) (22).

Once the windowed noise is found, it is transformed to the frequency domain. This is used in the final step. The final step for calculating the Wiener spectrum is to calculate the squared modulus of the transform of the windowed noise.

The Wiener spectrum just calculated is used in Eq.13 to produce the optimal Wiener filter. The noise reducing characteristic of the Wiener filter is not present in the unsharp masking technique. As previously indicated this is the second filter that will be examined.

The unsharp masking technique aids in the retention of detail and improves the visual appearance of an image by sharpening the edges between distinct objects in the image by exaggerating the density difference across the edge. (23). Yule (24,25) outlines the procedure for implementing the unsharp mask technique. An unsharp mask of an original scene is produced on a moderately low contrast film (contrast compression). The mask image is opposite in sign from the original. This may be achieved by contact printing the original and the mask with a sheet of diffusion material between them or by defocusing if using a projection system. The original and the mask are combined in register and used to produce a final image on a higher contrast material; the higher contrast material compensates for the contrast reduction resulting from the combination of the original and the mask.

The unsharp mask affects the final image by reducing the large scale tonal differences or density ranges of the low frequencies. The fine details are blurred by the unsharp mask. When the final image is made using the original image and mask the contrast differences of the fine details are maintained because they are not drastically effected by the subtraction. This results in an increased contrast difference along an edge gradient (improved sharpness) when the original contrast is restored. (26). Figure 6 is an illustration of the unsharp masking technique. Note the increased density range due to the hump and dip along the edge gradient resulting from the unsharp mask.



Action of unsharp mask in improving edge sharpness.

Figure 6. The Unsharp Masking Technique from Yule. (27).

The unsharp mask combined with the degraded image may be regarded as a filter in the frequency domain with its OTF increasing with spatial frequency. This results in an increase in output modulation and MTF values greater than unity, the phenomenon of contrast enhancement (28,29). Figure 7 is an illustration of contrast enhancement from a filter with a negative spread function.

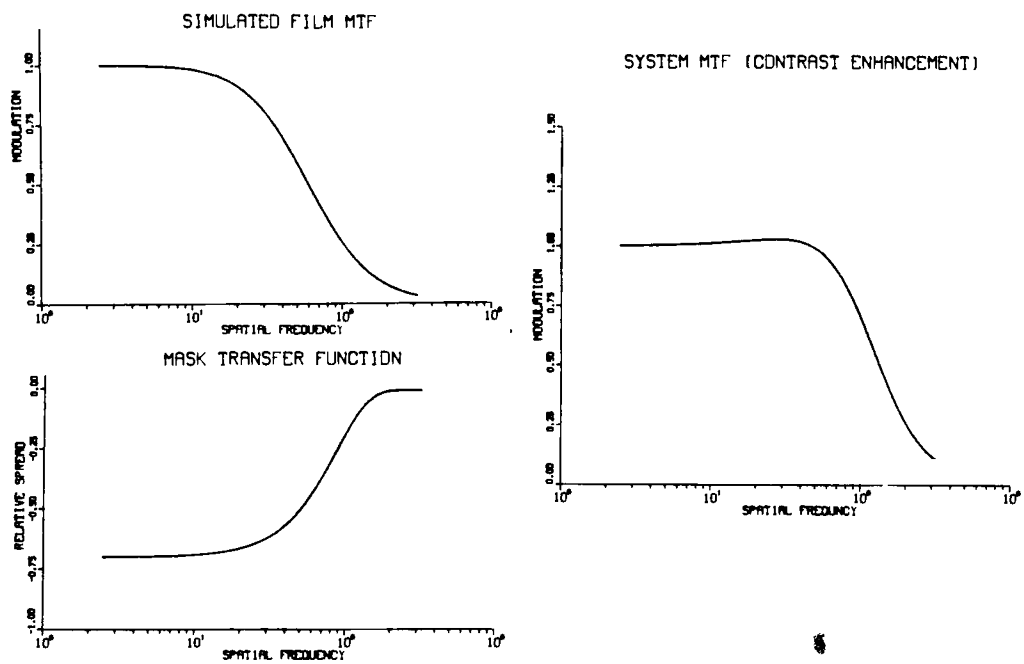


Figure 7. An Illustration of Contrast Enhancement.

From figure 7, observe that the unsharp mask, used here in the technique of contrast enhancement acts as a restoration filter and in its theoretical ideal case approximates an inverse filter. Two questions arise from this illustration: 1) How do we determine the line spread function used to create the desired unsharp mask? 2) Once the unsharp masking technique is modeled, how can it be described it in terms of modulation transfer theory? The following discussion will answer these question.

To create the desired unsharp mask and achieve the best object restoration through contrast enhancement, it is necessary to look at the mathematics involved. The following mathematics, utilizes Fourier techniques in the frequency domain, were developed by E.M. Granger (30,31). This work follows a line of thought similar to Armitage, Lohmann, and Herrick (32) with the added advantages of yielding an optimum restored image in a least squared sense, and it is not subject to the contrast limitations specified in the Armitage, Lohmann, and Herrick paper.

The following symbolism will be used:

O(F)	=	OBJECT	B1	=	CONSTANT
MTF(F)	=	FILM1	B2	=	CONSTANT
I(F)	=	IMAGE	G1	=	FILM GAMMA IMAGE
U(F)	=	FILM2	G2	=	FILM GAMMA MASK
MASK(F)	=	MASK	G3	=	FILM GAMMA FINAL
M2	=	2ND MOMENT	PI	=	3.14159
RO(F)	=	RESTORED OBJECT			

$$O(F) * MTF(F) = I(F) \qquad I(F) * U(F) = MASK(F) \qquad \text{(Eq.16,17)}$$

Solving simultaneously;

$$MASK(F) = O(F) * MTF(F) * U(F) \qquad \text{(Eq.18)}$$

Note that both MASK(F) and U(F) are unknown. Therefore, to solve for MASK(F), U(F) must be written in a non-variable form. A way to do this is extracted from the Second Moment Technique developed by E.M. Granger (33). This technique provides a means of estimating the MTF of a system/element in a system by the statistical second moment of the l(x). It is used here to determine the l(x) necessary to produce the unsharp mask. He found the following relationship exists:

$$MTF = 1 - (2pi^2 F^2 M2) \qquad \text{M2 is the Second Moment.} \qquad \text{(Eq.19)}$$

For the complete derivation see the reference.

Substituting the Second Moment Approximation for MTF(F) and U(F) into equations 16 and 17:

$$I(F)=O(F)*(1-(2pi^2*M2*F^2)) \quad \text{and} \qquad \text{(Eq.20)}$$

$$MASK(F)=I(F)*(1-(2pi^2*M2*F^2)) \qquad \text{(Eq.21)}$$

Therefore:

$$I(F)=O(F)*(1-(B1 * F^2)) \quad \text{where } B1=2pi^2*M2 \quad \text{and} \qquad \text{(Eq.22)}$$

$$MASK(F)=I(F)*(1-(B2 * F^2)) \quad \text{where } B2=2pi^2*M2 \qquad \text{(Eq.23)}$$

From Eq.18 or by combining Eq.22,23:

$$MASK(F) = O(F) * (1-(B1 * F^2)) * (1-(B2 * F^2)) \qquad \text{(Eq.24)}$$

$$MASK(F) = O(F) * (1-(B1+B2) * F^2) + \underbrace{(B1*B2 * F^4)}_{H.O.T.} \to 0 \qquad \text{(Eq.25)}$$

The next step in restoring the original object is to combine the image and the mask in register and exposed them onto a higher gamma film. The following mathematics illustrate this.

$$RO(F) = (I(F) + MASK(F)) \quad \text{(Eq.26)}$$

Substitute Eq.22,25 into 26. Include film gamma constants.

$$RO(F) = G3 * ((G1*O(F)*(1-B1*F^{2})) + (G2*O(F)*(1-(B1+B2)*F^{2})) \quad \text{(Eq.27)}$$

$$= G3((G1*O(F)-(G1*O(F)*B1*F^{2}) + (G2*O(F)-(G2*O(F)*B1*F^{2}) - (G2*O(F)*B2*F^{2}) \quad \text{(Eq.28)}$$

$$RO(F) = G3*((G1+G2)*O(F) - (G1+G2)*O(F)*B1*F^{2} - G2*O(F)*B2*F^{2}) \quad \text{(Eq.29)}$$

$$= G3*\left((G1+G2)*O(F)-O(F)*\left((G1+G2)*B1*F^{2}+G2*B2*F^{2} \right)\right) \quad \text{(Eq.30)}$$

$$RO(F) = G3 * O(F) \left((G1+G2) - ((G1+G2)*B1 + G2*B2) *F^{2} \right) \quad \text{(Eq.31)}$$

To obtain optimum enhancement set (G1+G2) * B1 + G2 * B2 equal to zero and thus eliminate the F^2 dependence.

Therefore: $$G1*B1_{\text{image}} = -G2*(B1+B2)_{\text{mask}} \quad \text{(Eq.32)}$$

$$-G1/G2 = (B1+B2)/B1 \quad \text{(Eq.33)}$$

Observe that the mask and image are opposite in contrast.

To restore original contrast chose values so that:

$$G3 * (G1+G2) = 1.0 \qquad \text{(Eq.34)}$$

The derivation above allows for direct modeling of the unsharp mask. To describe the unsharp masking technique as a linear filter in terms of modulation transfer theory the operation involved must be approximately linear. The mathematical linear approximation for masking, developed by Armitage, Lohmann, and Herrick (34) is:

$$Tr(s) = G1 - G2(m)*Tr(m) \qquad \text{(Eq.35)}$$

where Tr(s) is the system MTF (MTF of the final image), G2(m) is the mask contrast, and Tr(m) is the MTF of the mask. Incorporating the results of the Granger method into their approach allows the unsharp masking technique to be expressed in terms of a linear filter, free of small contrast limitations, and allows the final MTF of the masking technique to be easily calculated. For the complete mathematical derivation consult their paper. From the

formula it is apparent that Tr(m) must be negative to yield a final MTF of 1.0.

As previously illustrated (Figure 6), unsharp masking enhances edges by adding the original image with a blurred negative of the original in register. Therefore, any spread function and or film contrast combination may be chosen to create the mask. However, this may not optimum if the combination is not chosen according to the rules provided in equations 16-34. Other contrast values and spread function combinations will result in varied degrees of edge enhancement.

According to Scarff, (35) unsharp masking results in more visually pleasing photographs if the images are not made to sharp. In his experiments he learned that a final image with a maximum MTF of 1.3 produced from a scene with an input modulation of .70 was visually preferred over the same an image of the same scene produced at a higher or lower MTF. However, he does not draw any conclusions about images produced from scenes with other input modulations or the effects of S/N on image quality.

OBJECTIVES

It is the objective of this research to determine to what degree of image restoration is possible, what are the limitations to image enhancement/restoration using these two techniques, and how do these filters differ. This paper will quantify, analyze, and compare the digital image processing techniques of Wiener filtering and unsharp masking. To study these methods, noise and image components are cascaded, in the frequency domain using Fast Fourier Transforms, through a series of mathematical filters and transfer functions. The model allows the user to examine each stage of the processing. It is expected that a greater degree of image reconstruction (restoration) is obtainable with the Wiener filter than with the unsharp mask.

Another objective of this work is to design a user oriented computer model that will aid students in learning and understanding the use of Fourier Transform Techniques for one dimensional image processing and analysis.

EXPERIMENTAL

The experimental part of this research consisted of writing an image processing computer model and an analysis of data obtained from the model. The experimental model was designed in FORTRAN on an IBM/CMS computer system. The model can easily be incorporated into any system that supports FORTRAN 77 and the IMSL Library. The IMSL Library is a set of statistical and scientific FORTRAN programs available on many main frame computers. All graphics generated by this model were made using the 'DISSPLA' software package. 'DISSPLA' is a product of the Integrated Software Systems Corporation, San Diego, California 92121. The IBM system was used because of availability, speed, and the vast amount of scientific software supporting it.

The image processing model used has a modular design to provide flexibility, ease for the user, and a tool for learning. The modular design allows the user to choose predefined elements or substitute a simulation into the process. Various combinations were used and analyzed by this researcher with the major goal to obtain optimal reconstruction from the various inputted specifications. Figure 8 is an illustration of the image restoration process.

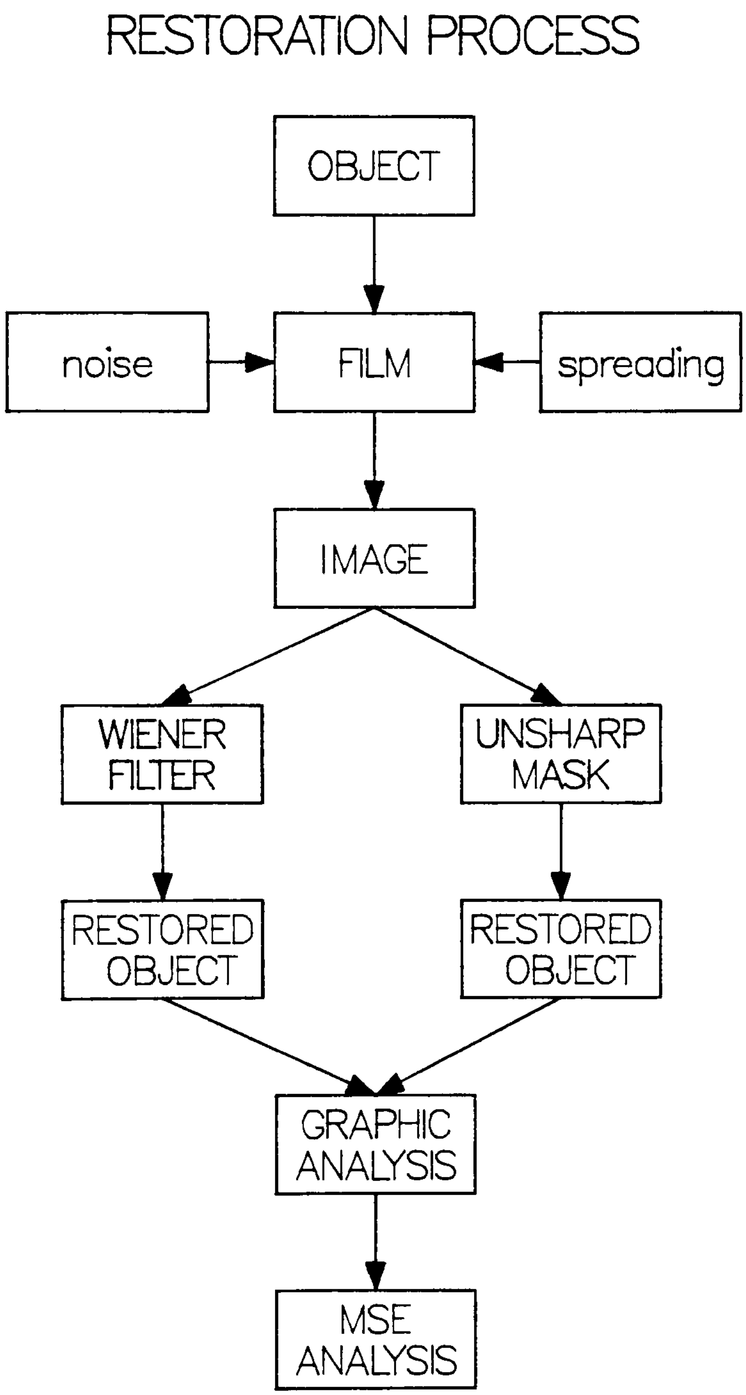


Figure 8. An Image Restoration Process.

Each process illustrated here is software supported. Documentation for and a copy of each experimental program used is provided in the Appendix.

Discrete FFT techniques were incorporated into the modular design. It is faster, easier, and less expensive to do this digital image processing in the frequency domain by cascading (multiplying) frequency components than in the spatial domain via convolution. A 256 point FFT was chosen to reduce processing time and because of limited computer memory allocation. Using 256 points imposes a limitation on the model; the transform of a bounded function in the spatial domain is theoretically unbounded in the frequency domain. Cutting off a function results in the loss of information at the higher frequencies beyond the cutoff frequency. However, as discussed in the Introduction, the Discrete FFT prevents the loss of information because of its periodic nature; the function in the spatial domain is exactly recovered by inverse transforming. Despite these limitations both frequency and spatial domains can be well represented with 256 points. For the 256 points, a frequency range of -160 to 160 cycles per millimeter (cyc/mm) was chosen because many films can be modeled well over this range. Using the mathematics discussed in the Introduction on sampling theory,

we calculate ΔF=1.25 cyc/mm, ΔX= 1/320 mm, and the spatial range from -.40mm to .40mm.

As illustrated on figure 8, the first step in the restoration process is to simulate an object. A 5cyc/mm tribar target was chosen for this work. Using this simple target for the object has the following advantages:

1) It is a widely used standard in image processing and evaluation.

2) Each cycle has 64 data points for analysis.

3) It is easy to see and examine the degradation and restoration graphically as the image passes through the system.

4) Tribar targets are easy to analyze mathematically.

The next step in the model is to degrade the object using a simulated film model. This is done by blurring the object and adding random gaussian noise to it, thus yielding a simulated film image. The object was blurred by multiplying it in the frequency domain with an MTF curve generated using the widely accepted Frieser model. The Frieser model is defined as:

$$MTF = 1.0 / (1.0 + (\alpha * f^2)) \qquad \text{(Eq.41)}$$

where f is frequency and α is the parameter controlling the width of the curve. Note that the MTF for the Frieser model does not go to 0. For this case, the periodic nature of the Discrete FFT is especially important in preventing the loss of information when performing digital image processing. To simulate the random density fluctuations found in exposed film, random numbers from a gaussian distribution with a mean of zero and a user defined variance are then added to the blurred image.

As shown in figure 8, the next step in the restoration process is to reconstruct the tribar object via the Wiener Filtering and Unsharp Masking techniques. This step in the simulation is the thrust of this research. The results of these two techniques will be analyzed, compared, and quantified in the results section of this thesis. The mathematics involved in these restorations have been thoroughly described in the Introduction and are used in the computer model.

The final step in the computer simulation is the generation of data and graphics from these two restoration methods. Two types of analysis will be performed; a graphics analysis which allows us to see a graphical representation of

every phase of the restoration process for the two filters and a mathematical analysis of the mean-square error of the restored object compared to the original. The graphic simulations generated by the model permits a visual comparison of the two techniques. The data collected by the mathematical technique is used to generate graphs showing the degree of image restoration and behavior of the filters for various levels of degradation. The data and graphs collected will be analyzed in the results section of this thesis.

RESULTS

This research has found the Wiener filtering technique to be superior to the unsharp masking technique for image restoration under all test conditions within the limits of the image processing model developed for this work. The results that will be presented in this section will show this quantitatively and graphically. Before any data could be collected, a digital image processing model to simulate the unsharp masking and Wiener filtering techniques was developed. The model allows the user to examine each stage of the simulation process graphically.

Figures 9 and 10 shows the simulation developed for this research. Illustrated here are some of the major imaging elements and processing stages simulated by the model. Figure 9 (from top left) simulates a 5cyc/mm tribar object being blurred by a hypothetical film with an MTF cutoff of 100cyc/mm under noiseless conditions to yield an image. As is easily seen this results in an image of the object that is blurred and rounded off particularly along the edges. Figure 10 shows the object restoration attainable with the Wiener filter and the unsharp mask.

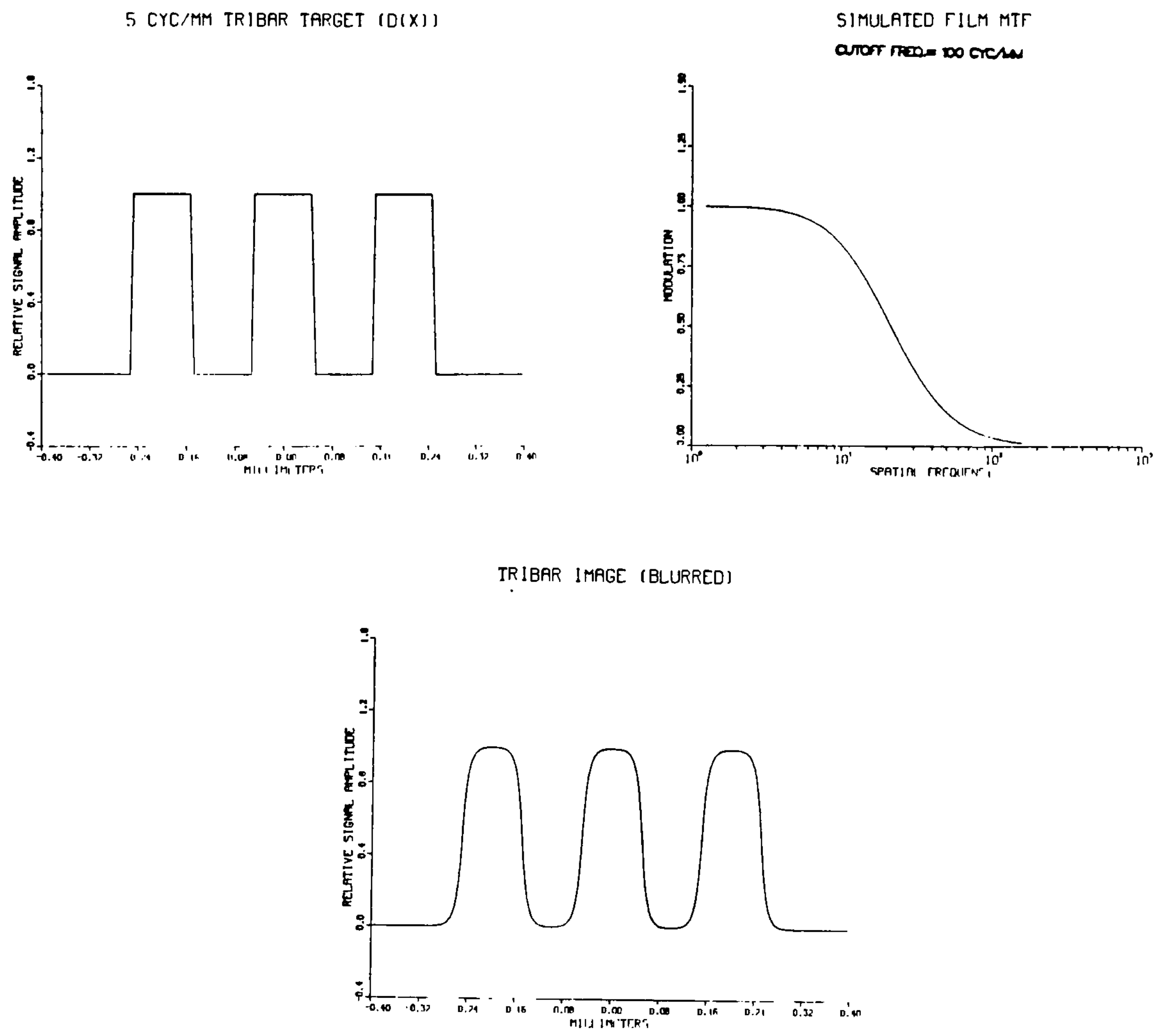


Figure 9. Simulating an Imaging System.

