X-RAY ENHANCEMENT SOFTWARE DEVELOPMENT AND TEST

FINAL REPORT

INFORMATION SYSTEMS

JOHN F. KENNEDY SPACE CENTER, NASA

DATA ANALYSIS FACILITY

TI-CSD-22A

AUGUST 1978

National Aeronautics and
Space Administration

John F. Kennedy Space Center

NASA

TR-1622

X-RAY ENHANCEMENT SOFTWARE DEVELOPMENT AND TEST

FINAL REPORT

PREPARED BY:

Robert L. Butterfield
ROBERT L. BUTTERFIELD

TI-CSD-22A

SUBMITTED BY:

C. L. Dillon
C. L. DILLON

TI-CSD-22A

AUGUST 1978

JOHN F. KENNEDY SPACE CENTER, NASA

STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
TR-1622		

4. Title and Subtitle	5. Report Date
X-Ray Enhancement Software Development and Test — Final Report	August 1978
	6. Performing Organization Code
7. Author(s): Robert L. Butterfield	8. Performing Organization Report No.
9. Performing Organization Name and Address: Data Analysis Facility, TI-CSD-22A, Kennedy Space Center, FL 32899	10. Work Unit No.
	11. Contract or Grant No.
	13. Type of Report and Period Covered: Final
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Kennedy Space Center, FL 32899	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A repertoire of software to optimally analyze various X-ray imagery has been successfully developed. Computer techniques are presented to solve many common problems involved in nondestructive testing X-ray analysis.

17. Key Words	18. Distribution Statement
X-ray, Image Processing, Image Enhancement, Software, Nondestructive Testing, Computer Analysis	Unlimited Distribution Announce Star

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

I. Accomplishment of Technical Objectives

A. Develop digital filter programs using large pixel windows

This capability has been completely developed and implemented. The BAF1 and BAF2 programs are quite versatile and effective. Complete specifications are contained in the software description.

B. Develop geometric correction programs necessary for registration of X-ray images

This objective has not been met to date. It is anticipated that this program will be implemented during the continuing effort on the N-Ray/X-Ray Study.

C. Develop an operational image processing system capable of handling large image arrays

This capability has been achieved using an approach which analyzes sub-arrays of the entire array. Upon receipt of equipment from the N-ray/X-ray study, this capability will be extended farther and optimized on even larger arrays; without additional funding.

D. Develop more efficient data transfer routines to be used by existing image processing systems

This objective has been attained and realized by every program listed in the software description. Execution times of one-tenth (0.1) the previous execution time are typical.

II. Overall Assessment

Capability for X-ray enhancement at the KSC Data Analysis Facility has been significantly increased as a result of this study. Information, heretofore unattainable, can now be extracted from X-ray imagery used in KSC support. A repertoire of software to optimally analyze various X-ray imagery has been successfully developed. Further X-ray enhancement capability (e.g. geometric correction, remoting of processed image to user site) will exist after completion of the N-ray/X-ray study. X-ray imagery used in KSC launch and support activities can now be analyzed more rapidly, thoroughly and correctly.

III. X-ray Enhancement Procedures Manual

Goal I: Sharpen edges to improve visibility of image detail. Extract subtle edges which are obscured by gradual shading effects. Minimize the effects of some shading nonlinearities.

Tool: High-pass Filter

Method:

1. Digitize X-ray using Video Averaging to increase signal-to-noise ratio in the image.
2. Using the BAF2 digital filter, input an appropriate high-pass filter transfer function--suppressing the low frequencies.
3. Using Contrast Stretch (Piecewise Linear or Linear) re-distribute the gray levels in the filtered image for optimum presentation to the eye.
4. If excessive "graininess" exists in output image, use the Variable Low-pass Filter program to slightly attenuate the highest spatial frequencies.

Example: In an analysis of an X-ray of the Space Shuttle solid rocket motor, the High-pass Filter and High-Emphasis Filter successfully revealed more definitive information about defects.

Goal II: Deblurr an X-ray to recover edge information and detail.

Tool: High-Emphasis Filter

Method:

1. Digitize X-ray using Video averaging to increase signal-to-noise ratio in image.
2. Using the BAF2 digital filter, input an appropriate high-emphasis filter transfer function--boosting the high frequencies on the original image. Extent of deblurring is determined by: (a) the slope of the transfer function, first, and then (b) the size of the filter. Gain of the transfer function should be just below the saturation level.