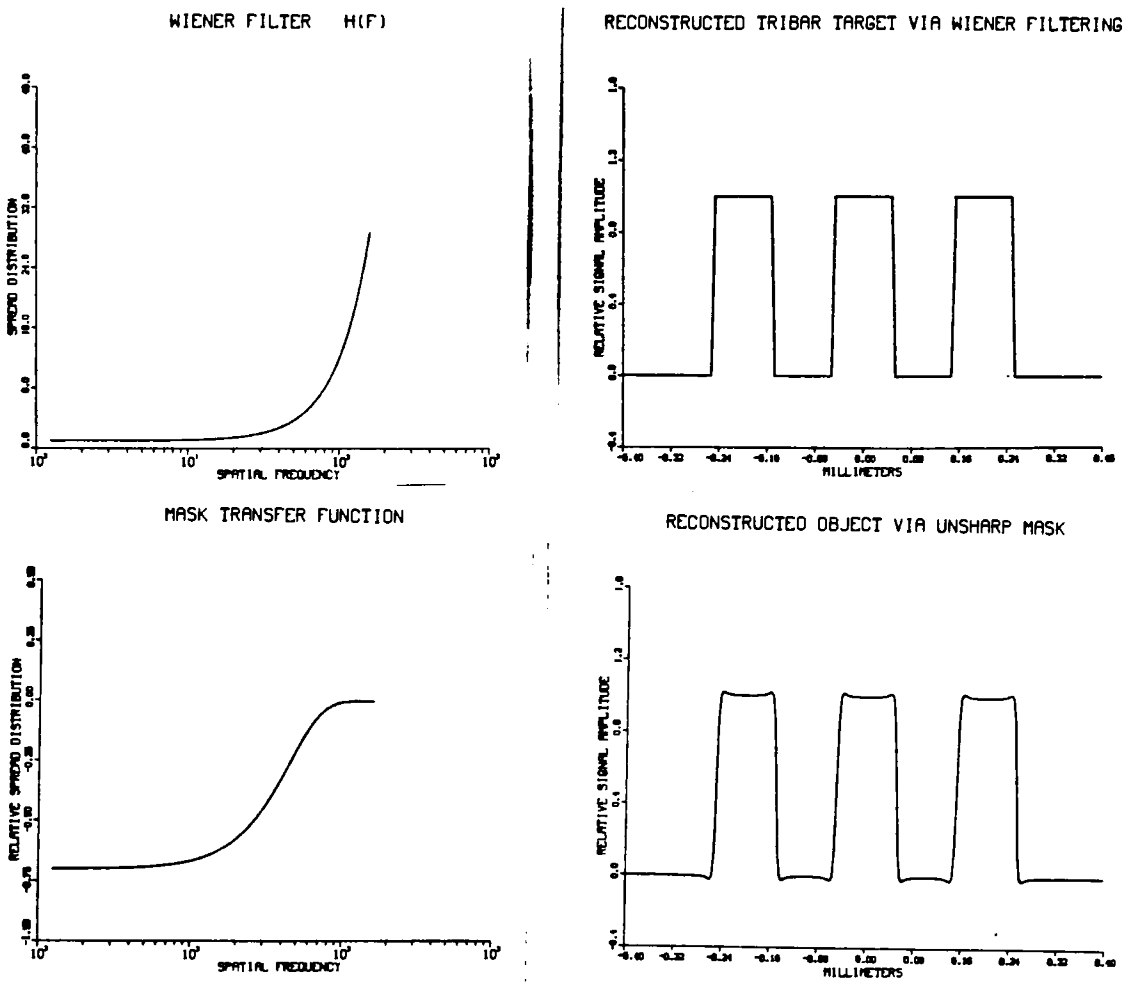


Figure 10. Simulating a Restoration System.

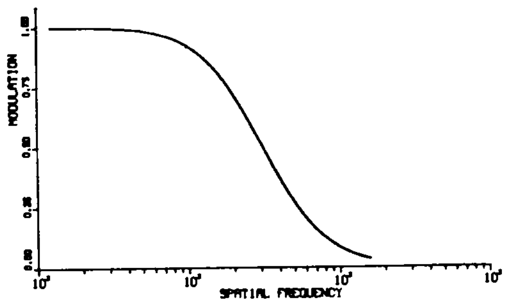
It was mathematically shown in the introduction that under theoretically noiseless conditions the Wiener filter was an inverse filter. Using an inverse filter to restore a noiseless image yields a perfect reconstruction of the object if the simulation does not suffer from the effects of sharp cutoffs. Because the Freiser model does not go to 0, all 256 points are used in both the spatial and frequency domains, and due to the periodic nature of Discrete FFTs, the simulation created for this research does not exhibit sharp cutoffs. The graph of the restored object demonstrates this perfect restoration using Wiener filtering. Also shown in the Introduction, figure 6, was that a smooth image combined with a smooth, blurred mask resulted in a sharper image with exaggerated edges. These results, shown in figure 10, for a noiseless system agree with the results expected from theory.

As was stated in the objectives, the data collected will be use to quantify, analyze, and compare the two techniques. Data collected separately allows each method to be quantified and analyzed individually. The information, results, and data gathered separately will be included with data obtained collectively to permit a thorough comparison of the two methods.

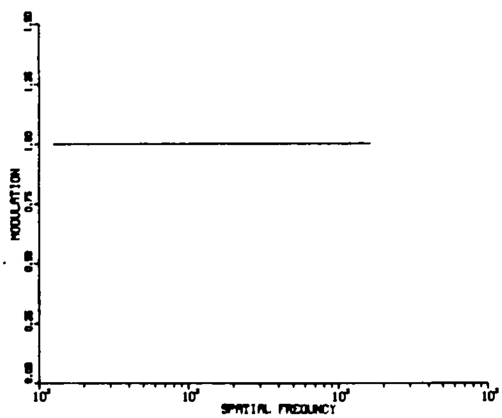
The Wiener filter will be looked at first. Note in figure 11, since the film does not go to 0, neither will the Wiener filter or the final MTF. The Wiener filter yields perfect restoration of the object in the absence of noise regardless of the spread function degrading the image. This is possible because the Wiener filter becomes the inverse of the film spread function so the effective system MTF over the entire frequency range simulated is 1.0, an ideal system. This is illustrated by the following equation and graph in figure 11:

$$H(f) = 1/L(f) \qquad\qquad 1/L(f) * L(f) = 1.0$$

SIMULATED FILM MTF



SYSTEM MTF (FILMMTF * WIENER FILTER)



WIENER FILTER H(F)

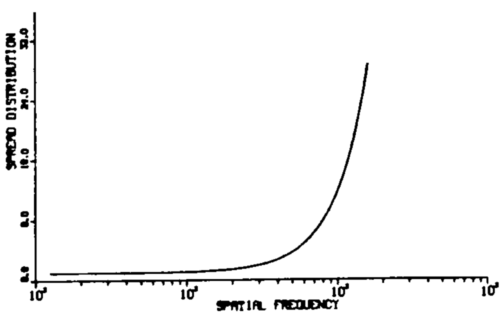


Figure 11. An Ideal System.

Figure 12 show the reconstructions of a 5cyc/mm object (left side) and a 10cyc/mm object (right side) blurred by a film with a cutoff frequency of 150cyc/mm using Wiener Filtering. The 10cyc/mm restored object is 2X scaled to provide ease in comparing the two restorations. Note perfect restoration in both cases is achieved even though the 10cyc/mm image is more degraded. Also note, that the filter for the more degraded image rises much faster.

In the presence of noise the Wiener filter responds differently. As indicated in the introduction, the filter seeks a compromise between noise removal and edge sharpening dependent on the degree of the degradation due to each factor. To determine the filter's overall response to varying noise a series of restorations runs were performed and analyzed. Figure 13 illustrates two of these runs. Shown are the images of a 5cyc/mm target degraded by a film with a cutoff frequency of 150cyc/mm Alpha noise level of .01 (left side, low noise) and .15 (right side, high noise). For the case of the low noise, high S/N, note the edges are enhanced but at the expense of increasing the noise. For the high noise case, low S/N, much of the noise is removed but little is done to sharpen edges. Also note, the filter for the lower noise case rise three times faster.

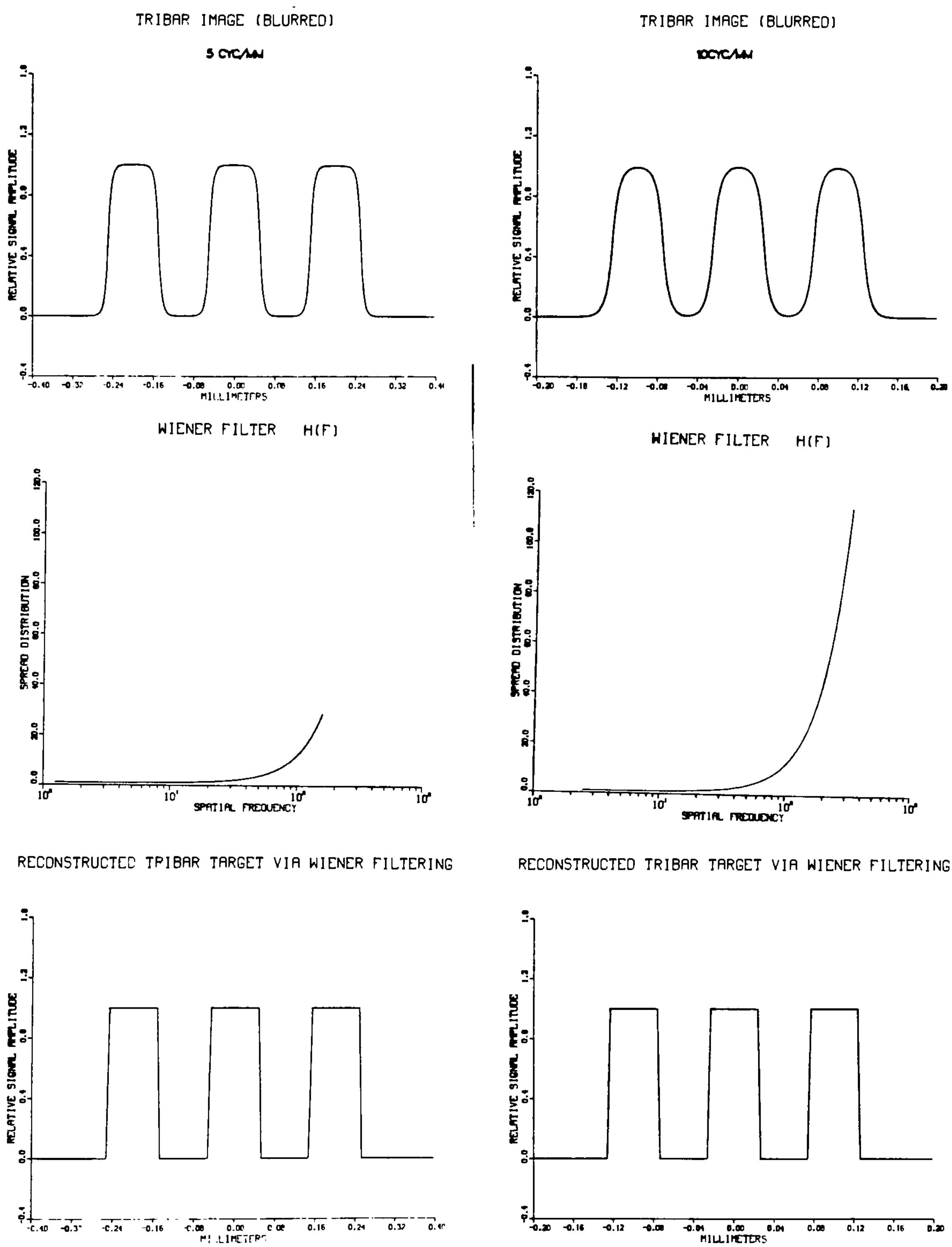


Figure 12. Restoring a Noiseless Object. Wiener filter

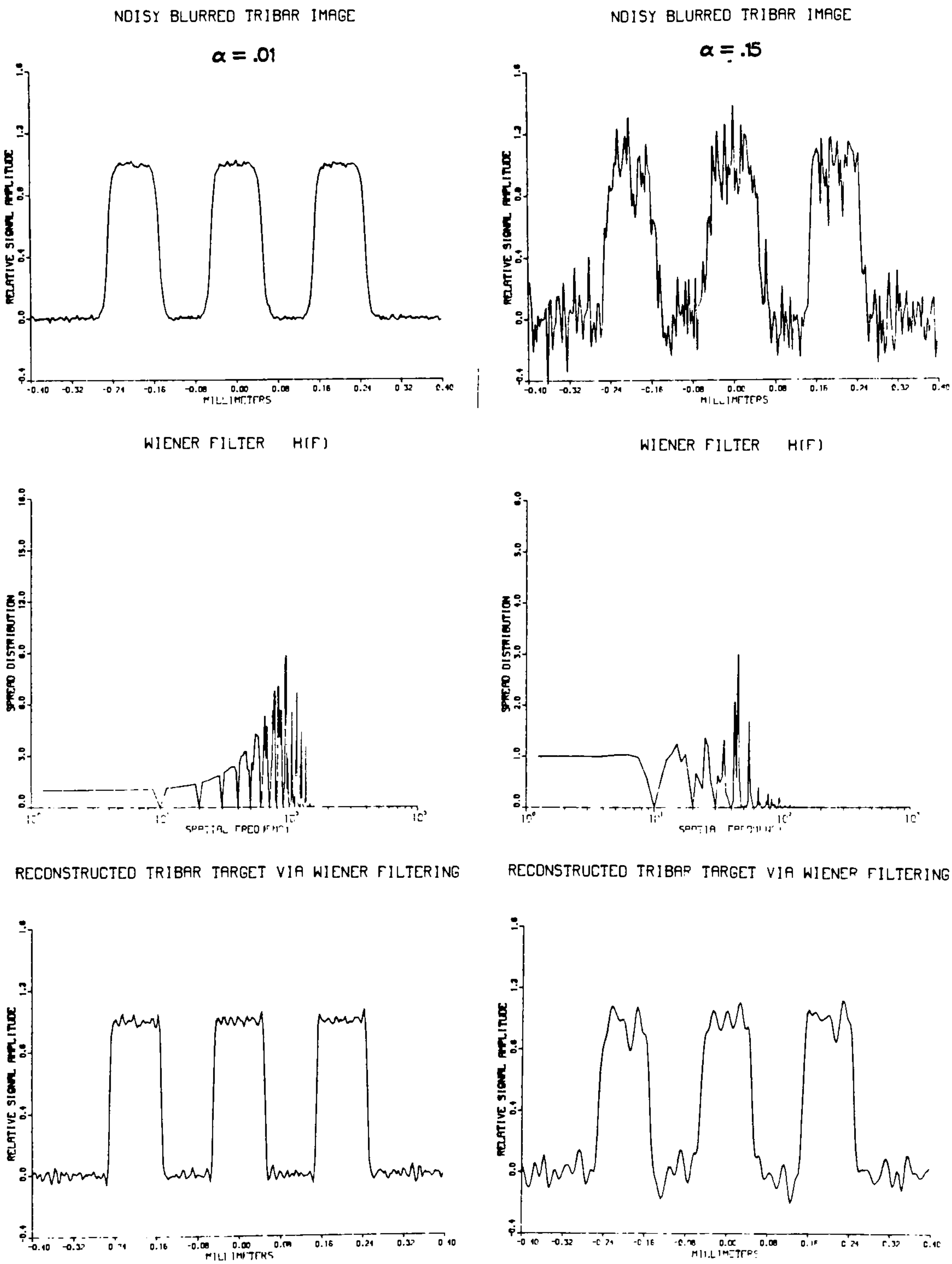


Figure 13. Restoring a Noisy Object. Wiener filter

This brought up an interesting question: At what point does the filter respond more like a noise removal filter and less like an inverse (edge enhancement) filter? figure 14 answers this question. It shows one result of this series of restoration runs. It is a graph showing S/N (in) vs S/N (out) of the system. It shows for S/N(in) less than 40:1 the filter reduces noise and for S/N(in) greater than 40:1 it will sacrifice (increase) the noise in favor of enhancing the edges. The data on figure 14 appears to fit a Y=aX**b curve. If this is so, it will plot as a straight line on a log-log graph and the Wiener filter's performance may be qualified. The fit is good statistically. Using a T test, it falls well within the 95% confidence limits for S/N(in) ratios less than 250/1.

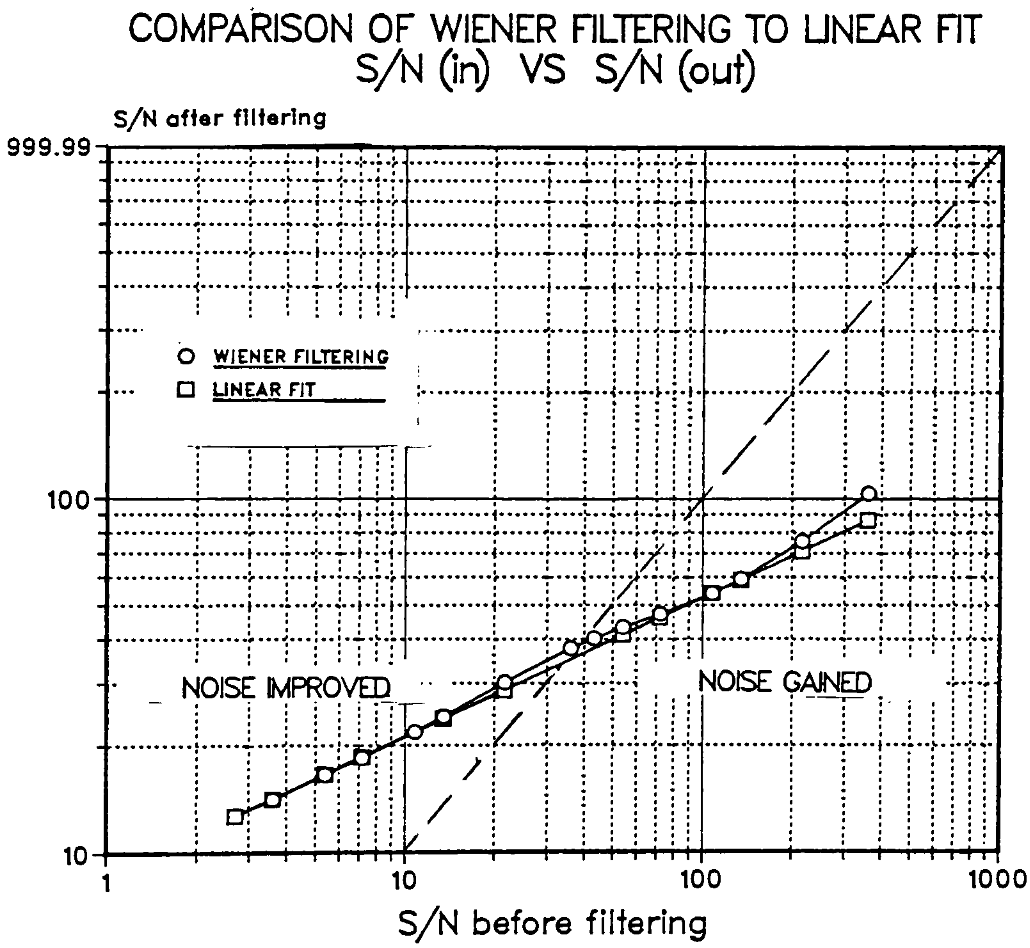


Figure 14. Linear Fit for Wiener Filter Noise Response.

Figure 14 leads an observer to wonder if there are circumstances that will result in a restored object that is worse than the image before filtering. In other words, can the noise degradation become so great as to cancel out the improvements of the edge enhancement? Figure 15 is a plot of the standard error for the image before restoration vs the standard error for the restored object at various levels of degradation, where:

$$\text{Standard Error} = \sqrt{\sum \left[\text{Restored Image} - \text{Object}\right]^2 / n} \qquad \text{(Eq.42)}$$

It shows that the Wiener filter always yields an improvement regardless of the error resulting from the noise and cutoff frequency selected.

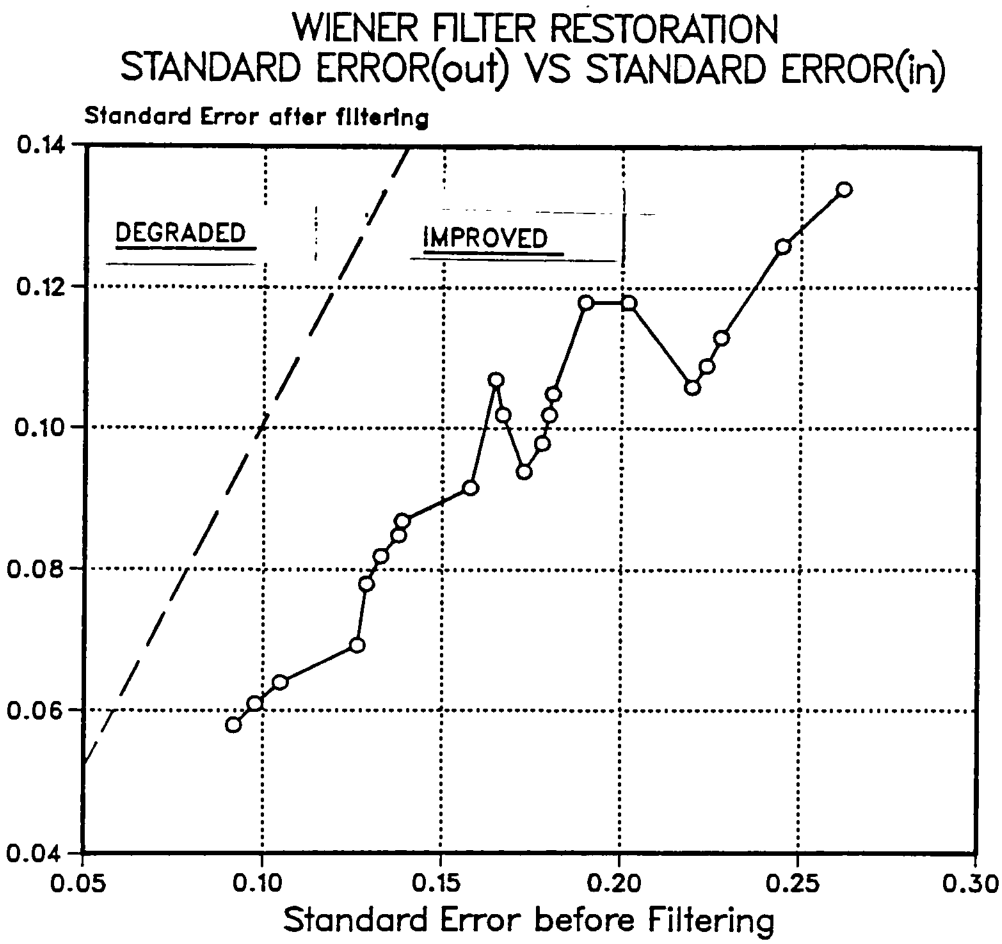


Figure 15. Wiener Filter Restoration Response.