Example: In an X-ray analysis of a pump, the High-Emphasis Filter successfully deblurred the X-ray. More detail was visible after the restoration. The initial blurr function was caused by an out-of-focus condition which resulted when the pump was not in the focal plane of the X-ray system.

Goal III: Remove random noise (e.g. graininess or salt and pepper effect) from an image to reveal information previously masked by noise.

Tool: Low-pass Filter

Method:
1. Digitize X-ray using Video averaging to minimize any random noise from the camera/digitizing system.
2. Using the BAF2 digital filter or Variable Low-pass Filter, input an appropriate low-pass filter transfer function--suppressing the high frequencies. Cutoff of the transfer function should be no lower than necessary to just reveal features of interest--further filtering will begin blurring the image unnecessarily.

Example: When enlarging a small area of many X-rays, film grain introduced random noise to the image. The Low-pass Filter successfully minimized this effect and revealed information previously undetectable. One particular example involved small cracks in a crane hook X-ray.

Goal IV: Extract only the edge information from an image to see if bounderies of features are correct. Display the different degrees of edges.

Tool: Nine Point Gradient

Method:

1. Digitize X-ray using Video averaging to increase signal-to-noise ratio in the image.
2. Using Linear Constrast Stretch, distribute image gray levels to fill the dynamic range of solid state memory (256 gray levels).
3. Apply the Nine Point Gradient program. Output gray levels are proportional to gradient (edge) magnitude of input image.

Example: In an analysis of an X-ray of a printed circuit board, the Nine Point Gradient successfully detected defects in the board and electrical components.

Goal V: To display small gray level variations using false colors for increased differentiation. To extract areas of similar gray levels using colors to represent each pattern.

Tool: Density Slicer

Method:
1. Digitize X-ray using Video Averaging to increase signal-to-noise ratio in the image.
2. Using Density Slicer in the automatic mode, set a cursor over the area of interest and specify the number of uniformly spaced intervals to be displayed in false colors. In the manual mode, input the gray level bounds for each interval. Each output false color represents one finite range of gray levels.

Example: In an analysis of metal test bar X-rays for a corrosion study, the Density Slicer was used to map the extent of corrosion growth quickly and effectively.

Goal VI: To improve the contrast of an X-ray, or any particular feature within an X-ray--yielding more information about the content of the X-ray. That is, to improve the display of small changes in gray levels so that they may be better differentiated from their surroundings.

Tool: Contrast Stretch (Linear and Piecewise Linear)

Method:
1. Digitize X-ray using Video Averaging to increase signal-to-noise ratio in the image.
2. In Linear mode, place cursor over area of interest, or input gray level bounds. The input gray level range will be expanded to fill the full dynamic range of memory (256 gray levels). In Piecewise Linear mode, enter break-point coordinates of the input/output gray level transfer function. Output gray levels are distributed according to the piecewise linear transfer function.

Example: As previously mentioned, the contrast stretch programs were used as a post-processing enhancement for most all X-ray enhancement investigations. Nearly every X-ray can benefit from improved contrast.

Goal VII: Minimize the effect of shading nonuniformity within an X-ray

Tool: Shading correction, Ramp Mask, Background Mask, Linear Combiner

Method:

1. Digitize X-ray using Video Averaging to increase signal-to-noise ratio in the image.
2. If a background mask is available, apply Shading Correction after the mask is registered.
3. If a mask is not available, use Ramp Mask (for a linear distortion) or Background Mask (for a nonlinear distortion) to generate a mask containing the nonuniformity in shading. For additive distortions, use Linear Combiner to subtract the mask from the original--effectively correcting the nonuniformity. For multiplicative distortions, use Shading Correction to correct the original by computing the ratio of the original to the mask.

Example: In a corrosion analysis of a Space Shuttle Thermal Protection System tile X-ray, the Ramp Mask and Linear Combiner programs effectively corrected a severe shading uniformity caused by the X-ray scanner system.

Goal VIII: Output enhanced X-rays on a high-resolution hard copy (film or half-tone image on line printer paper).

Tool: Gould Print (halftone), Micro-D Tape Write (film)

Method:

1. Use the Gould Print program to dump the enhanced image on the line printer. Single or double page resolution may be selected. Contrast Stretch may be necessary for best output on paper.
2. If film output is desired, use the Micro-D Tape Write program to dump the enhanced image onto magnetic tape. The tape may be input to the film printer of the micro-densitometer. A very high resolution transparency will be generated.