Figure 16 shows how much improvement is made. Observe that doubling the noise does not result in a restored object with twice the restoration error, as some might expect. Instead, a greater proportion of restoration is gained from a noisier image than from an image that has less noise. This is depicted by the gradual leveling off of the curves as the noise increases; the more noise there is, the easier it is to remove. A user can quite accurately predict the degree of restoration the model will yield from this graph.

Figure 17 is a graph of the same data used to create figure 16 except cutoff frequency is the independent variable instead of noise. Observe that restoration error is gradually increasing as cutoff frequency decreases (blurring increasing). Because figure 17 shows greater curvature than figure 16, restoration is more dependent on the degree of blurring than the amount of noise in the image.

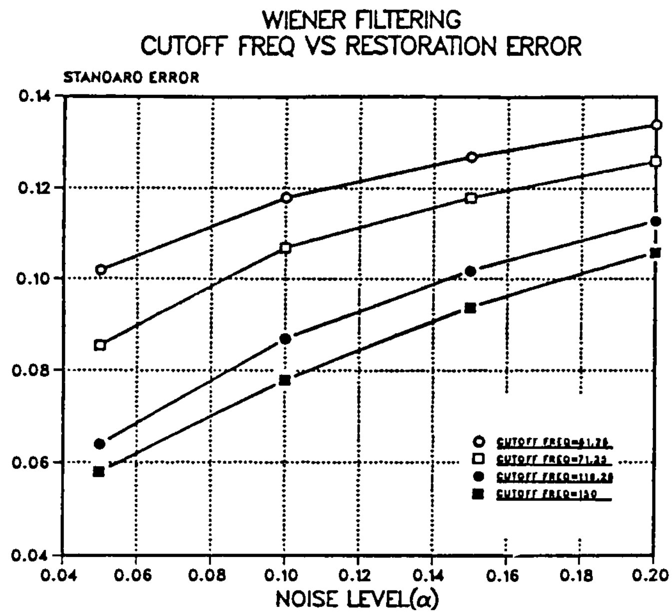


Figure 16. Restoring Noisy Images. Wiener Filter.

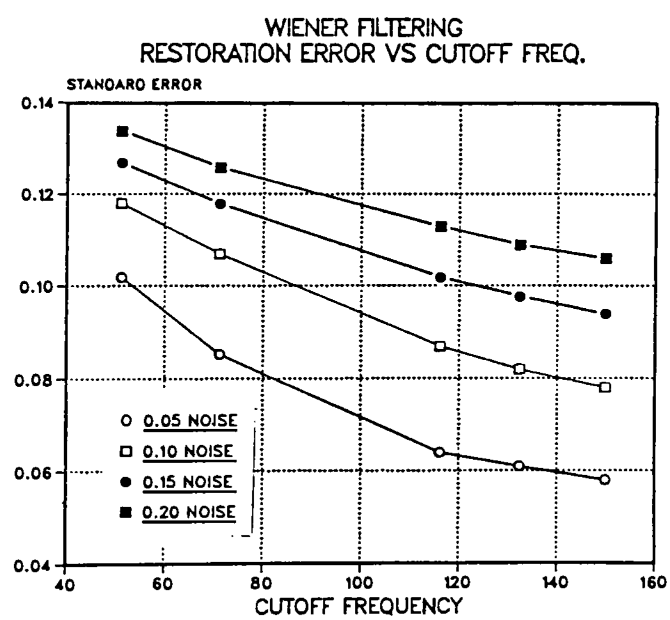


Figure 17. Restoring Blurred Images. Wiener Filter.

The performance of the unsharp mask will now be analyzed. Unsharp masking was simulated for an image on a film of gamma = 1.0, a mask of film gamma = .7, and the restored object on to a film with gamma = 3.334. Figure 18 shows the reconstructions of a 5cyc/mm object (left side) and a 10cyc/mm object (right side) degraded by a film with a cutoff frequency of 150cyc/mm under noiseless conditions via the direct method, an image placed in register with the unsharp mask, of unsharp masking. Note the 10cyc/mm object is 2X scaled to allow an easier comparison of the two restorations. Observe a better restoration of a target is attained from a less blurred image.

As just illustrated in the figure 18, unsharp masking is usually performed in the spatial domain. Since it went from a blurred image to a sharper image, the unsharp masking technique can be described in the frequency domain as a linear transfer filter. Using the mathematics developed in the Introduction, an approximation of a linear filter, the mask transfer function, for the technique is attainable. This is shown in figure 19. Observe in this figure that the same restoration as is shown in figure 18 resulted. Because the two methods yield identical results the model is not limited by the contrast criteria outlined by Armitage and Loehmann.

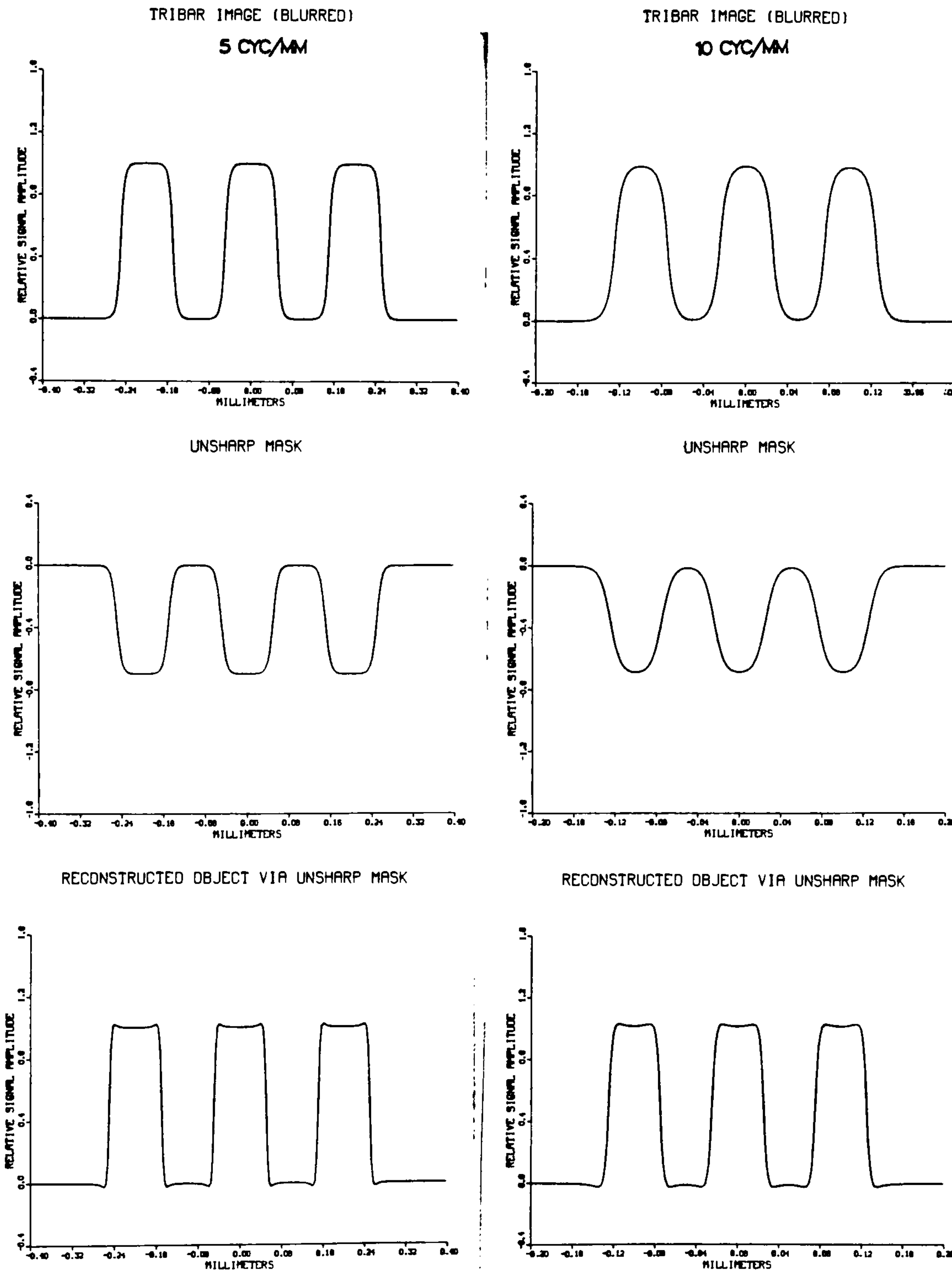


Figure 18. Restoring a Noiseless Object. Unsharp Masking.

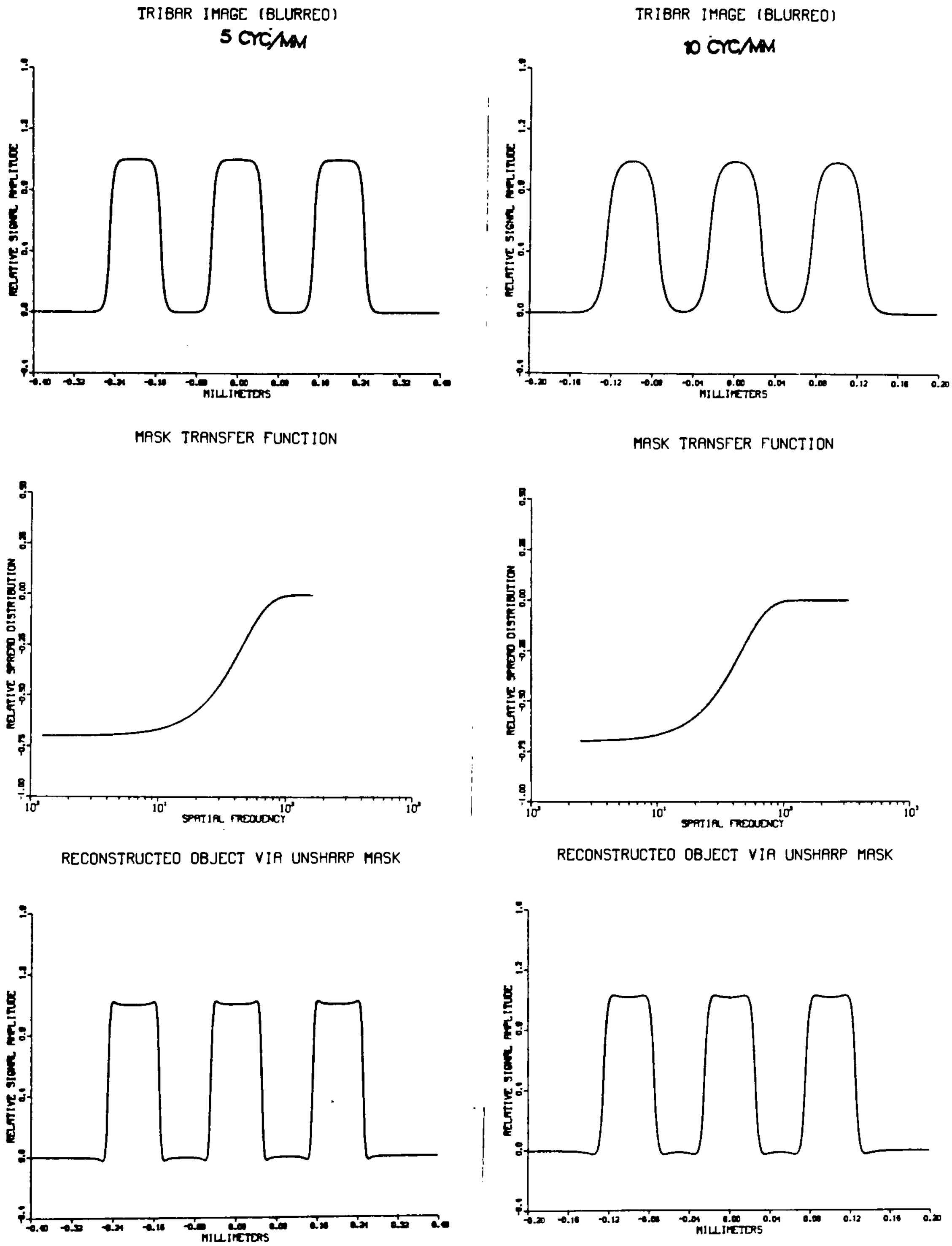


Figure 19. Restoring a Noiseless Object. Masking Filter.

Figure 20 shows the contrast enhancement (or the system MTF) achieved with the film specifications used to attain figures 18 and 19. Observe the system MTF rises above 1.0. This is why unsharp masking yields exaggerated edges.

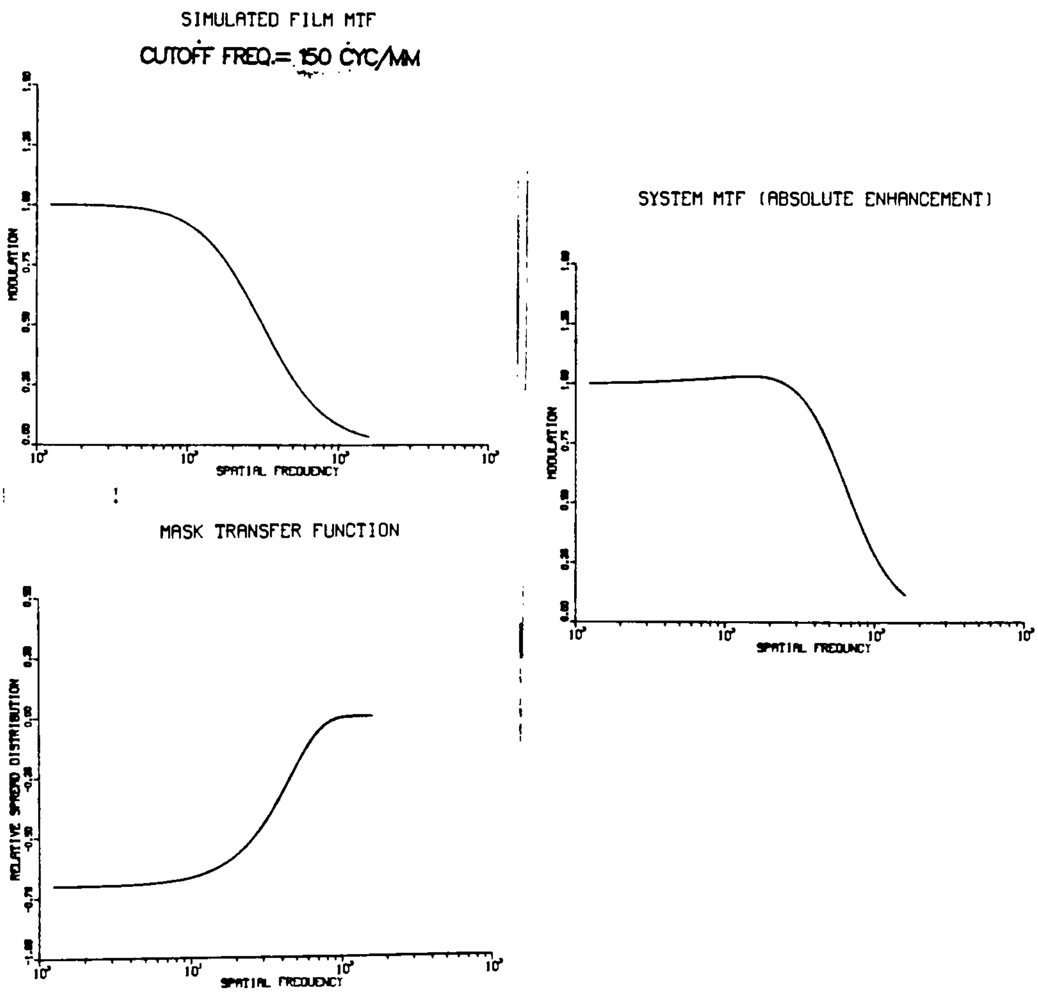


Figure 20. An Illustration of Contrast Enhancement.

It was just demonstrated that through unsharp masking, contrast enhancement, a noiseless image can be improved. But what about a noisy image? Will the unsharp mask improve it? The mathematics developed in the introduction to describe the technique does not include a noise component. Because the unsharp mask is a linear filter and does not compensate for noise, all the signals, desired signal as well as noise, increase in direct proportion to the filter's transfer function. Figure 21 illustrates the unsharp masking technique for a 5cyc/mm target degraded by a film with a cutoff frequency of 150cyc/mm and noise levels of .01 (left side) and .15 (right side). These are the same conditions set up in figure 13 for the Wiener filter. The graphs demonstrate that the unsharp masking technique sharpens edges at the expense of increasing the noise simultaneously. With the noise continually increasing, there comes a point in the image processing when more is lost than gained, when the image being restored is better than the restoration. For the high noise condition shown here this is indeed observed.

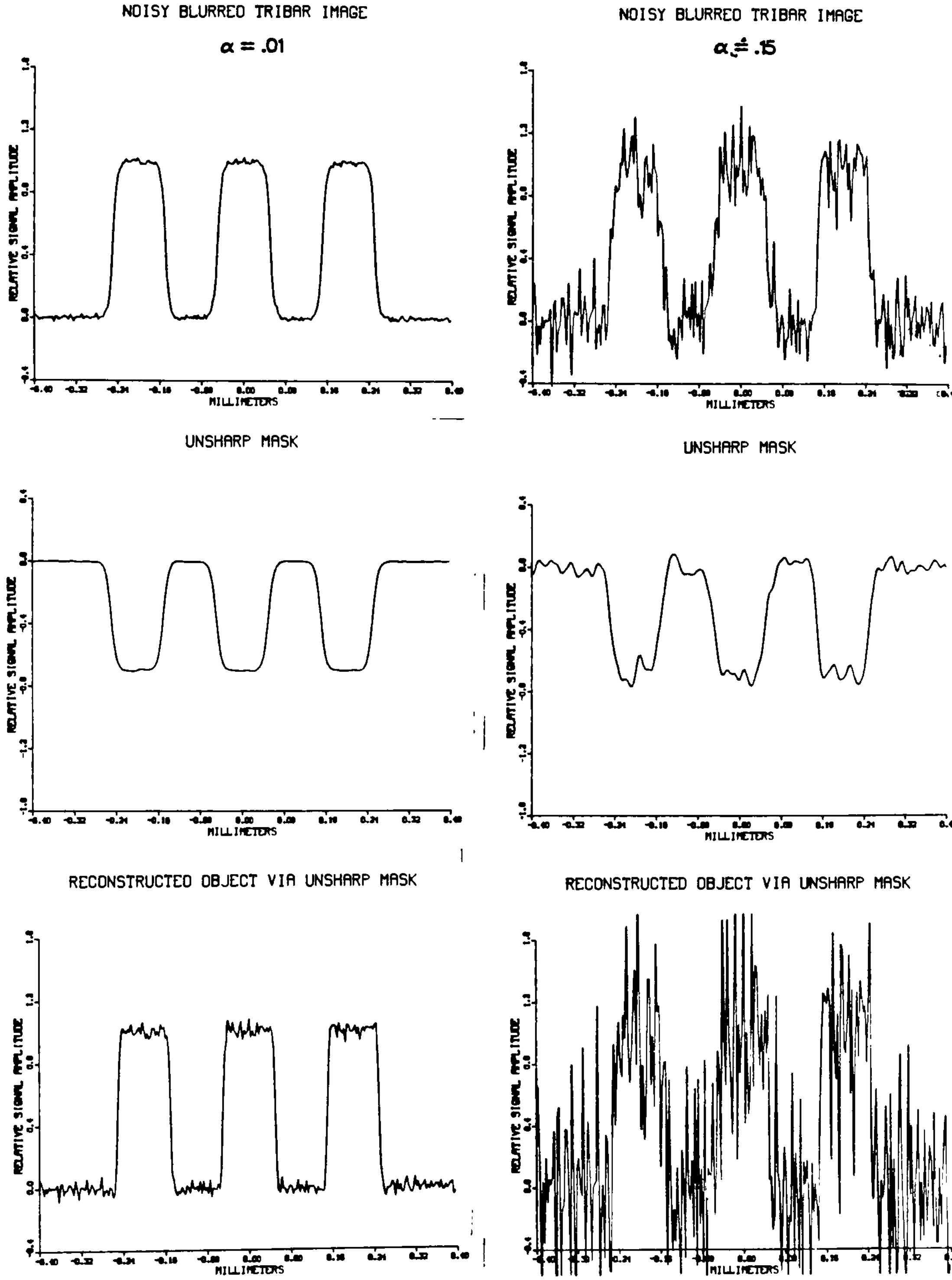


Figure 21. Restoring a Noisy Object. Unsharp Masking.

To determine at what point the unsharp masking technique is no longer beneficial, a series of restoration runs was performed using a 5cyc/mm tribar target, a film cutoff frequency of 150 cyc/mm, and variable noise to generate figure 22. Figure 22 reveals that at a certain point the ratio of the standard error for the restored object to the inputted image exceeded 1.0. For this case, the graph shows that for S/N(in) less than 42/1 more information is lost than recovered. This is due to the noise boosting degradation effect overwhelming the image restoration gains of its edge enhancement effect.

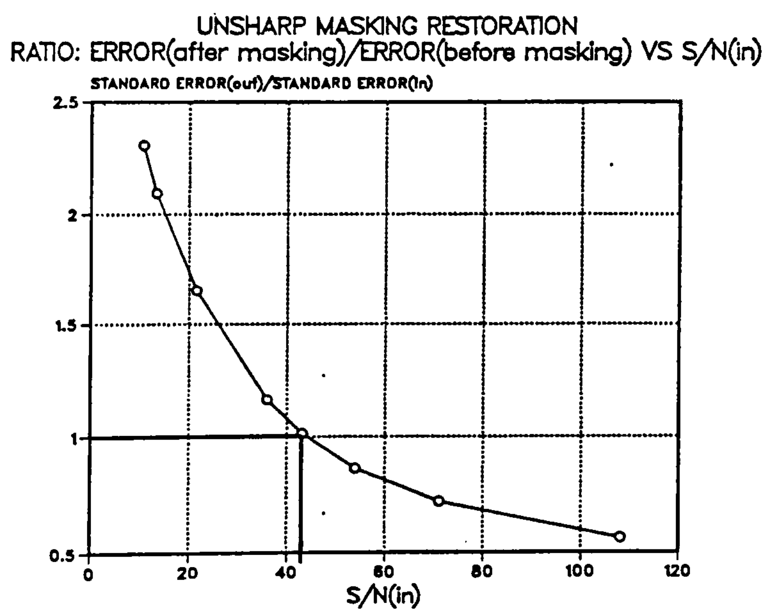


Figure 22. Performance of the Unsharp Masking Technique.