Example: These hard copy programs were used successfully on many X-rays.

Goal IX: Input precisely digitized X-rays from a microdensitometer for a more detailed analysis.

Tool: Micro-D Tape Read

Method:

1. Use microdensitometer to generate a magnetic tape containing the digitized X-ray. Quantization is very detailed: 256 gray level resolution and 12.5 micron spot size capability.
2. Use Micro-D Tape Read program to read the magnetic tape information into solid-state memory for further processing. X-Y geometric corrections may be performed.

Example: The Micro-D Tape Read program can be used for any X-ray of 228 X 228mm size. Improved spatial frequency quantization and gray level quantization will be realized.

IV. Brief Software Description

A. Previously existing software used in X-ray enhancement support

1. Densitometer

Purpose - To display the "density" (gray level) distribution of a chosen line for a given channel, field number, and raster line.

2. Video Scanner Averaging

Purpose - To reduce the noise (improve signal-to-noise ratio) caused by random processes in the television camera scanning system.

3. Digital Theme Print

Purpose - To reproduce a selected binary theme on the line printer/plotter.

4. Micro-D Tape Read/Write

Purpose - To input digitized image data from the high-resolution drum-scanning microdensitometer--enabling further processing by computer. Dumps data in Image 100 memory in a format readable by the microdensitometer--enabling film printing of the memory image.

5. Tape Registration

Purpose - Register a video image to another image or reference--enabling further processing of images on same spatial reference.

6. Shading Correction

Purpose - Correction of illumination nonuniformity over the field of view for better radiometric accuracy.

7. Single Cell Signature Acquisition

   Purpose - Determine the single cell signature from the statistical computations (frequency distribution-histograms) of the spectral intensity distribution within the training site.

8. Single Cell Histogram Display

   Purpose - Display histogram of single cell signature acquisition (# picture elements vs. gray level) allows modification of gray level limits.

9. Window Program

   Purpose - Expand (enlarge) a selected area by a factor of 2 to 9 and train on the area.

10. Ratio Mode

    Purpose - Reduce multiplicative errors in images.

11. Contrast Stretch

    Purpose - Maximize data range over available memory bits--for visual enhancement of data for viewing. Small gray level variations are made large.

12. Density Slicer

    Purpose - Examine the spectral limits for a given image and slice the range of gray level limits into a specified number of slices.

B. Software written as a result of the X-ray Enhancement Software Development and Test (attached)

1. <u>Nine Point Gradient</u>

   Purpose - To display the approximate magnitude of the gradient (rate of change in gray level) of an image for edge distribution analysis.

2. <u>Laplacian</u>

   Purpose - To display the Laplacian operator (second order derivative) of an image for edge enhancement--approximates high-pass filter over small areas.

3. <u>Variable Low-pass Filter</u>

   Purpose - To provide a variable degree of smoothing over an image by suppressing high frequency noise.

4. <u>Piecewise Linear Transfer Characteristic (Contrast Stretch)</u>

   Purpose - To display the contrast of gray levels in an image in the optimum manner, by allowing the user to construct any desired transfer characteristic.

5. <u>Gould Print of Full Screen</u>

   Purpose - To print a halftone reproduction of any image stored in memory.

6. <u>Background Mask Generator</u>

   Purpose - To aid in correction of illumination nonuniformities by generating a mask representing the distortion.

7. <u>Ramp Mask Generator</u>

   Purpose - To aid in correction of illumination nonuniformities by generating a ramp mask representing the piecewise linear distortion.

8. <u>Linear Combiner</u>

   Purpose - To allow synthesis of any linear combination of images for image enhancement.

9. <u>BAF Spatial Filter</u>

   Purpose - To allow flexible and efficient digital filtering for enhancement of an image's spatial frequency components.

Nine Point Gradient

PROGRAM SOURCES: GRAD.FTN, GRAD.MAC

PURPOSE: This program approximates the magnitude of the gradient of an image stored in one channel of video memory and outputs it to the same or another channel.

INPUTS: The user inputs the channel of the image for which the gradient is computed as well as the channel which receives the output image.

OUTPUTS: The computed gradient is output to the selected channel of video memory.

PROCESSING: Each element $\nabla_{i,k}$ of the output image matrix $\bar{\nabla}$ is computed according to the following algorithm. (Matrix $\bar{f}$ represents the image matrix for which the gradient magnitude is approximated.)

$$\begin{aligned}\nabla_{i,k} = & \left| f_{i,k} + 2f_{i,k+1} + f_{i,k+2} - f_{i+2,k} - 2f_{i+2,k+1} - f_{i+2,k+2} \right| \\ & + \left| f_{i,k} + 2f_{i+1,k} + f_{i+2,k} - f_{i,k+2} - 2f_{i+1,k+2} - f_{i+2,k+2} \right|\end{aligned}$$

The first quantity approximates the magnitude of the vertical partial derivative and the second quantity that of the horizontal partial derivative. To reduce computation time, these two magnitudes are summed rather than computing the Pythagorean distance of the two.

It should be observed that the $\bar{\nabla}$ matrix is shifted diagonally one pixel with respect to the input image matrix.

Nine Point Laplacian

PROGRAM SOURCES: LAPL. FTN, LAPL. MAC

PURPOSE: This program approximates the Laplacian operation on an image stored in one channel of video memory. The computed Laplacian image is placed in a selected channel of video memory.

INPUTS: The user enters the channel of the image for which the Laplacian is computed as well as the channel which receives the output image. Also, he enters an integer gain factor between 0 and 7 inclusive which scales the output by that power of two.

OUTPUTS: The computed Laplacian is output to the selected channel of video memory.

PROCESSING: Let us represent the output image by a matrix $\bar{\nabla}^2$ and the input image by a matrix $\bar{f}$. The matrix $\bar{\nabla}^2$ is computed according to the following algorithm:

$$\nabla^2_{i,k} = 128 + 2^N * [8f_{i+1,k+1} - f_{i,k} - f_{i+1,k} - f_{i+2,k} - f_{i,k+1} - f_{i+2,k+1} - f_{i,k+2} - f_{i+1,k+2} - f_{i+2,k+2}]$$

N is a user inputted scale factor between 0 and 7, inclusive. A bias of 128 is used so that positive and negative values of the quantity in brackets can be accommodated by the video memory.

It should be observed that the $\bar{\nabla}^2$ matrix is shifted diagonally one pixel with respect to the input image matrix.

Variable Lowpass Filter

PROGRAM SOURCES: VARLOW.FTN, VARLOW.MAC

PURPOSE: This program provides a variable degree of smoothing of an image.

INPUTS: The user inputs the channel of the image to be smoothed as well as the channel number on which to output the smoothed image. Also, an integer between 0 and 100 inclusive is input which represents the percentage of smoothing desired.

OUTPUTS: The smoothed data is output to the selected channel of video memory.

PROCESSING: Let us represent the output image by a matrix $\bar{S}$ and the input image by a matrix $\bar{f}$. The matrix $\bar{S}$ is computed according to the following algorithm:

$$S_{i,k} = \frac{100-N}{100} f_{i+1,k+1} + \frac{N}{100} \sum_{m=0}^{2} \sum_{n=0}^{2} f_{i+m,k+n}$$

N is the user inputted percentage of smoothing desired.

Observe that the $\bar{S}$ matrix is shifted diagonally one pixel with respect to the input image matrix $\bar{f}$.

PROGRAM NAME: PIECEWISE LINEAR TRANSFER CHARACTERISTIC

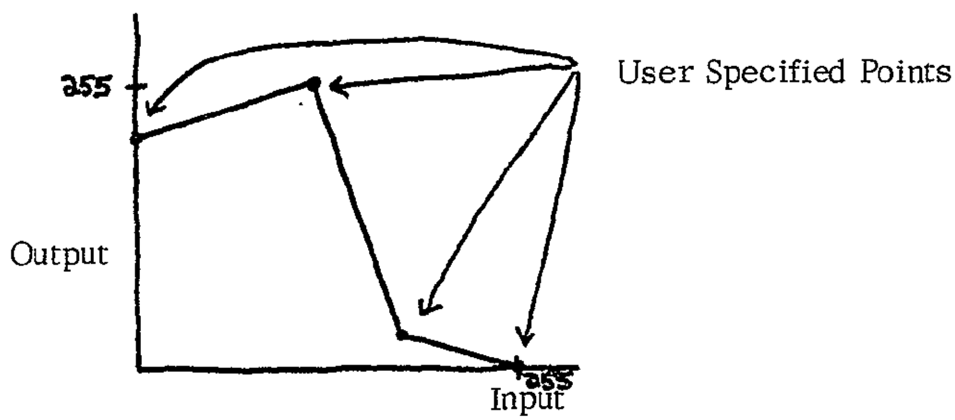
PURPOSE: This program allows the user to construct any desired transfer characteristic and apply it to any image in the video memory.