Figure 23 illustrates the effects of unsharp masking for various levels of degradation. Shown here is the standard error in reconstruction as a result of noise level for five different cutoff frequencies. Note the four curves are in very close proximity to each other, and they are rising quickly. From this it may be concluded that film cutoff frequency presents comparatively less hindrances to image recovery than is presented by noise. It is also interesting to note that unlike the Wiener filter (figure 16), doubling the noise here results in a restoration error that increases proportionally. To give a better idea of what this means, observe from the graph that the error associated with a cutoff frequency of 150cyc/mm and a noise level of .15 is 0.45. This is the degradation depicted in figure 21, a restoration almost indistinguishable.

Figure 24 is a graph of the same data used to generate figure 23 except cutoff frequency is the independent variable instead of noise. Observe that the results for the various levels are very close to parallel. This observation shows that no interaction between noise and spread exists; they are acted upon independently when performing unsharp masking. Also note that input noise has a great effect on restoration.

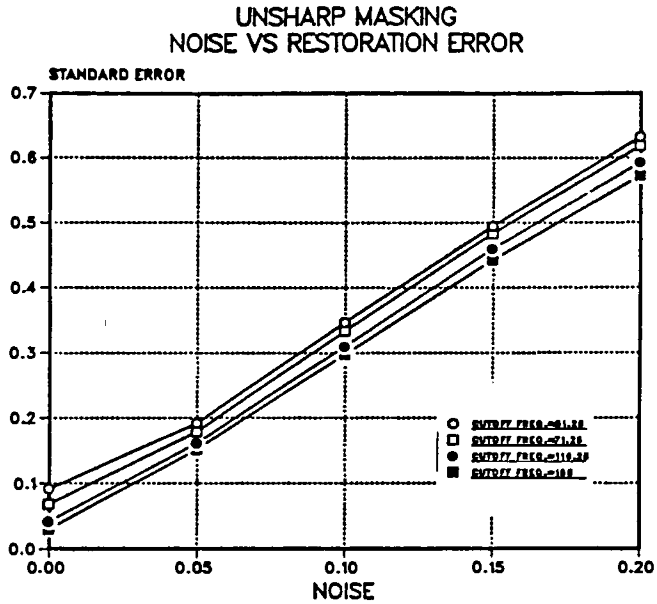


Figure 23. Restoring Noisy Images. Unsharp Masking.

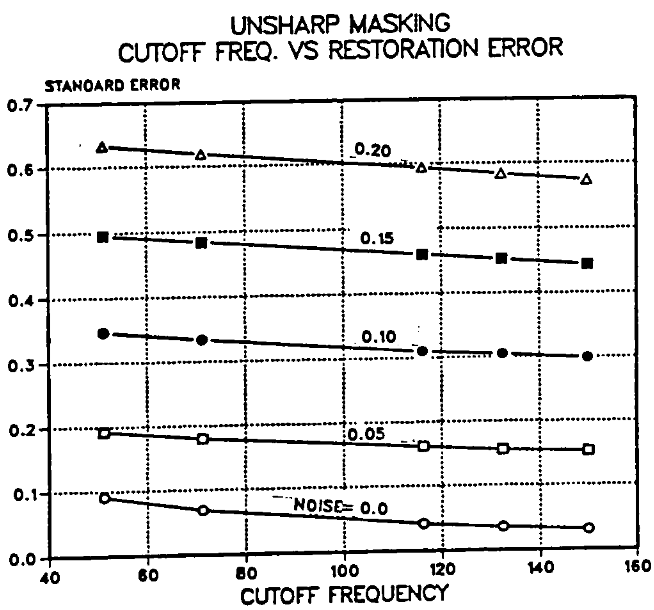


Figure 24. Restoring Blurred Images. Unsharp Masking.

A direct comparison of the Wiener filtering and unsharp masking techniques will now be presented. Many of the graphs following have already been seen, but here they are shown along side their counterpart. Figure 25 shows the two filters and the restored object attained from them for a noiseless image blurred by a film with a cutoff frequency of 150cyc/mm. Immediately it is observed that the Wiener filter is superior to unsharp masking for object restoration. Also seen in the figure is that both filters exhibit a rising spread distribution and similarity in shape but the Wiener filter does not level off, and it is rising much faster.

Figure 26 depicts the two filters and the restored object attained by the two techniques for an image degraded by a film with a cutoff frequency of150cyc/mm and a noise level of 0.15. Observe that the mask transfer function (filter) is unaltered in the presence of noise while the Wiener filter is. The up and down fluctuations that result gives the Wiener filter its noise removal capability. No noise removal is possible with the unsharp mask. Once again observe Wiener filtering is superior.

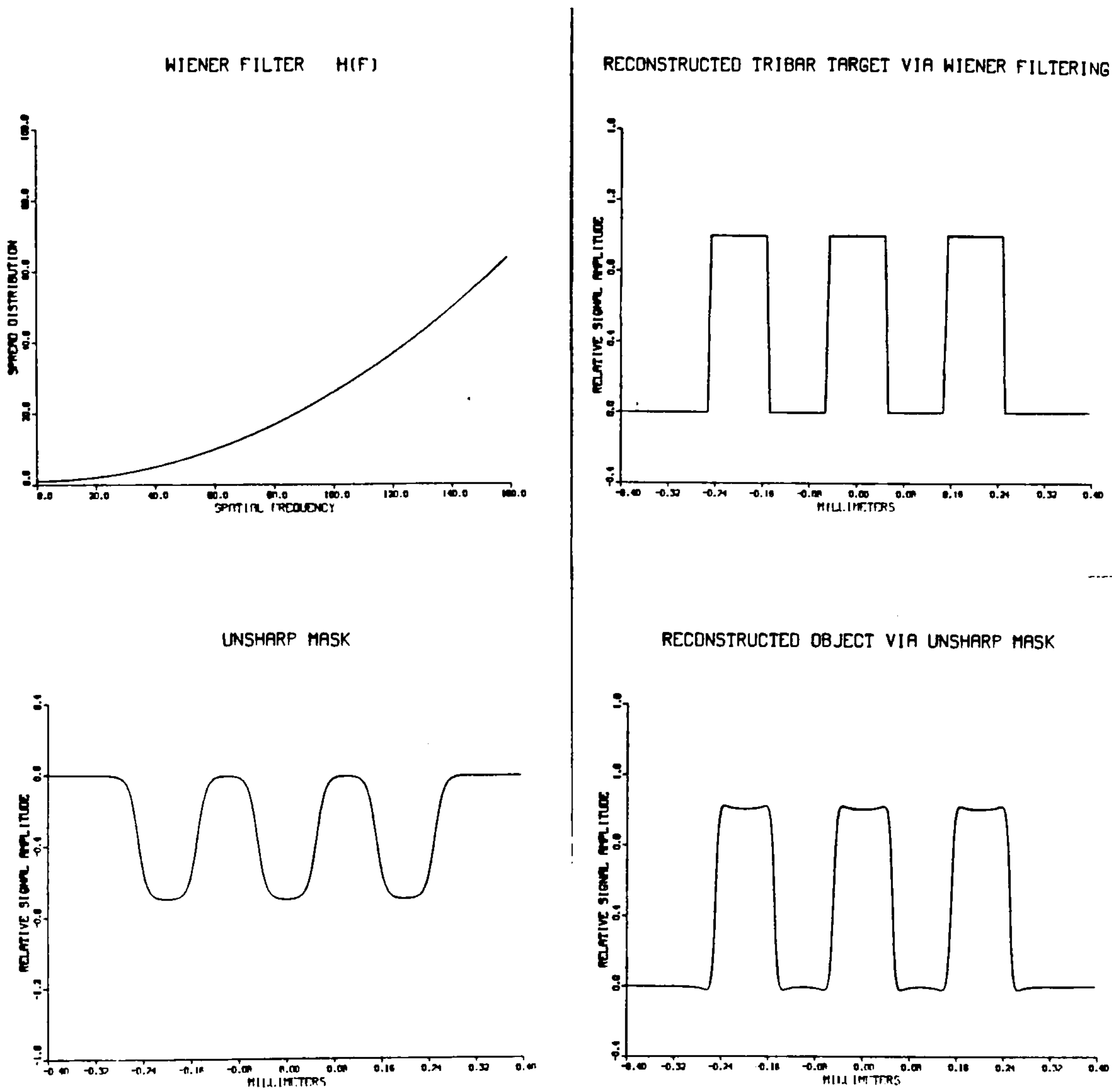


Figure 25. Noiseless Image Restoration.

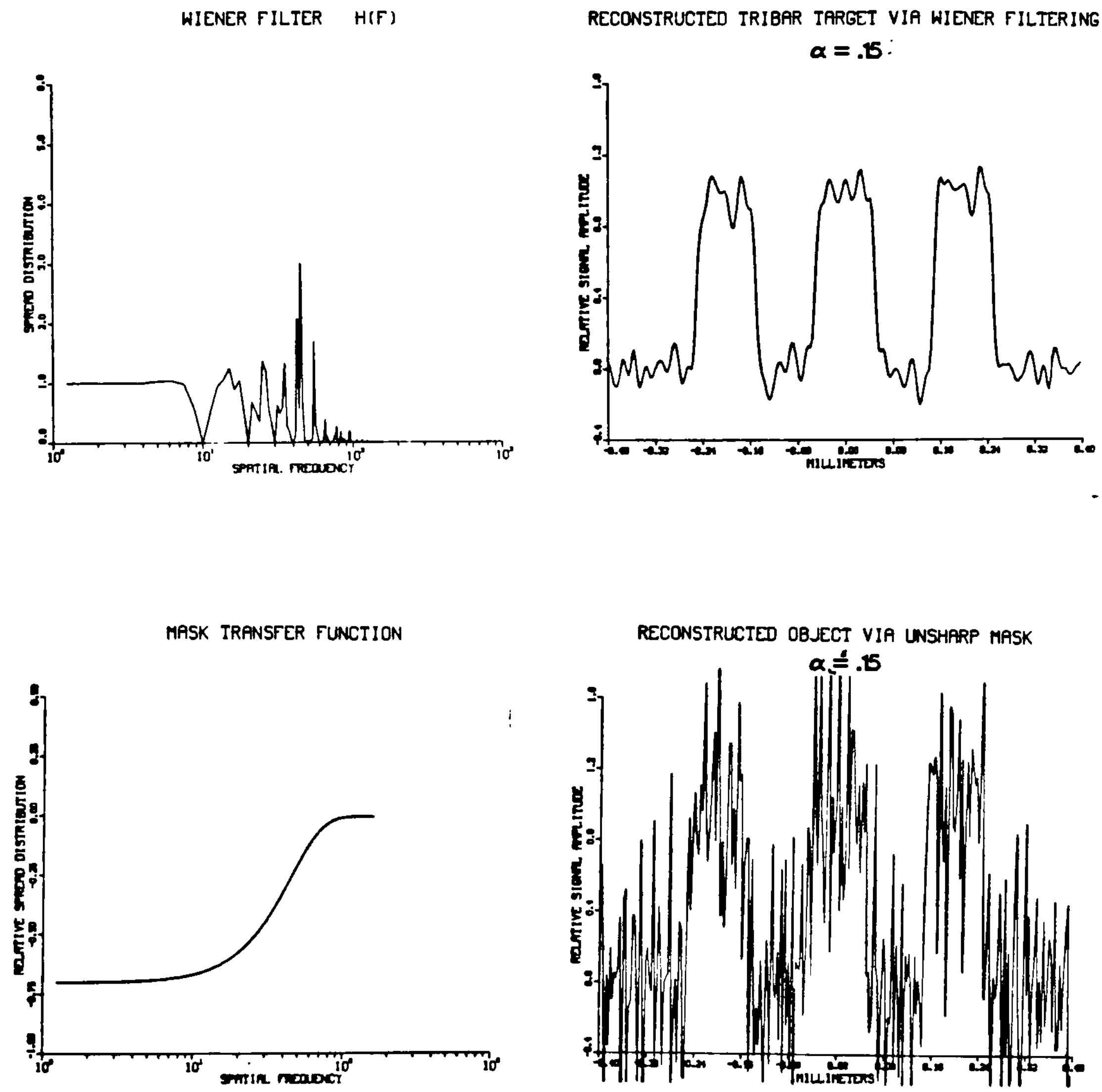


Figure 26. Noisy Image Restoration.

How much better is Wiener filtering than unsharp masking? Figures 27 and 28 show it to be a lot better, especially when the level of noise is high or increasing. In figure 27, it can be observed that the masking / Wiener filtering standard error ratios for restoration always exceeds 1.0. Figure 28 directly shows the Wiener filter performs better than the unsharp masking technique as blurring is varied.

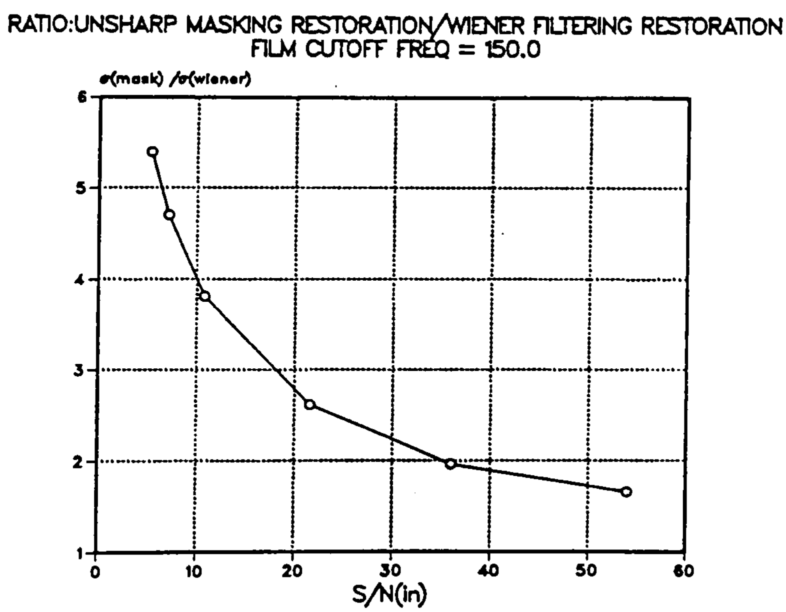


Figure 27. Comparison of Wiener Filter and Unsharp Masking Restoration Responses (Varying Noise).

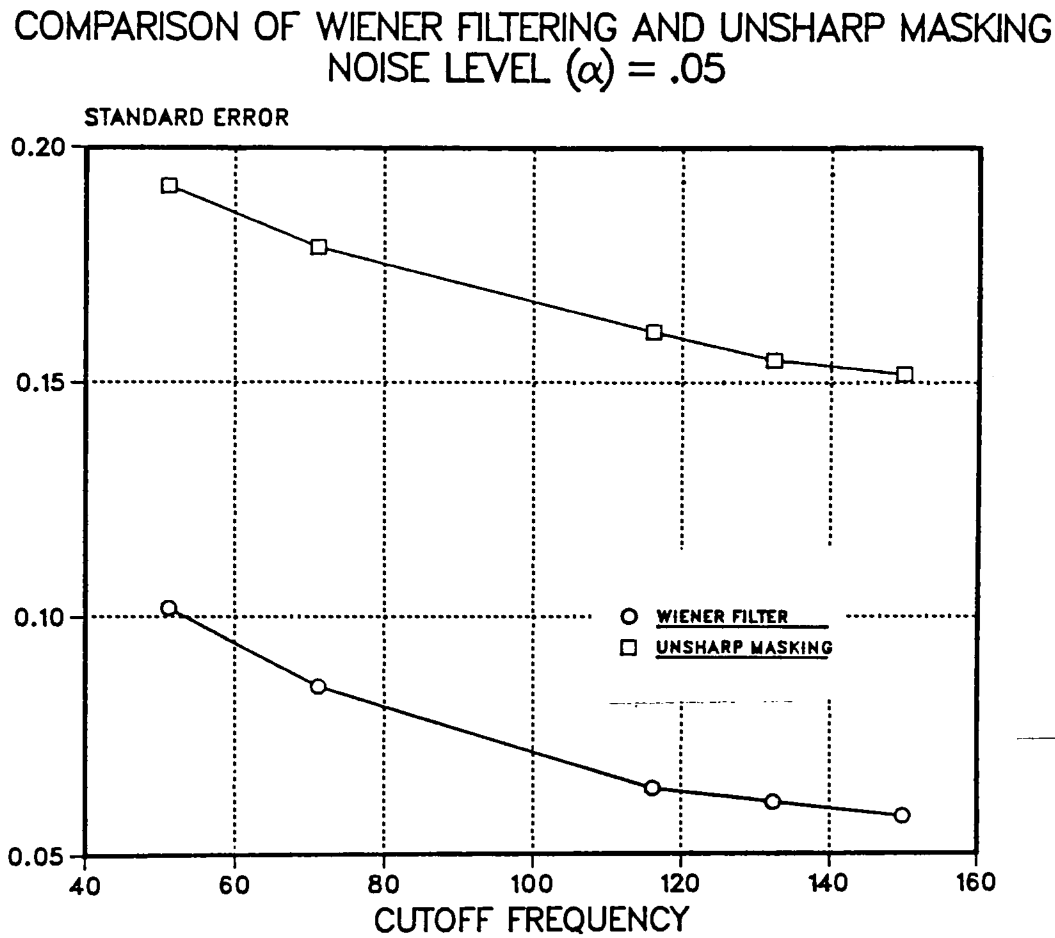


Figure 28. Comparison of Wiener Filter and Unsharp Masking Restoration Responses (Varying Cutoff Freq.).

It was stated in the introduction that unsharp masking approximated an inverse filter for a theoretical ideal case. The results presented in this section represent a realistic case that can be accomplished in the darkroom. The mathematics controlling the linear mask filter's range is dependent on the film Gammas chosen. With digital image processing Gamma is not limited as with film; any values may be selected. Using Gamma1 = 1.0, Gamma2 = .98, and Gamma3 = 50.0, and scaling the graph of the filter appropriately will illustrate this characteristic. Finding a real film with a consistently controllable Gamma3 value of 50.0 is unrealistic. Figure 29 shows the linear approximation to the unsharp mask closely resembles the noiseless Wiener filter (an inverse filter) when pushed to this extreme. Of course such an unsharp masking filter in the presence of noise is too detrimental to the restoration process to be of any use.

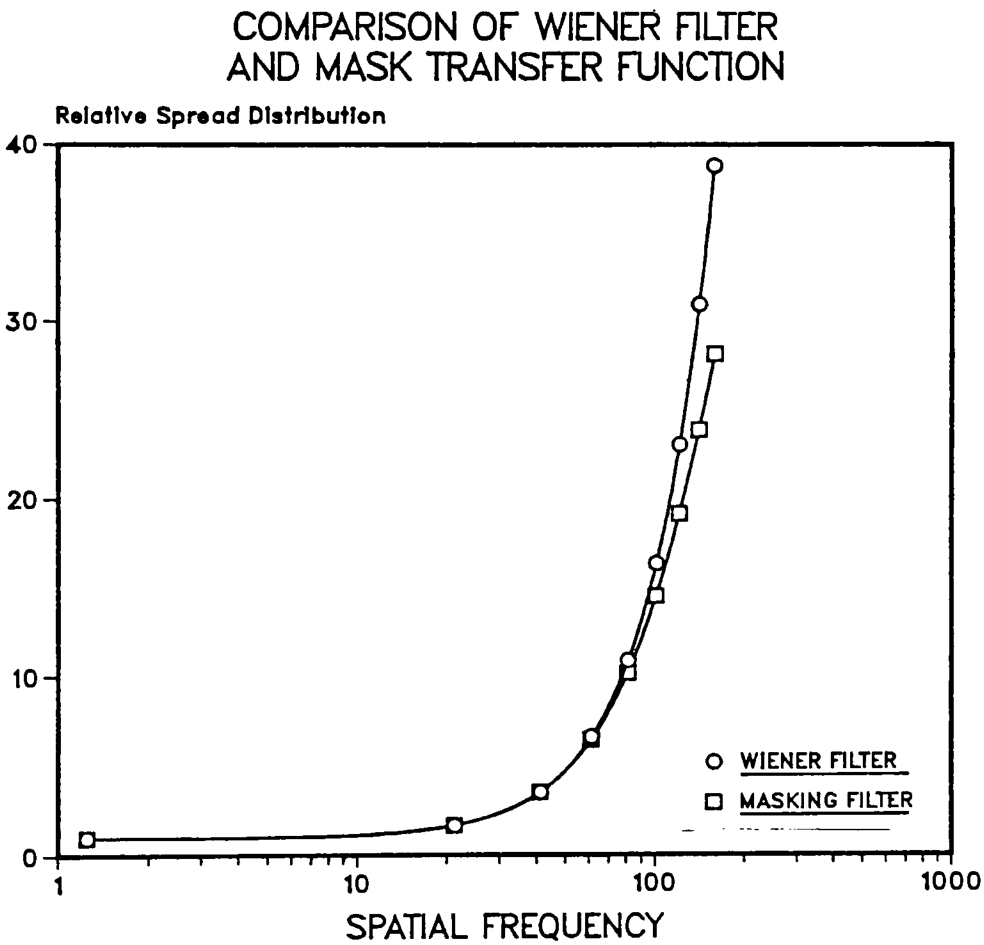


Figure 29. Unsharp Masking Approximation to the Inverse Filter.

CONCLUSIONS

This research concludes the Wiener filtering technique to be superior to the unsharp masking technique for digital image restoration. It was shown that for all levels of film blurring and noise degradation tested, the Wiener filter improved image quality and increased the information content available in the image via noise removal and edge enhancement. In contrast, unsharp masking is a high frequency content enhancer, and it does not discriminate between edges and noise. Therefore, both noise and edge signals are enhanced. Due to this, in the presence of high noise the unsharp masking technique fails as an image restoration technique.

It was also shown that in theory perfect restoration is attainable with Wiener filtering in the absence of noise because the filter becomes an inverse filter. Therefore, the Wiener filter in combination with the original film yields an ideal system with an MTF = 1.0 for the entire range of spatial frequencies. In comparison, perfect restoration is not attainable with unsharp masking under realistic conditions, and digitally it may only be approximated.

In addition, this paper has demonstrated that the unsharp mask can be expressed as a filter in terms of linear transfer theory. When this filter is cascaded with the original film MTF, the final MTF had values that exceeded 1.0 before dropping off. For this reason, exaggerated edges are observed in the restored objects.

RECOMMENDATIONS

It is recommended that future work on this topic be directed in three areas:

1. Two Dimensional Image Processing. The basic principles put forth in this research should be performed and evaluated using two dimensional digital image processing. This researcher feels the results contained within this thesis will be supported by such a work and further knowledge will be obtained.