PROGRAM COMPONENTS: PLTC. FTN, PLTC. MAC

INPUTS: The user enters ordered pairs of numbers (integers between 0 and 255, inclusive) which correspond to points in the cartesian plane describing input vs. output gray levels. (Entry of a number out of the 0 to 255 range is interpreted as signalling the end of input.) Also, the user specifies the input channel to which the transfer characteristic is applied and the output channel of memory in which the result is placed.

OUTPUTS: The transfer characteristic as specified is drawn on the Tektronix CRT terminal. The image resulting from application of the transfer characteristic to the input image is placed in the selected channel of video memory.

PROCESSING: To describe the mechanism of the transfer characteristic, consider the following plot on a cartesian plane. The X axis represents the input gray level and the Y axis the output gray level. Moving in the direction of ascending values of x, the user specified points are connected with line segments to form the transfer characteristic. The output image is formed by replacing the pixel value specified on the X axis with the value specified on the Y axis.



PROGRAM NAME: GOULD PRINT OF FULL SCREEN

PURPOSE: This program prints a halftone reproduction of any of the video channels on the GOULD 5000 graphics line printer. Two sizes of output are available. Halftones are represented in a 4 by 4 dot matrix.
COMPONENTS: GOUPRT.FTN, GOUPRT.MAC, GOULD.MAC
(GOULD.MAC is a G.E. Image 100 supplied driver for the GOULD-5000.)

INPUTS - The user inputs the desired size of the output, single sheet or double sheet Also, he enters the channel of video to be printed.

OUTPUTS - The specified channel of video is printed on the GOULD 5000 line printer in either one or two sheet format, depending on the selected output size.

PROCESSING: Two stages of processing occur: formatting the video data to match the specified output size; conversion of the 8 bit gray level to the corresponding 4 by 4 dot matrix used to represent a halftone.

In the case of the single sheet output, the 512 by 512 pixel image matrix is reduced to 256 by 256 pixels by averaging clusters of four pixels. For the double sheet output, the 512 by 512 pixel matrix is split into two 256 by 512 pixel matrices, corresponding to the left and right halves of the image. Each of these halves is subsequently processed and output.

Each element of the 256 pixel line is converted to a 4 by 4 dot matrix. Sixteen distinct dot patterns are used to produce sixteen distinct gray levels on the output. The upper four bits of the eight bit pixel value select which of the sixteen dot patterns is to be output. Each 256 pixel line thus produces four 1024 element lines on the GOULD printer formatted such that each 4 by 4 cluster of dots represents a halftone.

PROGRAM NAME: BACKGROUND MASK GENERATOR

PURPOSE: Generates a mask by extending a horizontal or vertical profile of an image across an entire channel of video display memory.

COMPONENTS: BGDMSK.FTN

INPUTS: The user selects the input channel from which the mask is to be generated and the output channel on which the mask is to be placed. Also, he selects either a vertical or horizontal profile and positions the cursor to designate where the profile is to be taken.

OUTPUTS: The profile is extended and a mask generated on the specified output channel.

PROCESSING: A horizontal (vertical) line is read, the position of which is specified by the cursor. On the specified output channel this line is printed in every horizontal (vertical) line on the screen so that the mask is generated.

PROGRAM NAME: RAMP MASK GENERATOR

PURPOSE: Extracts a ramp function from a selected portion of a horizontal or vertical line of data and generates a mask by extending this ramp profile across the entire screen.

COMPONENTS: RMPMSK. FTN

INPUTS: The user selects the input and output channels and either a horizontal or vertical mask. In addition, he positions the cursor (crosshair mode) over the portion of the line (vertical or horizontal) that he wishes to have the ramp function generated from.

OUTPUTS: The calculated mask is output to the selected channel.