2. Testing the Unsharp Masking Model Experimentally. The unsharp masking technique has the distinct advantage over the Wiener filter in that it can be performed in both the darkroom and on the computer. This affords the opportunity to refine and improve the model based on real results obtained experimentally.

3. Subjective Evaluation. Image produced two dimensionally using these techniques could be subjectively evaluated and analyzed, thus allowing the inclusion of the observer in future models.

REFERENCES

1. D.C.C. Wang, A.H. Vagnucci, and C.C. Li, "Digital Image Enhancement: A Survey," Computer Vision, Graphics, and Image Processing.,24, 363(1983).

2. J. D. Gaskill, Linear Systems, Fourier Transforms, and Optics. John Wiley and Sons, New York, 1978, pp 266-285.

3. J. W. Goodman, Introduction to Fourier Optics, McGraw-Hill, Inc., 1968, p21.

4. H.N. Todd and R.D. Zakia, Photographic Sensitometry The Study of Tone Reproduction, Morgan and Morgan, Inc., New York, 1969, p.264.

5. J.C. Croteau, "Comparison of Methods of MTF/OTF Analysis for Optical Systems," Thesis, Rochester Institute of Technology, Oct. 1983.

6. E.M. Granger, "Lecture to Image Science Class," Rochester Institute of Technology, January 1984.

7. H. Frieser, "Noise Spectrum of Developed Photographic Layers Exposed by Light, X-Rays, and Electrons," Photogr. Sci. and Eng., 3, 164, (July 1959).

8. J.C. Dainty and R. Shaw, Image Science, Academic Press, London, 1974, p.58.

9. Ibid.,p.58.

10. E.W.H. Selwyn, "A Theory of Graininess," Photogr. J., 75, 571, (1935).

11. P.S. Cosidine and R.A. Gonsalves, " Optical Image Enhancement and Image Restoration, " Topics In Applied Physics, 23, Sprig-Verlag, Berlin, 59, (1978).

12. Granger, April 1984.

13. Ibid.

14. Cosidine and Gonsalves.

15. Ibid.

16. H.J. Zweig and A. Silverstri, "Experiments in Digital Restoration of Defocused Grainy Photographs by Noise Cheating and Fourier Techniques," Proceedings Society of Photo-Optical Instrumentation Engineers, 74, (1975).

17. P.D. Burns and B.M. Levine, "Wiener Spectrum Estimation at Zero Frequency via Direct Digital Computation," Journal of Applied Photographic Engineering, 19, 78, (1983).

18. M. DeBelder and J. DeKerf, "Determination of the Wiener Spectrum of Photographic Emulsion Layers with Digital Methods," Photogr. Sci. Eng., 11, 371, (1967).

19. Dainty and Shaw, p.222.

20. R.C. Jones, "New Method of Describing and Measuring the Granularity of Photographic Materials," J. Opt. Soc. Am., 45, (1955).

21. N.C. Geckinli and D. Yavuz, "Some Novel Windows and a Concise Tutorial Comparison of Window Families," IEEE Trans. ASSP-26, 501, (Dec.1978).

22. Granger, April, 1984.

23. L.A. Scarff, "Quantification of the Unsharp Masking Technique of Image Enhancement," Thesis, Rochester Institute of Technology, June,1981.

24. J.A.C. Yule, "Unsharp Masks and a New Method of Increasing Definition in Prints," Phot. J. 84B, 321, (1944).

25. J.A.C. Yule, Principles of Color Reproduction, Wiley and Sons, Inc., NY, 1967, pp.74-78.

26. Ibid.

27. Ibid.

28. L. Levi, "Survey Unsharp Masking and Related Image Enhancement Techniques," Computer Graphics and Image Processing, 3, 163, (1974).

29. J.D. Armitage, A.W. Lohmann, and R.B. Herrick, "Absolute Contrast Enhancement," Applied Optics, 4, 445, (April 1965).

30. E. M. Granger, "A Summary Meassure of Image Quality," Proceedings of 27th Annual International Technical Symposium & Instrument Display, Aug 21/26 1983, vol 432.

31. Ibid.

32. E.M. Granger, "Lecture to Image Science Class," Rochester Institute of Technology, April 1984.

33. Armitage, Lohmann, and Herrick.

34. Ibid.

35. Scarff.

APPENDIX

The following documentation is a description of the programs and subroutines created for this research to model the imaging elements and perform the image processing and enhancement:

FAST. "FAST" is the major subroutine that sets up the parameters for the Fast Fourier Transform. FFTs allows the user to perform image processing in both the frequency or space domains. All data to be transformed is passed as a complex array to the IMSL routine FFT2C which performs the actual transformation. Scaling of the inverse transform is also performed by "FAST". Every program in the model that deals with imaging element and processing calls this subroutine.

BERCAN. "BERCAN" is the subroutine that performs all FFT wrap arounds and unwrapping. In performing an FFT, it is desirable to have the center of the imaging function at the zero data point in the array being transformed and the end point at the center of the array. This wrap around of the data puts the array in FFT format.

When displaying the function it is desirable to unwrap it and show it in its normal configuration.

DEVICE. "DEVICE" is a short subroutine that sets up some of the output parameters for the graphics produced by 'DISPLA'. DISPLA is a multipurpose FORTRAN graphics package developed by the Integrated Software Systems Corporation of San Diego, California, and is available on many main frame computers.

TRIBAR. The first step in the restoration model is to simulate an object in space. "TRIBAR" allows the user to simulate tri-bar targets of four different frequencies, an edge, and their Fourier Transforms.

FILM. "FILM" generates an MTF that simulates the amount of spreading expected by an image contact printed on to film. It allows the user to select an MTF cutoff frequency from 50 to 150 cycles/mm. The Frieser model is used here to generate the simulation. The Frieser model is defined as:

$$MTF = 1.0 / (1.0 + (X * F^{2}))$$

where F is frequency and X is a weighing function. Cutoff frequency is determined by the user's input

value for X. As can be readily seen, the MTF in the Frieser model does not go to 0 and therefore is not band limited. To prevent wrap around errors that can result from an unlimited function this researcher cuts the MTF at a modulation of .04. .04 is generally agreed upon by photointerpreters as the value where separate targets in an image are unresolvable.

NOISE. "NOISE" uses the IMSL subroutine 'GGNML' to generate gaussian distributed random numbers with a variance of 1.0. These numbers are then multiplied by the weighing function Alpha to be used as the random noise data. The value for Alpha is selected by the user. In addition, "NOISE" generates the smoothed or unsmoothed noise power spectrum necessary to create the Wiener filter.

WINDOW. "WINDOW" is a subroutine called by NOISE to perform triangular Bartlet or raised cosine windowing in the space domain. Each is used for calculating the smoothed noise power spectrum.

WIENER. "WIENER" generates the mathematical Wiener filter from the film MTF and the noise power spectrum. A thorough description of the Wiener filter is provided in the introduction.

SYSTEM. "SYSTEM" is a major program developed for this model. It cascades the image element simulations produced by TRIBAR, FILM, NOISE, and WIENER into a complete imaging system. The program outputs graphic displays and data files of the blurred image, the noisy blurred image, and the reconstructed image for analysis. The software performs the following mathematical (the symbolism is defined in the table of significant notation) operations:

```
O(X)  ==| O(F)*L(F)=IM(F)  ==| IM(X)+NO(X)  ==| NOIM(F)*H(F)
    ¶         ¶
  = O(F)  ==| O(X)
```

In addition, the program determines S/N before and after Wiener filtering using the Cosidine and Gonzalves (11) method outlined below:

$$S/N^{2} = \frac{(\ O(f) * L(f) * H(f)\ \ df\ )^{2}}{H(f)^{2} * N(f)^{2}\ \ df}$$

where the numerator is the filtered signal power and the denominator is the filtered noise power. To find

S/N before filtering remove the filter H(f) from the equation or set it equal to 1.0 for all frequencies. The values obtained here are used for further analysis.

MASK. "MASK" is the major program in the model performing the unsharp mask filtering routine to achieve image restoration. A very thorough description of the mathematical concepts used by this model is given in the introduction. To better understand what is going on in this program a review of this section is recommended. For this research, the following contrast values are generally used to illustrate this technique:

Original image contrast	Gamma1 = 1.0
Mask film contrast	Gamma2 = -.70
Restoration contrast	Gamma3 = 3.3334

Two processes are occurring in this program. Figure 30 illustrates the flow of these processes. The first flowchart shows the three inputs film, gamma1, gamma2 are used to produce the mask blurring function via the second moment technique. This result is used to produce the unsharp mask which when combined in register with the image and scaled back to the original contrast yields the restored object. The second flowchart illustrates how least error Contrast Enhancement is modeled by the program. The blurring

function, UOTF multiplied with the mask contrast, Gamma2 results in a linear approximation to the mask transfer function. The mask transfer function cascaded with the original film MTF then normalized yields the MTF of the entire degradation/restoration process for unsharp masking.

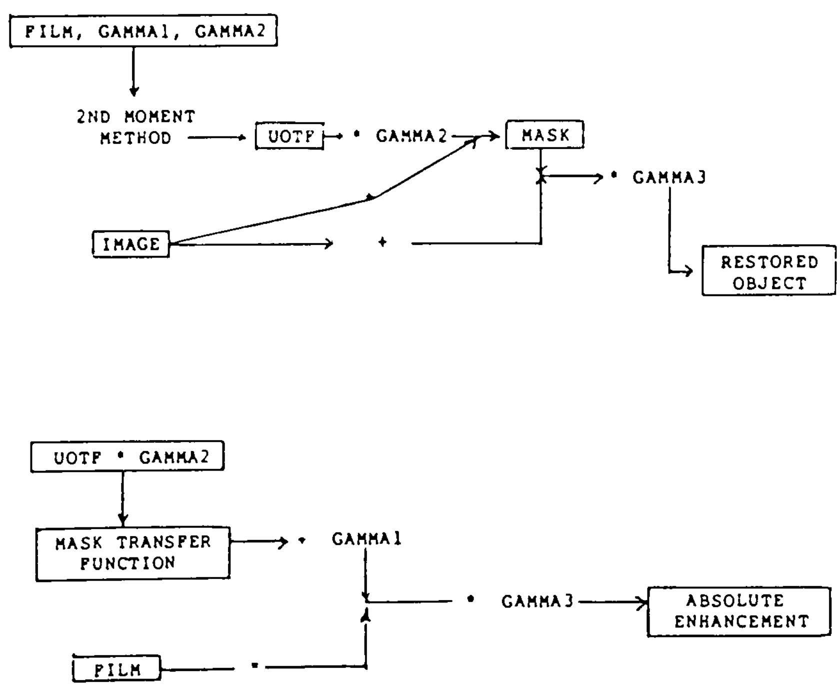


Figure 30. A Model For Unsharp Masking.

MEANSQER. "MEANSQER" is used to determine the mean-squared error (MSE) in the restored objects from the original object. This program first calculates the sum-squared error over three cycles of the reconstructed targets in the spatial domain; this is done so that noise in the data file outside our area of interest are not included in the analysis. The sum is then divided by the number of points over the three cycles to obtain the MSE. The MSE values for the two restoration methods calculated here are used for further analysis in the results section of this thesis.

EDGY and CONVOL. Two additional Image processing and evaluation routines developed by this researcher, "EDGY" and "CONVOL", are included in the appendix. Although these programs are not directly used by this model, they are included here because the image evaluator student could make good use of these software packages and find them to be a excellent aid to understanding many of the basic concepts of Imaging Science. "EDGY" determine the MTF of a function using the direct derivative edge trace method, and "CONVOL"

allows the user to input two functions and obtain a graphic and numerical display of their convolution.

```
      SUBROUTINE FAST(A,N,IWK,J)

C PROGRAM IDENTIFICATION :
C ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶
C
C
C  THIS PROGRAM COMPUTES THE FAST FOURIER TRANSFORM OF A COMPLEX
C  VALUED SEQUENCE OF LENGTH EQUAL TO A POWER OF 2 ACCORDING TO
C  THE FOLLOWING FORMULAE :
C  1.  FREQUENCY DOMAIN TO SPATIAL DOMAIN :
C      X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF
C        A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))
C      FOR K = 0,1,.....N-1 AND PI = 3.1415......
C  2.  SPATIAL DOMAIN TO FREQUENCY DOMAIN :
C     X(K+1) = SUM FROM J=0 TO N-1 OF
C           A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))
C     FOR K = 0,1,.....N-1 AND PI = 3.1415......
C  THE TRANSFORM IS EFFECTED BY CALLING THE LIBRARY SUBROUTINE
C  IMSL - FFT2C IN THE VAX-VMS SYSTEM.(I.E. CALL FFT2C (A,M,IWK))
C  SCALING FOR THE BACK TRANSFORM IS PERFORMED AFTER RETURNING
C  FROM FFT2C.
C
C********
C
C   VARIABLE IDENTIFICATION :
C¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶
C
C  N      =   NO OF POINTS IN FFT
C  M      =   INPUT EXPONENT TO WHICH 2 IS RAISED TO
C     produce the number of data points (N = 2**M)
C  A      =   COMPLEX VECTOR OF LENGTH N, WHERE N = 2**M.
C     On input A contains the Complex Valued Sequence
C     to be transformed.  On output A is repllaced by
C     the Fourier Transform.
C  IWK    =   WORK VECTOR OF LENGTH M+1
C  J      =   1  TRANSFORMS FROM SPACE TO FREQUENCY.
C  J      =   2   TRANSFORMS FROM FREQUENCY TO SPACE.
C
C  TYPE DECLARATION AND STORAGE ALLOCATION :
C ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶
C
       INTEGER M,IWK(256),I,J
       COMPLEX A(256)
       REAL  N
C
C*******
C
```

```
C  PROGRAM BLOCK :
C ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶
C

        M =(LOG10(N)/LOG10(2.0)+.5)

C  J=1 FORWARD TRANSFORM
C  J=2 INVERSE TRANSFER (FREQ TO SPACE)
        IF (J .EQ. 1) THEN
          CALL FFT2C (A,M,IWK)
           GOTO 700
        ELSE
          DO 10 I = 1,N
            A(I) = CONJG (A(I))
10        CONTINUE
        CALL FFT2C (A,M,IWK)

C  SCALES BACK TRANSFORM.
        DO 20 I = 1,N
          A(I) = CONJG(A(I))/N
20      CONTINUE


700     END IF
        RETURN
        END
```

```
      SUBROUTINE   BERCAN  (G,N,T)

C THIS SUBROUTINE WRAPS AROUND A SIMMULATED OBJECT/IMAGE INTO
C THE FFT FORMAT. IF ALREADY IN THE FFT FORMAT THE DATA WILL BE
C RETURNED TO ITS ORIGINAL FORMAT (THIS IS THE DESIRED FORMAT
C FOR INPUT/OUTPUT). DATA IS TRANSMITTED THROUGH ARRAY T().

      COMPLEX B(256),T(256),G(256)
      REAL N
      INTEGER I
      REAL L

      L=N/2.0

C  PUT T(129...256) INTO B(1...128)

      DO 10  I=1,L
         X=I+L
         B(I)=T(X)
10    CONTINUE

C    PUT T(1...128) INTO B(129...256)

      DO 20 I=1,L
        X=I+L
        B(X)=T(I)
20    CONTINUE

C   RETURN WITH OBJECT(IMAGE) BACK IN T(I)
      DO 30 I=1,N
        T(I)=B(I)
30    CONTINUE
       RETURN
       END
```

```
                    SUBROUTINE DEVICE

C  THIS ROUTINE INITIALIZES OUTPUT DEVICE AND SETS SOME STANDARD
C  PARAMETERS FOR THE GRAPHICS.

         CHARACTER*1 SIZE

        WRITE(6,*)'IS DISSPLA OUTPUT TO   1)IBM 3270    2)HP7221'
        WRITE(6,*)'    3)VT125      4)VAX PRINTER  ?'
        READ (5,*) DEVIC
           IF (DEVIC .EQ. 1) CALL IBM79
           IF (DEVIC .EQ. 2) CALL HP7221
           IF (DEVIC .EQ. 3) CALL PREGIS
           IF (DEVIC .EQ. 4) CALL PPNTNX
        CALL NOBRDR
        CALL PAGE (11,8.5)
C       WRITE (6,*)'DO YOU WANT TO SET THE PLOT SIZE?   (Y/N) '
C       WRITE(6,*)'MAX= 10.5 BY 6.5    " N " DEFAULTS TO 8 BY 6.'
C       READ (5,99) SIZE
C       IF(SIZE .EQ. 'Y')THEN
C        WRITE(6,*)'INPUT X'
C        READ(5,*) X
C        WRITE(6,*)'INPUT Y'
C        READ (5,*)Y
C        CALL AREA2D (X,Y)
C       ELSE
          CALL AREA2D (8,6)
C       ENDIF
90      RETURN
99      FORMAT(A)
        END
```

```
C                 TRIBAR.FOR BY JAY H. BERMAN

C     PROGRAM CREATES A SIMMULATED TRIBAR OBJECT FOR A 256 POINT
C     FFT 1 UNITS HIGH. HEIGHT IS ARBITRARILY CHOSSEN AT 1.0
C     FOR CONVIENCE. FREQUENCY (DELTA F)=1.25 ALSO CHOSSEN FOR
C     CONVIENCE. THE FOLLOWING TARGETS WERE DESIGNED USING THE
C     FORMULAR:  F*X*N=1.0.  FOR (DELTA X)= 1/320 MILLIMETER,
C     THE SPATIAL RANGE IS 256*1/320 = .80  OR -.40 MM TO .40 MM.
C     THE OBJECT(O(X)) IS OUTPUTTED IN WRAPP AROUND (FFT) FORMAT
C     TO TRIBAR DATA. IT'S TRANSFORM (O(F))IS OUTPUTTED TO FTRIBR
C     DATA. THE SUBROUTINE BEERCAN IS CALLED TO DO THE WRAP
C     AROUNDING.
C

C     PLOTS ARE DISPLAYED WITHOUT WRAP AROUND.

C     THE USER MAY CHOOSE 40,20,10, OR 5 CYCLES/MM.
C     CODING ADDED 3/21/85 TO GENERATE AN EDGE TO PASS THROUGH THE
C     SYSTEM TO DEMONSTRATE YULE'S PRINCIPLE (SHOW CORN-SWEET EDGE)

         COMPLEX T (256), G(256),A(256)
         DIMENSION XAXIS(256),YAXIS(256)
         CHARACTER CHOICE*1
         INTEGER IWK(256)
         REAL N
         N=256
         FRQ=1.25

C   G IS A DUMMY ARRAY NOT USED IN THIS PROGRAM BUT IS NEEDED FOR
C   LINKING TO SUBROUTINE BERCAN,WHICH USES G() FOR ANOTHER MAIN
C   PROGRAM.
        WRITE (6,*)'          TRIBAR FORTRAN BY JAY H. BERMAN
     C
     C                                                          '
        WRITE(6,*)'YOUR SELECTED TRIBAR TARGET AND ITS'
        WRITE(6,*)'  TRANSFORM WILL BE PLOTTED.
     C                                                          '
10      WRITE (6,*) '                                         '
        WRITE (6,*) 'CHOOSE A TRIBAR TARGET. (INPUT 1,2,3, OR 4)'
        WRITE (6,*) '    OR AN EDGE TARGET  (INPUT 5.)
     C                                                          '

        WRITE (6,*)  '  1)  40 CYC/MM              2)  20 CYC/MM  '
        WRITE (6,*)  '  3)  10 CYC/MM              4)   5 CYC/MM  '
        WRITE (6,*)  '  5)  AN EDGE                6)  10 CYC/MM OTHER'

        READ (5,*) CYCLES
```

```
          IF (CYCLES .EQ. 1) GO TO 100
          IF (CYCLES .EQ. 2) GO TO 200
          IF (CYCLES .EQ. 3) GO TO 300
          IF (CYCLES .EQ. 4) GO TO 400
          IF (CYCLES .EQ. 5) GO TO 500
C OBJECT 6 ADDED FOR GRANGER TEST
                                  IF (CYCLES .EQ. 6) GO TO 400

        WRITE (6,*) '         '
        WRITE (6,*)  '  YOU MADE AN ERROR.  CHOOSE AGAIN
     C                                                   '

        GO TO 10


C   FOLLOWING PRODUCES A 40 CYC/MM TRIBAR TARGET


100       DO 110 I=1,N
             IF (I .GE. 120 .AND. I .LE. 122) T(I)=1.0
             IF (I .GE. 128 .AND. I .LE. 130) T(I)=1.0
             IF (I .GE. 136 .AND. I .LE. 138)  T(I)=1.0
             IF (I .EQ. 119 .OR. I .EQ. 123) T(I)=.5
             IF (I .EQ. 127 .OR. I .EQ. 131) T(I)=.5
             IF (I .EQ. 135 .OR. I .EQ. 139) T(I)=.5
110       CONTINUE

            GO TO 800


C   FOLLOWING PRODUCES A 20 CYC/MM TRIBAR TARGET

200       DO 210 I=1,N
             IF (I .GE. 110 .AND. I .LE. 116) T(I)=1.0
             IF (I .GE. 126 .AND. I .LE. 132) T(I)=1.0
             IF (I .GE. 142 .AND. I .LE. 148)  T(I)=1.0
             IF (I .EQ. 109 .OR. I .EQ. 117) T(I)=.5
             IF (I .EQ. 125 .OR. I .EQ. 133) T(I)=.5
             IF (I .EQ. 141 .OR. I .EQ. 149) T(I)=.5
210       CONTINUE

            GO TO 800


C   FOLLOWING PRODUCES A 10 CYC/MM TRIBAR TARGET
```

```
300     DO 310 I=1,N
          IF (I .GE. 90 .AND. I .LE. 104) T(I)=1.0
          IF (I .GE. 122 .AND. I .LE. 136) T(I)=1.0
          IF (I .GE. 154 .AND. I .LE. 168)  T(I)=1.0
          IF (I .EQ. 89 .OR. I .EQ. 105) T(I)=.5
          IF (I .EQ. 121 .OR. I .EQ. 137) T(I)=.5
          IF (I .EQ. 153 .OR. I .EQ. 169) T(I)=.5
310     CONTINUE

          GO TO 800


C   FOLLOWING PRODUCES A 5 CYC/MM TRIBAR TARGET
C                  OR DOUBLE SCALED 10 CYC/MM TARGET

400     DO 410 I=1,N
          IF (I .GE. 50 .AND. I .LE. 80)   T(I)=1.0
          IF (I .GE. 114 .AND. I .LE. 144) T(I)=1.0
          IF (I .GE. 178 .AND. I .LE. 208) T(I)=1.0
          IF (I .EQ. 49 .OR. I .EQ. 81)    T(I)=.5
          IF (I .EQ. 113 .OR. I .EQ. 145)  T(I)=.5
          IF (I .EQ. 177 .OR. I .EQ. 209)  T(I)=.5
410     CONTINUE

           GO TO 800

C   FOLLOWING PRODUCES AN EDGE

500     DO 510 I=1,N
          IF (I .LT. 129) T(I)=.4
          IF (I .EQ. 129) T(I)= .9
          IF (I .GT. 129) T(I)=1.4
510     CONTINUE


C   FOLLOWING DISPLAYS THE MODEL TRIBAR TARGET USING 'DISPLA'.

800      DO 810 I=1,N
           XAXIS(I)=(I-129.0)/320.0
           YAXIS(I)= REAL(T(I))
810     CONTINUE

        CALL DEVICE
        CALL XNAME('MILLIMETERS$',100)
```

```
          CALL YNAME ('RELATIVE SIGNAL AMPLITUDE$',100)
          IF (CYCLES .EQ. 1) CALL HEADIN ('40 CYC/MM TRIBAR TARGET (O(X))
     C$',100,1.5,1)
          IF (CYCLES .EQ. 2) CALL HEADIN ('20 CYC/MM TRIBAR TARGET (O(X))
     C$',100,1.5,1)
          IF (CYCLES .EQ. 3) CALL HEADIN ('10 CYC/MM TRIBAR TARGET (O(X))
     C$',100,1.5,1)
          IF (CYCLES .EQ. 4) CALL HEADIN ('5 CYC/MM TRIBAR TARGET (O(X))$
     C',100,1.5,1)


C FOLLOWING FOR DOUBLE SCALED 10 CYC/MM
          IF (CYCLES .EQ. 6) THEN
                CALL HEADIN('10 CYC/MM TRIBAR TARGET (O(X))$',100,1.5,1)
820        DO 830 I=1,N
             XAXIS(I)=(I-129.0)/640.0
830       CONTINUE
                CALL GRAF (-.20,.04,.20,-.4,.4,1.6)
          ELSE

             CALL GRAF (-.40,.08,.40,-.4,.4,1.6)
          ENDIF

          CALL SETCLR ('RED')
          CALL CURVE (XAXIS,YAXIS,256,0)
          CALL ENDPL (0)
          CALL DONEPL
          CLOSE(16)

C   TRIBAR TARGET WILL BE WRAPPED AROUND INTO FFT FORMAT,
C   AND STORED IN TRIBAR DATA(O(X)).
          CALL BERCAN (G,N,T)
          OPEN (1)
          DO 900 I=1,N
             WRITE (1,*) T(I)
900       CONTINUE
          CLOSE(1)
C
C   O(X) ==|O(F) (FTRIBR DATA) AND IS STORED IN FFT FORMAT.
          DO 901 I=1,N
           A(I)=T(I)
901       CONTINUE
          CALL FAST (A,N,IWK,1)

          OPEN (11)
          DO 902 I=1,N
            WRITE(11,*)A(I)
902       CONTINUE
```

```
      CLOSE(11)

C********** FOLLOWING CODING ADDED 3/13 FOR ANNALYSIS********
C  FIND AREA UNDER THE CURVE FOR O(F) TO THE FILM CUTOFF FREQ
        WRITE(6,*)'
     C                                                        '
        WRITE(6,*)'DO YOU WANT THE AREA UNDER O(F)?   Y/N'
        READ (5,998)CHOICE
          IF (CHOICE .EQ. 'Y') THEN

              FRAREA=0.0
        WRITE(6,*)'INPUT CUTOFF FREQ OF FILM BEING USED.'
        READ(5,*)CUTOFF
          RNG=CUTOFF/1.25
        DO 925 X=1,RNG
          FRAREA = FRAREA + ABS(A(X))*1.25
925     CONTINUE
        DO 926 X=N-RNG,N
          FRAREA = FRAREA + ABS(A(X))*1.25
926     CONTINUE
         WRITE(6,*)'FREQUENCY DOMAIN AREA =',FRAREA

          ELSE
           ENDIF

C***************************************************************


C  UNWRAP FTRIBR, SCALE TO 1.0 AND DISPLA.
      DO 903 I=1,N
        T(I)=A(I)
903   CONTINUE
      CALL BERCAN (G,N,T)
      WRITE(6,*)'
     C                                                        '
      WRITE(6,*)'SCALER= T(129)=',ABS(T(129))
      WRITE(6,*)'
     C                                                        '
905   DO 910 I=1,N
          XAXIS(I)=(I-N/2) *FRQ
          YAXIS(I)= (T(I)/T(129))
910   CONTINUE

      CALL DEVICE
      CALL YNAME ('RELATIVE SIGNAL AMPLITUDE$',100)
      CALL XNAME('SPATIAL FREQUENCY$',100)
```

```
          CALL GRAF (-160.0,40.0,160.0,-.4,.4,1.6)
C         CALL MARKER (4)
          CALL SETCLR ('RED')
          IF(CYCLES.EQ.1)CALL HEADIN('O(F) = ABS (FFT(40 CYC/MM TRIBAR)$'
     C,100,1.5,1)
          IF(CYCLES.EQ.2)CALL HEADIN('O(F) = ABS (FFT(20 CYC/MM TRIBAR)$'
     C,100,1.5,1)
          IF(CYCLES.EQ.3)CALL HEADIN('O(F) = ABS (FFT(10 CYC/MM TRIBAR)$'
     C,100,1.5,1)
          IF(CYCLES.EQ.4)CALL HEADIN('O(F) = ABS (FFT(5 CYC/MM TRIBAR)$'
     C,100,1.5,1)
          CALL CURVE (XAXIS,YAXIS,256,0)
          CALL ENDPL (0)
          CALL DONEPL
          CLOSE(16)

C  *   USER OUTPUT MESSAGE
          WRITE (6,*) '
     C
     C                                                                  '
          WRITE (6,*)'YOUR SELECTED TRIBAR TARGET(O(X)) IS IN FILE TRIBAR
     C DATA AND ITS TRANSFORM IS IN FTRIBR DATA (O(F)). '
         WRITE(6,*)' THEY ARE WRAPPED AROUND.(FFT FORMATED).'
998       FORMAT(A)
999       STOP
          END
```

```
C                   FILM FORTRAN BY JAY H BERMAN

C   FILM FORTRAN PRODUCES AN MTF THAT CLOSELY MODELS THE MTF OF
C   FILM. THE USER SELECTS A VALUE FOR X AND THE MODEL IS
C   PRODUCED. THE FILM MTF IS USED AS THE L(F) (OTF) FOR THE
C   SYSTEM FORTRAN PROGRAM. MTF VALUES = .04 IS REGARGDED
C   AS THE CUTOFF FREQUENCY.
C   THIS VALUE IS USED BECAUSE PHOTOINTERPERTERS AGREE THAT
C   MTF VALUES BELOW THIS ARE UNRESOLVABLE. THE USER WILL BE
C   INFORMED OF THE CUTOFF FREQUENCY.  THE CUTOFF FREQUENCY IS
C   FROM 50 CYC/MM TO 155 CYC/MM.  THE PROGRAM USES H.H. FREISER
C   FILM MODEL:
C                    MTF= 1/(1+X*F*F)

C*******************DECLARATIONS BLOCK*******************

C  G IS AN UNUSED DUMMY ARRAY NEEDED FOR CALLING BERCAN.
C  XAXIS AND YAXIS ARE USED BY THE PLOTTER.

        COMPLEX T(256),G(256),A(256)
        DIMENSION MTF(256)
        DIMENSION XAXIS (128)
        DIMENSION YAXIS (128)
                CHARACTER*1 CHOICE
        INTEGER IWK(256)
        REAL MTF
        REAL N
        N=256
        FRQ=1.25
        FLAG=0.0
C
C*********************************************************

C********ADDED FOR DOUBLE SCALED 10 CYC/MM OBJECT*******
        WRITE(6,*)' ARE YOU USING THE DOUBLE SCALED TARGET?  Y/N'
        READ (5,999)CHOICE
999     FORMAT (A1)
          IF (CHOICE .EQ. 'Y') FRQ=2.5


1       WRITE(6,*)'                                             '
        WRITE(6,*) ' INPUT A VALUE FOR X= ( .0011 TO .0100 ).  X IS A'
        WRITE(6,*) ' WEIGHING FUNCTION;  FOR X=.0011 FILM CUTOFF MTF='
        WRITE(6,*) ' 155CC/MM AND FOR X=.01 FILM CUTOFF MTF= 50CC/MM.'
        READ(5,*) X
```

```
C   CHECKS INPUT
C       IF (X .LT. .0011 .OR. X .GT. .0100) THEN
C       WRITE (6,*)'
C    C          INPUT ERROR MADE. BOUNDARY EXCEEDED.   TRY AGAIN.
C    C                                                       '
C        GO TO 1

C       ELSE
C       END IF


C  CREATE THE FILM SPREAD FUNCTION. THE MTF CUT OFF FREQ IS
C  CALCULATED BY THE PROGRAM.


C   FINDS RIGHT SIDE OF MTF FUNCTION.

      B=129
      DO 10 F=0,N/2*FRQ,FRQ
          MTF(B)= 1.0/(1.0+X*F*F)
           IF(MTF(B) .LT. .04) THEN
             IF(FLAG .EQ. 1.0) GOTO 7
               FLAG = 1.0
               B=B+1
               CUTOFF = (B-129) * FRQ
          ELSE
7        B=B+1
      END IF
10    CONTINUE


C    FOLLOWING STORES CUTOFF FREQUENCY. MTF(129) BECOMES MTF(0) AFTER
C     IT IS WRAPPED AROUND.
C

C   IT IS STANDARD AMONG THE INDUSTRY TO ONLY GRAPHICALLY SHOW THE RIGHT
C   HALF OF THE FILM MTF CURVE ON A XLOG SCALE.  FOR THIS REASON THIS
C   IS WHAT WILL BE OUTPUTTED TO LOGICAL 32 IF ON VAX OR LOG 16 ON IBM.

C   DISSPLA PLOTING CALLS. (DISSPLA IS A PLOTTING PACKAGE BY ISSCO OF
C   SAN DIEGO, CALIF.

C        CREATE THE X AXIS FOR THE GRAPH. XAXIS VALUES START AT F(1)
C            FOR EASE OF DISPLAYING.

      DO 14 I=1,128,1
```

```
          XAXIS(I) = I*FRQ
14      CONTINUE

C  FOLLOWING CREATES THE Y AXIS.
        DO 15 B=1,128,1
           YAXIS (B) = MTF(B+128)
15      CONTINUE

        WRITE (6,*)'DO YOU WANT A  1)XLOG   OR  2)CARTESIAN GRAPH ?
     C                                                INPUT 1 OR 2.'
        READ (5,*) GRAPH

        CALL DEVICE
        CALL XNAME ('SPATIAL FREQUENCY$',100)
        CALL YNAME ('MODULATION$',100)
           IF (GRAPH .EQ. 1) CALL XLOG (1.0,2.5,0.0,.25)
           IF (GRAPH .EQ. 2) CALL GRAF(0.0,20.0,160.0,0.0,.2,1.2)
        CALL HEADIN('SIMULATED FILM MTF$',100,1.5,1)

        CALL SETCLR ('RED')
        CALL CURVE (XAXIS,YAXIS,128,0)
        CALL ENDPL (0)
        CALL DONEPL
        CLOSE(16)


C   FINDS LEFT SIDE OF SPREAD FUNCTION. SINCE THE FUNCTION IS
C   SYMMETRIC ABOUT THE AXIS THE CUTOFF FREQ IS THE SAME
C   FOR BOTH SIDES OF THE FUNCTION.

          B=128

        DO 16 F=FRQ,N/2*FRQ,FRQ
            MTF(B)= 1.0/(1.0+X*F*F)
                 B=B-1
16      CONTINUE


C  GET MODEL FILM MTF INTO FFT FORMAT AND OUTPUT IT TO FILMMTF.

        DO 20 B=1,N
         T(B)=MTF(B)
20      CONTINUE

        CALL BERCAN (G,N,T)

          OPEN (1)
```

```
      DO 30 B=1,N
       WRITE (1,*) T(B)
30    CONTINUE

      CLOSE(1)

C   USER INFORMATION MESSAGES

       WRITE (6,*)'
     C
     C
     C                                                          '
       WRITE (6,*)'THE MTF CUTS OFF AT A FREQ = ',CUTOFF
       WRITE (6,*)'
     C                                                           '
      WRITE(6,*)'FILMMTF DATA CONTAINS THE MODEL FILM IN FFT FORMAT'
       STOP
       END
```

```
C                     NOISE.FOR BY JAY H BERMAN


C  THIS PROGRAM CREATES A GAUSSIAN DISTIBUTED NOISE ARRAY OF 256
C  RANDOM NUMBERS WITH A VARRIANCE OF 1.
C  THE VAX ROUTINE GGNML WILL BE USE TO GENERATE THE NOISE N(X).
C  THEN N(X) * ALPHA (A NOISE LEVEL). THE NOISE IS WINDOWED AND
C  OUTPUT ON LOG 010, THE POWER SPECTRUM IS OUTPUTED ON LOG 007,
C  AND THE SPATIAL NOISE POWER ON LOG 009.
C  THIS PROGRAM IS LINKED TO FAST,WINDOW,IMSL/LIB.


C     THE FOLLOWING ARRANGEMENT OF MATHEMATICAL MANIPULATIONS
C     ARE PERFORMED BY THE PROGRAM:

C          N(X) * ALPHA ==| THEN WINDOWS IT (OPTIONAL)   ==|
C          N(F) * CONJ N(F) (ITS POWER SPECTRUM)  ==| N(X)**2


C     ARRAY DECLARATIONS

       COMPLEX G(256)
       COMPLEX A(256)
       COMPLEX T(256),NOISE(256),POWER(256)
       INTEGER  NR,IWK(256)
       REAL  R(256)
       DOUBLE PRECISION DSEED
       REAL N
       INTEGER I,J
       DIMENSION XAXIS(256),YAXIS(256)
       CHARACTER CHOICE*1
C******CONSTANTS*******

       N=256
       FRQ=1.25

C  THE FOLLOWING READS IN 256 RANDOM NUMBERS GENERATED BY GGNML

       NR=256
       DSEED=254787.D0
       CALL GGNML (DSEED,NR,R)


C  THE FOLLOWING MOVES THE NUMBERS INTO A(I) WHERE IT IS THEN SENT
C  TO FAST TO BE TRANSFORMED. AFTER IT RETURNS IT IS MULTIPLIED BY
C  ALPHA.  (NOISE IS PUT IN 'A').
```

```
       WRITE (6,*)'INPUT A VALUE (LESS THAN .30 FOR BEST RESULTS) FOR'
       WRITE (6,*)'NOISE LEVEL ALPHA. AN ALPHA VALUE OF 0 MEANS THE'
       WRITE (6,*)'SYSTEM IS NOISE FREE.
     C                                '
        READ (5,*) ALPHA

        DO 10  I=1,N
           NOISE(I) = R(I) * ALPHA
10      CONTINUE

C  OUTPUT NOISE TO LOG 10
         DO 11 I=1,N
          WRITE (10,*)NOISE(I)
11       CONTINUE


C  FOLLOWING TRANSFORMS NOISE N(X)*ALPHA TO FREQ, GETS THE POWER
C  SPECTRUM TRANSFORMS IT BACK TO SPACE AND OUTPUTS THEM
C  (SPECTRUM TO 008, AND IT'S TRANSFORM TO 009), IN FFT FORMAT.

           DO 15 I=1,N
              A(I)=NOISE(I)
15         CONTINUE

C 'TRANSFORMS FROM SPACE TO FREQUENCY'
          CALL FAST(A,N,IWK,1)

         DO 20  I=1,N
            POWER(I)=A(I) * CONJG(A(I))
20        CONTINUE

          DO 21 I=1,N
            WRITE (8,*) POWER(I)
21        CONTINUE

C 'TRANSFORMS FROM FREQ BACK TO SPACE'
         CALL FAST(A,N,IWK,2)
         DO 22 I=1,N
           WRITE(9,*)POWER(I)
22       CONTINUE


35       WRITE(6,*)'
     C                                                            '
         WRITE(6,*)'DO YOU WANT A GRAPH OF THE NOISE THAT IS TO'
         WRITE(6,*)'BE ADDED TO THE SYSTEM ?
     C                            INPUT Y OR  N.  '
         READ (5,701) CHOICE
```

```
          IF (CHOICE .EQ. 'N') GO TO 80
           REWIND 10
           READ(10,*) (T(I), I=1,N)
C UNWRAP FOR OUTPUT GRAPH DISPLAY
           CALL BERCAN (G,N,T)

        DO 60 I=1,N
           YAXIS(I)= REAL ( T(I) )
           XAXIS(I)= (I-129.0)/320.0
60      CONTINUE

C  CALL DEVICE INITIALIZES GRAPHIC OUTPUT DEVICE AND SETS
C  PAGE AND AXIS PARAMETERS.
           CALL DEVICE
           CALL HEADIN ('RANDOM GAUSSIAN NOISE$',100,1.5,1)
           CALL XNAME ('MILLIMETERS$',100)
           CALL YNAME ('NOISE AMPLITUDE$',100)
           IF (ALPHA .GE. .2) THEN
               CALL GRAF (-.40,.08,.40,-1.0,.5,1.0)
           ELSE
               CALL GRAF (-.40,.08,.40,-.5,.25,.5)
           ENDIF
           CALL MARKER (4)
           CALL SETCLR ('RED')
           CALL CURVE (XAXIS,YAXIS,256,0)
           CALL ENDGR(0)
           CALL ENDPL (0)
           CALL DONEPL
           CLOSE(16)


C****FOLLOWING SETS UP THE POWER SPECTRUM GRAPHICS****

C THE USER MAY WINDOW THE NOISE  ( A WINDOW THAT IS CALLED
C IN FROM A SUBROUTINE IS USED). THE RESULT IS RETURNED IN
C G(I).THIS IS USED FOR POWER SPECTRUM ESTIMATION OF THE
C SYSTEM NOISE. SINCE NOISE IS RANDOM, WRAP AROUND IS
C EFFECTIVELY ACHIEVED BY USING A WRAPPED AROUND WINDOW.

80      WRITE (6,*)'
     C
     C                                                        '
        WRITE(6,*)'DO YOU WANT A POWER SPECTRUM GRAPH?'
        WRITE(6,*)'         Y  OR  N
     C                                                        '
        READ (5,701) CHOICE
         IF (CHOICE .EQ. 'N') GO TO 200
C
```

```
         WRITE (6,*) 'DO YOU WISH TO WINDOW THE NOISE FOR SMOOTHED'
         WRITE (6,*) 'POWER SPECTRUM ESTIMATION GRAPH ?
      C                                                        '
         WRITE (6,*) '  1.YES       2.NO      ( INPUT 1 OR 2 )
      C                                                        '
         READ (5,*) WND
            IF (WND .EQ. 1) THEN

C GO TO WINDOW.FOR SUBROUTINE THEN CALCULATES SMOOTHED SPECTRUM.
       CALL WINDOW (G,N,T)
       DO 90 I=1,N
          A(I)=NOISE(I)*G(I)
90     CONTINUE

       CALL FAST(A,N,IWK,1)
       DO 95 I=1,N
          T(I)=A(I)*CONJG(A(I))
95     CONTINUE

       ELSE
           REWIND 8
           READ (8,*)(T(I), I=1,N)
       ENDIF


C UNWRAP FOR OUTPUT GRAPH DISPLA
          CALL BERCAN (G,N,T)

       DO 110 I=1,N
          YAXIS(I)= REAL (T(I))
          XAXIS(I)= (I-(N/2.0)) * FRQ
110    CONTINUE

        CALL DEVICE
        CALL HEADIN ('NOISE POWER SPECTRUM, N(F)**2$',100,1.5,1)
        CALL XNAME ('FREQUENCY$',100)
        CALL YNAME ('NOISE POWER AMPLITUDE$',100)
         IF (WND .EQ. 2 .AND. ALPHA .LE. .2)THEN
           CALL GRAF (-160.0,32.0,160.0,0.0,10.0,50.0)
         ELSE
           IF (WND .EQ. 2 .AND. ALPHA .GT. .2 ) THEN
             CALL GRAF(-160.0,32.0,160.0,0.0,20.0,120.0)
           ELSE
           IF ( ALPHA .GT. .2 ) THEN
             CALL GRAF(-160.0,32.0,160.0,0.0,5.0,30.0)
           ELSE
             CALL GRAF(-160.0,32.0,160.0,0.0,3.0,15.0)
           ENDIF
```

```
          ENDIF
        ENDIF
       CALL MARKER(4)
       CALL SETCLR ('RED')
       CALL CURVE (XAXIS,YAXIS,256,0)
       CALL ENDGR(0)
       CALL ENDPL(0)
       CALL DONEPL
        WRITE(6,*)'
     C
     C
     C                                                              '
200   WRITE(6,*)'THE POWER SECTRUM IS ON LOG 008, ITS TRANSFORM IS'
      WRITE(6,*)'ON LOG009, AND THE WINDOWED NOISE IS ON LOG010.'
      WRITE(6,*)'IF ALPHA = 0 WAS CHOSSEN, THEN THESE WILL = 0.'
        CLOSE(8)
        CLOSE(9)
        CLOSE (10)

701    FORMAT(A1)
800    STOP
       END
```

```
C             SUBROUTINE WINDOW.FOR BY JAY H BERMAN

C    THIS ROUTINE PRODUCES THE BARTLET OR RAISED COSINE WINDOW IN BEER
C    CANNED FORM TO BE USED WITH NOISE.FOR. THE WINDOW IS UNIT HEIGHT
C    AND COVERS HALF THE POINTS IN THE SPACE DOMAIN.
C    THE USER MAY INCORPERATE HIS OWN WINDOW (READ IN FROM LOGICAL 20
C    FREE-FORMATTED).

      SUBROUTINE WINDOW (G,N,T)

      COMPLEX G(256),T(256),TEMP(256)
      DIMENSION XAXIS(256),YAXIS(256)
      REAL N
      INTEGER I
      CHARACTER*1 CHOICE
      PI=3.14159

C                SELECT A WINDOW

1     WRITE (6,*) 'WHAT KIND OF NOISE WINDOW DO YOU WANT?
     C                                           '

       WRITE (6,*) '    1. BARTLET                    2. RAISED COSINE'
       WRITE (6,*) '    3. USER PROVIDED WINDOW
     C                                                            '

       WRITE (6,*) ' ( TYPE   1, 2, OR 3 )
     C                                                            '

      READ (5,*) W
         IF (W .EQ. 1) GO TO 5
         IF (W .EQ. 2) GO TO 50
         IF (W .EQ. 3) GO TO 80

      WRITE (6,*) 'INPUT ERROR MADE.  TRY AGAIN
     C                                                          '
         GO TO 1

C    GENERATE A BARTLET WINDOW DIRECTLY IN FFT FORMAT.

5     DO 10 I=1,65
         G(I)=1.0-((I-1.0)/64.0)
10    CONTINUE
```

```
      DO 20 I=66,192
          G(I)=CMPLX(0,0)
20    CONTINUE

      DO 30 I=193,N
          G(I)=(I-193.0)/64.0
30    CONTINUE

C FOLLOWING UNWRAPS BARTLET WINDOW FOR GRAPHICS DISPLA.
       DO 31 I=1,N
         T(I)= G(I)
31     CONTINUE
       CALL BERCAN (G,N,T)
          GO TO 500



C   GENERATE A RAISED COSINE WINDOW THEN SEND TO BEERCAN TO
C   GET IT IN FFT FORMAT. PUT WINDOW IN G(I) THEN RETURN TO
C   NOISE.FOR PROGRAM

50    DO 60 I=1,N
           X=(I-129.0)/129.0
         IF (X .LT. -.5 .OR. X .GT. .5) THEN
            T(I)=0.0
         ELSE
            T(I) = .5 + .5 * ( COS (2.0*PI*X) )
         END IF
C                     TEMPORARY STORAGE OF UNWAPPED WINDOW
                    TEMP(I)=T(I)
60    CONTINUE

      CALL BERCAN (G,N,T)

      DO 70 I=1,N
         G(I)=T(I)
70    CONTINUE
                    DO 71 I=1,N
                       T(I)=TEMP(I)
71                  CONTINUE

      GO TO 500

C   READ IN USER PROVIDED WINDOW FROM LOGICAL 20. G(I) CONTAINS FFT FOR-   -
C   MATTED WINDOW AND T(I) THE UNWRAPPED VERSION FOR GRAPIHC AT COMPLE-
C   TION OF READ IN.
```

```
80       WRITE(6,*)'DOES YOUR WINDOW NEED TO BE WRAPPED INTO FFT FORMAT ?
     C                            INPUT  Y OR N.'
         READ (5,700) CHOICE
           IF (CHOICE .EQ. 'Y') THEN
             READ (20,*) (T(I), I=1,N)
             CALL BERCAN (G,N,T)
             DO 81 I=1,N
               G(I)= T(I)
81           CONTINUE
           ELSE
             READ(20,*) (G(I), I=1,N)
             DO 82 I=1,N
               T(I)=G(I)
82           CONTINUE
             CALL BERCAN(G,N,T)
         ENDIF
500     WRITE(6,*)'
     C                                                                  '
        WRITE(6,*)'DO YOU WANT TO GRAPH THE WINDOW ?    Y OR N '
        READ(5,700) CHOICE
          IF (CHOICE .EQ. 'N') GO TO 600

C  FOLLOWING PERFORMS THE GRAPHICS. T(I) WHICH CONTAINS THE UNWRAPPED
C  WINDOW WILL BE OUTPUTTED.
        DO 501 I=1,N
          XAXIS(I)= (I-129.0)/320.0
          YAXIS(I)=  REAL(T(I))
501     CONTINUE
        CALL DEVICE
        CALL XNAME('MILIMETERS$',100)
        CALL YNAME('FILTERING FACTORS$',100)
        CALL HEADIN('NOISE WINDOW FILTER  W(X)$',100,1.5,1)
        CALL GRAF(-.40,.08,.40,0.0,-.4,1.6)
        CALL MARKER(16)
        CALL SETCLR('RED')
        CALL CURVE(XAXIS,YAXIS,256,0)
        CALL ENDPL(0)
        CALL DONEPL
        CLOSE(16)


600     RETURN
700     FORMAT (A)
        END
```

```
C                          WIENER.FOR BY JAY H BERMAN

C     THIS PROGRAM PROOUCES THE WIENER FILTER H(F) WHICH IS USEO IN THE
C     RECONSTRUCTION OF A DERGRADED IMAGE IN AN OPTICAL SYSTEM.

C     THE FOLLOWING MATHEMATICAL MANIPULATIONS ARE MAOE TO PRODUCE H(F)

C        CONJ L(F) / L(F)**2 + ( NO(F)**2 / O(F)**2 ) - H(F)

C     MAKE THE FOLLOWING ASSIGNMENTS PRIOR TO RUNNING THE PROGRAM

C          L(F)    THE OTF OF THE LENS/FILM          INPUT ON LOG 009
C          NO(F)**2    THE POWER SPECTRUM            INPUT ON LOG 010
C          O(F)    TRANSFORM OF THE OBJECT           INPUT ON LOG 011
C          H(F)    THE WIENER FILTER                 OUTPUT ON LOG 007

C    ARRAY DECLARATIONS
C    ------------------
         COMPLEX H(256),R(256),L(256),O(256),NO(256),DENOM(256),T(256)
         DIMENSION XAXIS(12B),YAXIS(12B)
         INTEGER  F,X
         REAL N
         N=256
         FRQ=1.25
         YMAX =0.0
                WRITE(6,*) 'INPUT FRQ. 1.25 OR 2.5 (OOUBLE SCALE 10)'
                READ (5,*) FRQ


C   PROGRAM BLOCK
C   -------------


C   REAOS IN L(F),THEN CONJUGATES IT INTO R(F)(WORKING ARRAY)

        READ (9,*) (L(F),F=1,N)

        DO 10 F=1,N
        R(F)   CONJG (L(F))
10      CONTINUE


C  REAOS IN THE POWER SPECTRUM,        AND SQUARES L(F)
```

```
        READ (10,*) (NO(F),F-1,N)

        DO 20 F-1,N
        L(F)= L(F)* CONJG (L(F))
20      CONTINUE


C  READS IN O(F) THEN MULTIPLIES IT WITH ITS CDNJG = O(F)**2

        READ (11,*) (O(F),F=1,N)

       DO 30   F=1,N
          O(F)   O(F) * CONJG (O(F))
30      CONTINUE


C  CACULATES H(F) WITH ABOVE EQUATION

         DO 40 F=1,N
C         IF (O(F) .EQ. 0) THEN
                    IF (O(F) .EQ. 0) O(F)=.00000000001
           DENOM(F)= L(F) + ( NO(F)/O(F) )


        IF (DENOM(F) .EQ. 0) THEN
          H(F)=0
             ELSE
         H(F)   R(F) / DENOM(F)
C          WRITE(6,*)H(F)
             END IF
40      CONTINUE

C  UNWRAPS AND GRAPHS THE WIENER FILTER IN THE FREQUENCY DOMAIN
         DO 50 F-1,N
           T(F)=H(F)
50       CONTINUE
         CALL BERCAN(G,N,T)

         DO 100 F=1,N/2
           XAXIS(F)=(F)*FRQ
           YAXIS(F)= ABS(T(F+N/2+1))
             IF (YAXIS(F) .GT. YMAX) THEN
                YMAX = YAXIS(F)
              ENDIF
                   WRITE(99,*) XAXIS(F),YAXIS(F)
100      CONTINUE
```

```
          IF (YMAX .LT. 300) YHLD =300.0
          IF (YMAX .LT. 200) YHLD =200.0
          IF (YMAX .LT. 100.0) YHLD -100.0
          IF (YMAX .LT. 40.0) YHLD =40.0
          IF(YMAX .LT. 15.0) YHLD =15.0
          IF(YMAX .LT. 5.0) YHLD =5.0
      MRKS = YHLD/5.0



      CALL DEVICE
      CALL HEADIN('WIENER FILTER   H(F)$',100,1.5,1)
      CALL XNAME('SPATIAL FREQUENCY$',100)
      CALL YNAME('SPREAD DISTRIBUTION$',100)
C     CALL GRAF (0.0,20.0,160.0,0.0,MRKS,YHLD)
      CALL XLOG (1.0,2.5,0,MRKS)
      CALL SETCLR ('RED')
      CALL CURVE(XAXIS,YAXIS,127,0)
      CALL ENDPL(0)
      CALL DONEPL


C   OUTPUTS H(F) WIENER FILTER TO LOG 007

      WRITE(7.*) (H(F),F=1,N)

      WRITE(6,*) '
     C
     C                                  PROGRAM COMPLETED.'
      WRITE (6,*) 'WIENER FILTER IS IN WIENER DATA'
       END
```

```
C                    SYSTEM.FOR BY JAY H BERMAN

C  THIS PROGRAM READS IN AN OBJECT FUNCTION,PASSES THROUGH A
C  LENS,ADDS RANDOM COS*NOISE TO THE IMAGE,THEN FILTERS OUT
C  THE NOISE TO RECONSTRUCT THE ORIGINAL IMAGE.

C   THE FOLLOWING MATHEMATICAL OPERATIONS WILL BE PERFORMED:

C  O(X)  ==| O(F)*L(F)-IM(F)  ==| IM(X)+NO(X)  ==| NOIM(F)*H(F)
C     ¶         ¶
C     O(F)  ==| O(X)

C   PROGRAM REQUIRES THE FOLLOWING LOGICAL ASSIGNMENT
C   ------------------------------------------------
C       O(X)             LOG 010       INPUT
C       L(F)             LOG 011       INPUT
C       NO(X)            LOG 012       INPUT
C       H(X)             LOG 013       INPUT
C       ¶
C       O(X)             LOG 007       OUTPUT
C       IM(X)            LOG 008       OUTPUT
C       IM(X)+NO(X)      LOG 009       OUTPUT

C   SYMBOLS
C   -------
C   O(X)=INPUT OBJECT;        O(F) ITS TRANSFORM
C   L(F)=FILM'S MTF
C   IM(X)=IMAGE FUNCTION;     IM(F) ITS TRANSFORM
C   NO(X)=NOISE FUNCTION;     NO(F) ITS TRANSFORM
C   NOIM(X)=NO(X) + IM(X);    NOIM(F)=NO(F)+IM(F)
C                             POWER(F)= NO(F)**2
C   H(X)=WEINER FUNCTION;     H(F) ITS TRANSFORM
C   ¶                                ¶
C   O(X)=RECONSTRUCTED OBJECT;       O(F)=ITS TRANSFORM

C  NOT ALL OF THESE SYMBOLS APPEAR IN THE ACTURAL PROGRAM, BUT
C  ARE USED IN  COMMENTS FOR EASE IN FOLLOWING THE PROGRAM FLOW.
```

```
C     ARRAY DECLARATIDNS
C     ------------------
        CDMPLEX  A(256), O(256), L(256), NO(256), NOIM(256),
      C   POWER(256), H(256), T(256), G(256)
        COMPLEX  NUM, NUMER, DENOM, DEN
        DIMENSION XAXIS(256), YAXIS(256)
        INTEGER   IWK(256), F, X
        CHARACTER*1  CHOICE, DOUBLE
        REAL    N

C     SET PARAMETERS
C     --------------
        N=256
        FQ=1.25
        NUM=(0.0,0.0)
        NUMER=(0.0,0.0)
        DENOM=(0.0,0.0)
        DEN=(0.0,0.0)
        SPC=320.0
                  WRITE(6,*) 'IS THIS DDUBLE SCALED 10?   Y/N'
                  READ (5,999) DOUBLE
                  IF(DOUBLE .EQ. 'Y') THEN
                    FQ=2.5
                    SPC=640.0
                  ENDIF


        WRITE(6,*)'IS THIS A NOISELESS SYSTEM ?   Y/N'
        READ (5,999)CHOICE


C    FOLLOWING READS D(X) INTO 'A' ==|O(F)
C    ----------------------------------
        READ (10,*)(A(I),I=1,N)

C  TRANSFORMS FROM SPACE TO FREQ
           CALL FAST(A,N,IWK,1)

        DO 10 F-1,N
           O(F)=A(F)
10      CONTINUE


C READS L(F),THEN O(F)*L(F)=IM(F) ==|IM(X)(RETURNS IN'A')
C ------------------------------------------------------
```

```
        READ (11,*)(L(F),F=1,N)
        DO 20 F=1,N
           A(F)=O(F)*L(F)
2D      CONTINUE

C 'TRANSFORMS FROM FREQ TO SPACE'
           CALL FAST (A,N,IWK,2)
C
C
C   SEND IM(X) TO LOG OOB OUTPUT

         DO  25 X=1,N
           WRITE(B,*) A(X)
25       CONTINUE


          IF(CHOICE .EQ. 'Y') THEN
C GRAPH OUTPUT OF BLURRED IMAGE
                 DO 26 I=1,N
                  T(I)=A(I)
26               CONTINUE
                 CALL BERCAN(G,N,T)
                 DO 27 I=1,N
                  XAXIS(I)=(I-129.0)/SPC
                  YAXIS(I)= (T(I))
27               CONTINUE
                 CALL DEVICE
                 CALL XNAME('MILLIMETERS$',100)
                 CALL YNAME ('RELATIVE SIGNAL AMPLITUDE$',100)
                 CALL HEADIN('TRIBAR IMAGE (BLURRED)$',100,1.5,1)
                       IF(DOUBLE .EQ. 'Y') THEN
                         CALL GRAF(-.20,.04,.20,-.4,.4,1.6)
                       ELSE
                 CALL GRAF(-.40,.08,.40,-.4,.4,1.6)
                       ENDIF
                 CALL SETCLR('RED')
                 CALL CURVE (XAXIS,YAXIS,256,0)
                 CALL ENDPL(0)
                 CALL DONEPL
          ELSE
            ENDIF

C        CALL DEVICE

C   READS IN NO(X); THEN IM(X)+NO(X) = A(X)   ==| NOIM(F)
C    ( NOIM(F) RETURNSS IN 'A')
C   ---------------------------------------------
      READ(12,*)(NO(X),X=1,N)
```

```
      DO 30 X=1,N
         A(X)=A(X)+NO(X)
30     CONTINUE

C  SEND IM(X)+NO(X) TO LOG 009 OUTPUT

         DO 35 X=1,N
           WRITE (9,*)A(X)
35       CONTINUE


         IF(CHOICE .EQ. 'N') THEN
C GRAPH OUTPUT OF NOISEY BLURREO IMAGE
                00 36 I=1,N
                 T(I)=A(I)
36              CONTINUE
                CALL BERCAN(G,N,T)


                00 37 I=1,N
                 XAXIS(I)=(I-129.0)/SPC
                 YAXIS(I)=REAL(T(I))
37              CONTINUE
                CALL OEVICE
                CALL XNAME('MILLIMETERS$',100)
                CALL YNAME ('RELATIVE SIGNAL AMPLITUOE$',100)
                CALL HEAOIN('NOISY BLURREO TRIBAR IMAGE$',
     C             100,1.5,1)
                      IF(OOUBLE .EQ. 'Y') THEN
                        CALL GRAF(-.20,.04,.20,-.4,.4,1.6)
                      ELSE
                CALL GRAF(-.40,.0B,.40,-.4,.4,1.6)
                      ENOIF
                CALL SETCLR('REO')
                CALL CURVE (XAXIS,YAXIS,256,0)
                CALL ENDPL(0)
                CALL DONEPL

        ELSE
        ENOIF
C 'TRANSFORMS FROM SPACE TO FREQ'
          CALL FAST(A,N,IWK,1)


C   MOVES 'A' (WHICH =THE FFT OF IM(X)+NO(X)) INTO NOIM(F)
C   ----------------------------------------------------------
       00 40 F=1,N
```

```
          NOIM(F)=A(F)
40      CONTINUE


C    GET POWER NO(F)**2, OETERMINE SIGNAL/NOISE RATIO (SNR)
C     BEFORE WIENER FILTERING.
C    -----------------------------------------------------

C   SNR IS DETERMINED BY FINDING THE INTENSITY (SIGNAL**2)
C   AREA UNOER THE CURVE AND OIVIOING IT BY THE AREA UNOER
C   THE CURVE FOR THE NOISE POWER SPECTRUM(N**2), THEN TAKING
C   ITS SQUARE ROOT.

              REAO(14,*) (POWER(F),F=1,N)

        DO 47 F=1,N
            NUM    O(F) * L(F) * FQ
          NUMER - NUMER + NUM
          OENOM - DENOM + ( POWER(F) * FQ)
47      CONTINUE

          IF (DENOM .EQ. 0)   THEN
              WRITE (6,*)'UNFILTEREO SNR (NOISELESS)= 1:0'
          ELSE
              SNR   (ABS (NUMER))**2 / DENOM
              WRITE (6,*)'UNFILTERED SNR =', SQRT(SNR), ':1'
          END IF

C   REINITIALIZE VARIABLES
          NUM=(0.0,0.0)
          NUMER=(0.0,0.0)
          DENOM=(0.0,0.0)
          OEN=(0.0,0.0)

C
C    READS WEINER FILTER H(F);   H(F)*NOIM(F)=O(F)
C    ------------------------------------------
        REAO(13,*)(H(F),F=1,N)

        DO 50 F=1,N
          A(F)=H(F)*NOIM(F)
50      CONTINUE


C    DETERMINE SNR AFTER WEINER FILTERING
C    ------------------------------------
        DO 51 F=1,N
            NUM - H(F) * O(F) * L(F) * FQ
```

```
          NUMER = NUMER + NUM
            DEN =  ABS(H(F))**2 * POWER(F) * FQ
          DENOM = DENOM + DEN
51      CONTINUE

           IF (DENOM .EQ. 0)   THEN
             WRITE(6,*)'WEINER FILTERED (NOISELESS) SNR = 1:0'
           ELSE
             SNR   NUMER**2 / DENOM
           WRITE(6,*)'WEINER FILTERED  SNR =',SQRT(SNR), ' :1'
           ENDIF

C                 ¶                    ¶       ¶     ¶
C    'A' CONTAINS O(F);NOW TRANSFORM O(F) ==|O(X) (O(X)
C       RETURNS IN 'A')
C    ------------------------------------------------------
C  'TRANSFORM FROM FREQ TO SPACE'
            CALL FAST (A,N,IWK,2)


C                          ¶
C    FOLLOWING OUTPUTS O(X) TO LOGICAL 007 AND GRAPHS IT'S
C      ABS VALUE
C    ------------------------------------------------------
         DO 60 X=1,N
           WRITE(7,*) A(X)
60       CONTINUE

        WRITE(6,*)'RECONSTRUCTED OBJECT IS ON LOG 007'


                    DO 80 I=1,N
                     T(I)=A(I)
80                  CONTINUE
                    CALL BERCAN (G,N,T)
                    DO 87 I=1,N
                     XAXIS(I)=(I-129.0)/SPC
                     YAXIS(I)= (T(I))
87                  CONTINUE
                    CALL DEVICE
                    CALL XNAME('MILLIMETERS$',100)
                    CALL YNAME ('RELATIVE SIGNAL AMPLITUDE$',100)
                    CALL HEADIN('RECONSTRUCTED TRIBAR TARGET VIA WIENER FILT
     CERING$',        100,1.5,1)
                         IF(DOUBLE .EQ. 'Y') THEN
                           CALL GRAF(-.20,.04,.20,-.4,.4,1.6)
                         ELSE
                    CALL GRAF(-.40,.08,.40,-.4,.4,1.6)
```

```
                  ENDIF
             CALL SETCLR('RED')
             CALL CURVE (XAXIS,YAXIS,256,0)
             CALL ENDPL(0)
999     FORMAT(A)
        END
```

```
C                          MASK.FOR BY JAY H BERMAN


C   THIS PROGRAM CREATES AN UNSHARP MASK FILTER TO BE USED IN
C   THE RECONSTRUCTION OF A DEGRADED IMAGE.  TO GET A FULL
C   UNDERSTANDING OF THE STEPS IN THIS PROGRAM READ THE SECTION
C   ABOUT MASKING IN THE THESIS INTRODUCTION.


C                  DECLARATIONS
C                  ------------

        COMPLEX A(256),RO(256),MASK(256),G(256),T(256),IMAGE(256),
     C          IMAGEF(256),TF(256),FILMTF(256)
        DIMENSION XAXIS(256),YAXIS(256),TFMASK(256),FINAL(256),
     C          UOTF(256)
        REAL   N
        COMPLEX MOMENT,B1,B2
        INTEGER IWK(256)
        COMPLEX HOLD, TEMP
             CHARACTER*1 DOUBLE
        N = 256
        MOMENT -(0.0,0.0)
        GAMA1 = 1.0
        GAMA2 = -.7
        GAMA3 = 1.0/.30
        PI = 3.1415927
        FRQ=1.25
        DX=1.0/320.0
              SPAC   320.0

C*************FOR DOUBLE SCALED TARGET**************
        WRITE (6,*)'ARE YOU USING A DOUBLE SCALED TARGET?   Y/N'
        READ (5,999) DOUBLE
999     FORMAT (A1)
        IF(DOUBLE .EQ. 'Y') THEN
           FRQ=2.5
           DX= 1.0/640.0
           SPAC = 640.0
        ENDIF


C****************TEST PUSHINNG THE MASK 5/B/B5*********
C                         GAMA1 = 1.0
C                         GAMA2 = -.96
C                         GAMA3 = 25.0
C*********************************
```

```
C    READS IN L(F) (MTF), SCALES IT THEN TRANSFORMS TO L(X).
C    L(X) IS USED TD FIND THE SECOND MOMENT.

          READ (7,*) (FILMTF(I), I=1,N)

          HDLD=FILMTF(1)
           DO 5 I=1,N
              A(I) - FILMTF(I)/HOLD
5          CONTINUE

C   TRANSFORMS FROM FREQ TO SPACE.
           CALL FAST (A,N,IWK,2)


C   FOLLOWING FINDS SECOND MOMENT. FUNCTION IS SYMETRICAL.

               MOMENT=0.0
               TEMP= A(1)*DX**2
          DO 10 X=1,N/2
             MOMENT=MOMENT +( A(X)*( ( X-1)*DX )**2 )
10        CONTINUE
             MOMENT= ABS(2*MOMENT + TEMP)
          WRITE (6,*) 'MOMENT=', MOMENT

C   DELTA FUNCTIONS.

C   FINDS B1 AND B2, SCALING CONSTANTS.

          B1=2*(PI**2)*MOMENT

          B2   B1*(-GAMA1/GAMA2)-B1

          WRITE (6,*) 'B1= ',B1
          WRITE (6,*) 'B2   ',B2


C   DETERMNINE THE OTF(UOTF) TO BLUR THE IMAGE WITH IN ORDER
C   TO PRODUCE THE UNSHARP MASK.
C   UOTF(F)IS MADE DIRECTLY IN FFT FORMAT.
         DO 21 F=1,N/2
           UOTF(F)=EXP(-B2*(F*FRQ)**2)
21       CONTINUE

         DO 22 F=(N/2)+1,N
           UOTF(F)=EXP(-B2*((F-N-1)*FRQ)**2)
22       CONTINUE
```

```
C****************************************************************
C   FOLLOWING CODING WAS USED TO SHOW A CORNSWEET EDGE AS
C   REQUESTED BY DR. BROUWER.
C   RUN THE PROGRAMS TWICE. FIRST TIME THRDUGH WRITE A NARROW
C   UOTF AND SAVE IT IN FILE 15 (SET 1 EXECUTED, SET 2 NOT ).
C   THIS BLURRING FUNCTION WILL BE THE UOTF USED WITH A WIDER
C   FILM MTF THE SECOND TIME THROUGH THE PROGRAM  (SET 2
C   EXECUTED, SET 1 NOT). THIS CAUSE THE SAVED UOTF PREVIOUSLY
C   CALCULATED TO WRITE OVER THE UOTF CALCULATED WITH THE ABOVE
C   CODING, WHICH IS USED TO DEMONSTRATE CONTRAST ENHANCEMENT.

C   SET 1
           OPEN(15)
           DO 23 F=1,N
             WRITE(15,*) UOTF(F)
23        CONTINUE
          CLOSE (15)

C   SET 2
C         OPEN(15)
C         DO 24 F=1,N
C           READ(15,*) UOTF(F)
C24        CONTINUE
C          CLOSE (15)
C****************************************************************


C         ****************GRAPHICS**************

                     DO 25 F=1,N/2
                      XAXIS(F)=F*FRQ
                      YAXIS(F)= UOTF(F)
25                   CONTINUE
                     CALL DEVICE
                     CALL XNAME('SPATIAL FREQUENCY$',100)
                     CALL YNAME('MODULATION$',100)
         CALL HEADIN('MTF YIELDING THE UNSHARP MASK$',
     C 100,1.5,1)
                     CALL XLOG(1.0,2.5,0.0,.25)
                     CALL SETCLR('RED')
                     CALL CURVE (XAXIS,YAXIS,128,0)
                     CALL ENDPL(0)
                     CALL DONEPL
                     CLOSE (16)


C READS IN IMAGE(NOISYIM) INTO A(),TRANSFORMS IT,THEN CONVOLVES
C IT WITH THE SPREAD FUNCTION UOTF, TRANSFORMS AGAIN THEN
```

```
C MULTIPLIES IT WITH THE SCALING FUNCTION GAMA2 TO MAKE THE
C UNSHARP MASK.

        READ(11,*) (IMAGE(I),I=1,N)
        DO 26 I-1,N
           A(I)=IMAGE(I)
26         CONTINUE
C   TRANFORM TO FREQ.
        CALL FAST(A,N,IWK,1)
        DO 27 F=1,N
           IMAGEF(F)=A(F)
27      CONTINUE


        DO 40 F=1,N
         A(F)=IMAGEF(F)*UOTF(F)
40      CONTINUE

C  TRANSFORM BACK TO SPACE, SCALES BY GAMMA2, AND UNWRAP FOR
C  GRAPHICS.
        CALL FAST(A,N,IWK,2)
          DO 45 X=1,N
           MASK(X)= A(X)*GAMA2
45        CONTINUE

C*****************************GRAPHICS*************************

                   DO 46 X=1,N
                   T(X)=MASK(X)
46                 CONTINUE
                   CALL BERCAN(G,N,T)
                   DO 47 I=1,N
                   XAXIS(I)=(I-129.0)/SPAC
                   YAXIS(I)  (T(I))
47                 CONTINUE

                    CALL DEVICE
                    CALL XNAME('MILLIMETERS$',100)
                    CALL YNAME ('RELATIVE SIGNAL AMPLITUDE$',100)
                    CALL HEADIN ('UNSHARP MASK$',100,1.5,1)
         IF (DOUBLE .EQ. 'Y') THEN
         CALL GRAF (-.20,.040,.20,-1.6,.40,.4)
         ELSE
                    CALL GRAF (-.40,.080,.40,-1.6,.40,.4)
         ENDIF
                    CALL SETCLR('RED')
                    CALL CURVE (XAXIS,YAXIS,256,0)
                    CALL ENDPL(0)
```

```
                    CALL DONEPL
                    CLOSE (16)

C**************************************************************

C  RECONSTRUCTS OBJECT BY ADDING THE LOWER CONTRAST NEGATIVE
C  MASK WITH THE HIGHER CONTRAST POSITVE IMAGE.
C  SCALES ON 3.334 GAMA PAPER TO GET BACK TO ORIGINAL CONTRAST
C  THEN OUTPUTS RESTORED OBJECT.

        DO 50 X=1,N
           RO(X)=GAMA3*( (GAMA1*IMAGE(X)) + MASK(X) )
50      CONTINUE

        DO 55 I=1,N
          WRITE(8,*) RO(I)
55      CONTINUE

C***********************GRAPHICS***********************

C  UNWRAP FOR GRAPHIC OUTPUT.
                   DO 60 X=1,N
                   T(X)=RO(X)
60                 CONTINUE


                   CALL BERCAN(G,N,T)

                   DO 70 I=1,N
                   XAXIS(I)=(I-129.0)/SPAC
                   YAXIS(I)= (T(I))
70                 CONTINUE

              CALL DEVICE
              CALL XNAME('MILLIMETERS$',100)
              CALL YNAME('RELATIVE SIGNAL AMPLITUDE$',100)
              CALL HEADIN('RECONSTRUCTED OBJECT VIA UNSHARP MASK $',
     C             100,1.5,1)
         IF (DOUBLE .EQ. 'Y') THEN
         CALL GRAF (-.20,.040,.20,-.4,.40,1.6)
         ELSE
                    CALL GRAF (-.40,.080,.40,-.4,.40,1.6)
         ENDIF
              CALL SETCLR('RED')
              CALL CURVE (XAXIS,YAXIS,256,0)
              CALL ENDPL(0)
              CALL DONEPL
```

```
          CLOSE (16)

      DO 85 F=1,N
         TF(F)  GAMA2*UOTF(F)
85     CONTINUE

       DO 86 F=1,N
        TFMASK(F)    (TF(F))
86     CONTINUE
C*****************************************************




C***************************GRAPHICS*******************************
               DO 90 F=1,N
                 XAXIS(F)=F*FRQ
                 YAXIS(F)= TFMASK(F)
90             CONTINUE

          CALL DEVICE
          CALL XNAME('SPATIAL FREQUENCY$',100)
          CALL YNAME('RELATIVE SPREAD DISTRIBUTION$',100)
          CALL HEADIN('MASK TRANSFER FUNCTION$',100,1.5,1)
          CALL XLOG(1.0,2.5,-1.0,.25)
C         CALL GRAF(-160.0,40.0,160.0,0.0,5.0,30.0)
          CALL SETCLR('RED')
          CALL CURVE (XAXIS,YAXIS,128,0)
          CALL ENDPL(0)
          CALL DONEPL
          CLOSE (16)

C DETERMINE MTF ACHIEVED THROUGH CONTRAST ENHANCEMENT VIA THE
C UNSHARP MASK.
      DO 100 F=1,N
       FINAL(F)=  GAMA3*(FILMTF(F)* (GAMA1+TF(F) ) )

100   CONTINUE

C***************************GRAPHICS*******************************
               DO 110 F=1,N
                 XAXIS(F)=F*FRQ
                 YAXIS(F)= FINAL(F)
110               CONTINUE

          CALL DEVICE
          CALL XNAME('SPATIAL FREQUNCY$',100)
          CALL YNAME('MODULATION$',100)
      CALL HEADIN('SYSTEM MTF (CONTRAST ENHANCEMENT)$',100,1.5,1)
```

```
      CALL XLOG(1.0,2.5,0.0,.25)
      CALL SETCLR('RED')
      CALL CURVE (XAXIS,YAXIS,128,0)
      CALL ENDPL(0)
      CALL DONEPL
      CLOSE (16)
END
```

```
C                              MEANSQER
C                                 BY
C                           JAY H. BERMAN


C COMPARE THEOETICALLY PERFECT OBJECT WITH THE ENHANCED/
C RESTORED IMAGE.
C AND FINDS MEAN SQUARED ERROR BETWEEN THEM.


C       O(X)      ==  TRIBAR OBJECT
C       ROW(X)    ==  RECONSTRUCTED OBJECT (WIENER FILTERING)
C       ROU(X)    ==  RECONSTRUCTED OBJECT (UNSHARP MSKING)
C       NOISY(X)  ==  IMAGE BEING FILTERED (MAY OR MAY NOT BE NOISY)


        COMPLEX O(256),ROW(256),ROU(256),NOISY(256)
        REAL MSE,MSE2,MSE3
        N=256
        MSE   0.0
        MSE2=0.0
        MSE3   0.0
        SSE   0.0
        SSE2=0.0
        SSE3   0.0

        OPEN(1)
        OPEN(2)
        OPEN(3)
        OPEN(11)

        READ(1,*) (O(X),X=1,N)
        READ(2,*) (ROW(X),X=1,N)
        READ(3,*) (ROU(X),X=1,N)
        READ(11,*) (NOISY(X),X=1,N)

        CLOSE(1)
        CLOSE(2)
        CLOSE(3)
        CLOSE (11)

        DO 5 X=1,N
          ROW(X) = ABS(ROW(X))
          ROU(X) = ABS(ROU(X))
          NOISY(X) = ABS(NOISY(X))
 5      CONTINUE
```

```
C   OETERMINE RANGE TO OVER WHICH TO SUM THE SQUAREO ERRORS
C   IN OROER TO CALCULATE MEAN SQUAREO ERROR (MSE).

        WRITE(6,*)'WHICH TARGET ARE YOU RECONSRUCTING ?'
        WRITE(6,*)'THE VALUE INPUTTEO PERTAINS TO A SCALING FACTOR FOR
     C CALCULATING NUMBER OF POINTS.
     C
        WRITE(6,*)'1.  5 CYC/MM                 2. 10 CYC/MM'
        WRITE(6,*)'4. 20 CYC/MM                 B. 40 CYC/MM'
        WRITE(6,*)'
     C                            INPUT 1,2.4. OR 8.'
        REAO(5,*) RNG
        IRANGE=N/2/RNG


C FOLLOWING FINOS SUM SQUAREO ERROR

        OO 10 X=1,IRANGE
         SSE   SSE +(O(X)-ROW(X))**2
10      CONTINUE
        OO 15 X=N-IRANGE,N
         SSE = SSE +(O(X)-ROW(X))**2
15      CONTINUE
        MSE=SSE/(IRANGE*2)
        WRITE(6,*) 'MEAN SQUAREO ERROR VIA WIENER FILTERING == ',MSE
        WRITE(6,*) 'MEAN ERROR VIA WIENER FILTERING == ',SQRT(MSE)


        OO 20 X=1,IRANGE
         SSE2   SSE2 +(O(X)-ROU(X))**2
20      CONTINUE
        OO 25 X=N-IRANGE,N
         SSE2   SSE2 +(O(X)-ROU(X))**2
25      CONTINUE
        MSE2=SSE2/(IRANGE*2)
        WRITE(6,*) 'MEAN SQUAREO ERROR VIA UNSHARP MASKING == ',MSE2
        WRITE(6,*) 'MEAN ERROR VIA UNSHARP MASKING == ',SQRT(MSE2)

        OO 30 X=1,IRANGE
         SSE3   SSE3 +(O(X)-NOISY(X))**2
30      CONTINUE
        OO 35 X=N-IRANGE,N
         SSE3 = SSE3 +(O(X)-NOISY(X))**2
35      CONTINUE
        MSE3=SSE3/(IRANGE*2)
```

```
WRITE(6,*) 'MEAN SQUARED ERROR WITHOUT ANY FILTERING == ',MSE3
WRITE(6,*) 'MEAN ERROR WITHOUT ANY FILTERING == ',SQRT(MSE3)

RATIO = MSE2/MSE
WRITE(6,*) 'MSE(M)/MSE(W) == ',RATIO
END
```

```
C EOGE TRACE ANALYSIS BY JAY H BERMAN

        REAL Y
        OIMENSION Y(0:15)
        OIMENSION L(0:20)
        COMPLEX O,SUM,TOT,L
        OATA Y/0.0,0.0,0.0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0,1.0,1.0,1.0/
C       OATA Y/0.0,0.0,0.0,.2,.4,.6,.8,1.0,1.0/
        X=.01
        OO 10 I=0,15
        PRINT*,Y(I)
10      CONTINUE
        PRINT*,'F=          L(F)=                      MTF='
        OO 40 F=0,20
        TOT=(0.0,0.0)
        OO 25 I=0,14
        SUM=(0.0,0.0)
        C=COS(2*3.141*F*X*(I+.5))
        S=(-1)*SIN(2*3.141*F*X*(I+.5))
        O=CMPLX(C,S)
        SUM=(Y(I+1)-Y(I))*O
        TOT=TOT+SUM
25      CONTINUE
        Z=3.141*F*X
        IF (F .EQ. 0) GO TO 26
        GO TO 27
26      L(F)-1*TOT
        GO TO 28
27       L(F)=SIN(3.141*F*X)/Z*TOT
28       WRITE (*,30)F,L(F),ABS(L(F))
30       FORMAT (F7.2,6X,(F7.4,4X,F7.4),4X,(F7.4,F7.4))
40       CONTINUE
         STOP
         ENO
```

```
C               CONVOL FORTRAN BY JAY H BERMAN

C  THIS PROGRAM ALLOWS THE USER TO INPUT A FUNCTION ANO A FILTER
C  IN THE FORM OF A SERIES OF DELTA FUNCTIONS TO BE CONVOLVEO WITH
C  EACH OTHER. THE PROGRAM PROVIOES THE OPTION OF CONVOLVING A CANNEO
C  FUNCTION WITH A CANNEO HIGH PASS OR LOW PASS FILTER AS A TOTORIAL.
C  THE PROGRAM ALLOWS INPUT OF UP TO AN 11 POINT FILTER (SPREAO FUNCT)
C  ANO UP TO A 65 POINT FUNCTION WHICH CAN BE KEYEO IN FROM THE TERMINAL
C  REAO IN FROM UNIT 1.


      OIMENSION F(-38:38), H(-70:70), G(-38:38)
      OIMENSION YAXIS(-38:38), XAXIS(-38:38), YAXIS2(-20:20),
     C  XAXIS2(-20:20),F2(-20:20),H2(-20:20)
       CHARACTER *1 CANNEO

             OO 1 I=-38,38
               XAXIS(I)=I
1            CONTINUE
             OO 2 I=-20,20
               XAXIS2(I)=I
2            CONTINUE
      PRINT*.'CONVOLUTIONS BY JAY H. BERMAN'
      PRINT*.'    '
      PRINT*,'    '

      WRITE(6,*) 'OO YOU WANT TO USE THE CANNEO HIGH OR LOW PASS FILT
     CER ROUTINE                      INPUT Y OR N.'
      REAO (5,99)CANNEO
         IF (CANNEO .EQ. 'Y') GOTO 35

       WRITE(6.*)'HOW MANY NUMBERS OO YOU WISH TO INPUT?  65 PTS MAX'
       REAO (5,*) INUMS
          NUMS= (INUMS/2)
          WRITE(6,*)'INPUT RANGE =',-NUMS,'   :',NUMS
          WRITE (6,*)'                                     '
      WRITE (6,*)'WILL INPUT FUNCTION BE  1)INPUTTED BY KEYBOARO OR
     C  2)REAO IN FROM LOGICAL FILE 1 ?
     C                                                          '
      REAO(5,*)OEVIC
      IF (OEVIC .EQ. 2) THEN
        REAO (1,*)(F(I),I=-NUMS,NUMS)
      ELSE
        OO 10 I=-NUMS,NUMS
          PRINT *,' INPUT OELTA VALUES FOR FUNCTION F(I),FOR I=',I
```

```
            READ (5,*) F(I)
10        CONTINUE

          WRITE(6,*)'THE FOLLOWING IS THE KEYED IN FUNCTION TO BE FILTE
     CRED.'
          WRITE (6,*)(F(I), I=-NUMS,NUMS)
        ENDIF
           I=0
          WRITE(6,*)'                                              '

        DO 30 N=-5,5
          PRINT *,'INPUT VALUES OF H(N),FOR N=',N
          READ (5,*) H(N)
30      CONTINUE
        WRITE (6,*)'                                              '
        WRITE (6,*)'THE FOLLOWING IS YOUR FILTER TO BE CONVOLVED.'
        WRITE (6,*)(H(N), N=-5,5)
C USER INPUTTED VALUES ARE NOT SCALED.
        SCALER=1
        GO TO 40


35      NUMS=9

C FOLLOWING ARE THE VALUES FOR THE FUNCTION F(I). IT HAS 5 SHARP EDGES
C AND 5 FLAT AREAS TO PERMIT DEMONSTRATION OF HIGH AND LOW PASS FILTERS.
        F(-9)=1
        F(-8)=2
        F(-7)=2
        F(-6)=-7
        F(-5)=3
        F(-4)=4
        F(-3)=4
        F(-2)=-5
        F(-1)=-5
        F(0)=-6
        F(1)=-5
        F(2)=5
        F(3)=-3
        F(4)=-3
        F(5)=-4
        F(6)=-4
        F(7)=6
        F(8)=-1
        F(9)=0
C FOLLOWING ARE THE HIGH AND LOW PASS FILTERS. THE DIFFERENCE BETWEEN
C THEM IS THE H(0) VALUE.
           H(-2)=1
```

```
            H(-1)=2
            H(1)=2
            H(2)=1
         WRITE(6,*)'OO YOU WANT THE 1) HIGH PASS OR  2)LOW PASS FILTER?'
         REAO(5,*) FILT
           IF (FILT .EQ. 1)THEN
            H(0)=-6
            SCALER=8
           ELSE
            H(0)=6
            SCALER=12
           ENOIF
C FOLLOWING COPIES H() INTO H2().
40         OO 41 I=-20,20
             H2(I)=H(I)
41         CONTINUE
C FOR HIGH PASS SCALER IS ARBTRARILY CHOSSEN TO PUT ON GRAPHICAL SCALE
C SIMILIAR TO THE ORIGINIAL FUNCTION FOR EASE OF COMPARISON. FOR LOW
C PASS FILTER SCALER IS THE AREA UNOER THE CURVE FOR H(). SCALER=1 FOR
C USER'S RESULTANT FUNCTION (OUTPUT OF G()IS UNSCALEO).  USER'S RESULTAN
C GRAPH WILL BE SCALEO UPON OUTPUT.

C FOLLOWING PERFORMS THE ACTUAL CONVOLUTION.
           N=0
        PRINT*,'N=                  G(N)='
       OO 50 N=-NUMS-6,NUMS+6
         G(N)=0
         OO 45 I=-NUMS,NUMS
              G(N)=G(N)+F(I)*H(N-I)
45         CONTINUE
        WRITE (6,100) N,G(N)/SCALER
50      CONTINUE

C FOR GRAPHICS THE USER'S RESULTANT CONVOLUTION WILL BE SCALEO TO -8,8
C ANO SET UP THE AXISES.
       IF (CANNEO .EQ. 'N') THEN
          MAX=0
          OO 60 N=-NUMS-6,NUMS+6
            IF( ABS(G(N)) .GE. MAX ) MAX=G(N)
60        CONTINUE
          SCALER=MAX/8
          SCALER= INT(SCALER+1)
          WRITE(6,*) '        SCALER=',SCALER,'
     C                                                  '
         ELSE
       ENOIF

       IF (NUMS .LE. 13) THEN
```

```
          DO 70 N=-NUMS-6,NUMS+6
            XAXIS2(N)=N
            YAXIS2(N)=G(N)/SCALER
            F2(N)=F(N)
70        CONTINUE
       ELSE
          DO 71 N=-NUMS-6,NUMS+6
            XAXIS(N)=N
            YAXIS(N)=G(N)/SCALER
71        CONTINUE
       ENDIF
C GRAPH RESULTANT CONVOLUTION
         WRITE (6,*) 'PLOT SIZE CAN NOT BE ALTERED. ANY VALUES INPUTTED
     C WILL BE IGNORED.
     C                                                       '
        CALL DEVICE
                       CALL BGNPL (-1)
                  CALL PHYSOR(2.,.5)
                  CALL AREA2D (5,3)
        CALL XNAME ('N     SPATIAL DISTRIBUTION$',100)
        CALL YNAME ('G(N) INTENSITY DISTRIBUTION$',100)
        CALL MARKER (16)
        CALL SETCLR('WHITE')
        IF (NUMS .LE. 13) THEN
           CALL GRAF(-20.0,5.0,20.0,-8.0,4.0,8.0)
           CALL SETCLR('RED')
           CALL CURVE(XAXIS2,YAXIS2.41.0)
        ELSE
           CALL GRAF(-40.0,10.0,40.0,-8.0,4.0,8.0)
           CALL SETCLR('RED')
           CALL CURVE(XAXIS,YAXIS,77,0)
        ENDIF
                 CALL ENDGR(1)

C  BEGIN SUBPLOT 2.
                 CALL PHYSOR(2.,7.3)
                 CALL XNAME ('     $',100)
                 CALL YNAME ('F(I)  FUNCTION$',100)
                 CALL AREA2D (5,3)
        CALL HEADIN('CONVOLUTION: G(N)= INTERGRAL(F(I) * H(N-I))$',100,
     C1.25,1)
              CALL SETCLR('WHITE')
        IF (NUMS .LE. 13) THEN
           CALL GRAF(-20.0,5.0,20.0,-8.0,4.0,8.0)
           CALL SETCLR('RED')
           CALL CURVE(XAXIS2,F2.41,0)
        ELSE
                 CALL GRAF(-40.0,10.0,40.0,-8.0,4.0,8.0)
```

```
                    CALL SETCLR('REO')
                    CALL CURVE(XAXIS,F,77,0)
          ENOIF
                    CALL ENOGR (2)

C    WRITE TITLE ANO BEGIN SUBPLOT 3.
                    CALL PHYSOR(2.,3.9)
                    CALL YNAME ('H(N)    FILTER$',100)
                    CALL XNAME ('      $',100)
                    CALL AREA2O (5,3)
                    CALL SETCLR('WHITE')
          IF (NUMS .LE. 13) THEN
             CALL GRAF(-20.0,5.0,20.0,-8.0,4.0,B.0)
             CALL SETCLR('REO')
             CALL CURVE(XAXIS2,H2,41,0)
          ELSE
                    CALL GRAF(-40.0,10.0,40.0,-8.0,4.0,8.0)
                    CALL SETCLR('REO')
                    CALL CURVE(XAXIS,H,77,0)
          ENOIF
             CALL ENOGR(3)
          CALL ENOPL(0)
          CALL OONEPL
99         FORMAT(A)
100        FORMAT(1X,I3,10X,F7.3)
          ENO
```

VITA

The author of this paper is currently a Captain in the United States Air Force (USAF) attending the Rochester Institute of Technology on an Air Force Institute of Technology sponsored program.

Captain Jay H. Berman was born in Brooklyn, New York on May 30, 1956. After completing his Bachelor's degrees in Chemistry and Science at North Carolina State University he entered the USAF. Upon entering the military, he was trained to be a Communications Electronics Officer and spent three years at Strategic Air Command, Omaha, Nebraska in that capacity before being selected to attend the Rochester Institute of Technology to obtain a Master of Science degree in Imaging and Photographic Science for which this thesis was written as a partial fulfillment of.

In August, 1985 Captain Berman will be reassigned to the USAF Space Division in Los Angeles, California where he will be putting to use the education he has received courtesy of the USAF.