PROCESSING: The ramp function is calculated as follows:

$$\hat{y}_i = a(i-1) \quad \text{for } a \geq 0$$
$$\hat{y}_i = a(i-512) \quad \text{for } a < 0$$
$$1 \leq i \leq 512$$

where $\hat{y}_i$ is the i^{th} pixel of the computed ramp function. (Values for $\hat{y}_i$ greater than 255 are set equal to 255.) The slope of the ramp, a, is computed using least squares linear regression according to the following algorithm:

$$a = \frac{\sum_{i=m}^{n} y_i \, (i-\alpha)}{\beta - \alpha \Phi}$$

where: m and n are the boundary pixel numbers of the cursored portion of the selected input line.

y_i is the i^{th} pixel of the input line.

$$\Phi = \sum_{i=m}^{n} i = \tfrac{1}{2}(n+m)(n-m+1); \quad \beta = \sum_{i=m}^{n} i^2 = \tfrac{1}{6}\left[n(n+1)(2n+1) - m(m-1)(2m-1)\right]; \quad \alpha = \tfrac{1}{2}(n+m)$$

This algorithm for the slope a is derived from the general formula for fitting a straight line to a set of points using the least squares method:

$$a = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}$$

Set of points
$\{(x_i, y_i) \; i = 1, 2, \ldots n\}$

Once computed, this ramp function is printed in every horizontal or vertical line on the output channel (depending on whether a horizontal or vertical mask has been selected) so that the ramp mask is generated.

PROGRAM NAME: ONE TO FOUR CHANNEL LINEAR COMBINER

DESCRIPTION: This program allows the user to construct an image which is a linea combination of up to four other images. The user specifies how the combination is to be done.

PROGRAM COMPONENTS: LINCOM.FTN, LINCOM.MAC

INPUTS: The program requests the following information

(1) The number of channels to combine (1 to 4).

(2) Which channels to combine. (Any combination of the five video channels.)

(3) The weight for each channel. (A real number between -1.0 and 1.0, inclusive.)

(4) Gray level bias. (An integer between 0 and 255, inclusive.)

(5) Output channel. (Any of the five video channels.)

OUTPUTS: The combination specified is computed and output to the selected channel of the video memory.

PROCESSING: The output image is computed as follows:

$$\hat{f}_o = \hat{b} + \sum_{i=1}^{N} g_i \hat{f}_i$$

where:

$\hat{f}o$ is the output image matrix
N is the number of input channels combined
$\hat{f}_i$ is the i^{th} input image matrix
g_i is the assigned weight corresponding to the image $\hat{f}i$

b is a 512 by 512 matrix, each element of which is equal to the specified gray level bias.

1. Introduction

The BAF spatial filter program is supplied in two versions, BAF1 and BAF2. The two versions differ only in the mechanism for inputting filter characteristics, as described in section 1.2.2.2.

1.1 Filter Type

The BAF spatial filter operates as a convolution type filter. The operation is that of a running weighted summation.

Fig. 1 demonstrates the operation of the running weighted summation. Assume we have selected the summation to be done over a 3 by 3 pixel window. The first part of Fig. 1 represents this window by a rectangle enclosing a 3 by 3 pixel area of the original image. The second part of the figure represents the weighted summation, where each pixel in the window is multiplied by the corresponding element of the filter weight matrix and these products summed, producing the output pixel. The last part of Fig. 1 shows the position of this pixel in the output image. Each pixel of the output image is computed in this way, that is, centering the window over the corresponding pixel of the input image and performing the weighted summation.

1.2 Filter Design

The filter weight matrix is designed by the user for each run of the filter. The user specifies the size of the matrix (and the corresponding window) and supplies the specifications for the filter weights.

1.2.1 Filter Size

The filter matrix is a square matrix of dimension N. The value of N is selected by the user and can be any odd integer between 3 and 19, inclusive. Thus, the maximum filter size is a 19 by 19 filter weight matrix.

1.2.2 Filter Weights

Radial symmetry about the center filter weight is imposed so as to prevent phase shift in the output image. This symmetry is illustrated in Fig. 2 for a 5 by 5 matrix, in which weights with the same number are forced to be equal. The user has the choice of inputting the spatial filter weights directly or inputting weights describing the frequency response of the filter.

1.2.2.1 Spatial Filter Weights

In version BAF1, the user can assign values to the filter weights directly. Fig. 3 illustrates the assignment procedure for a 5 by 5 filter matrix.

The program will request the weights by the X and Y coordinates, which correspond to the numbered pixels in the figure. (The values entered must include a decimal point.) The remainder of the filter weight matrix will be assigned values according to the symmetry described above.

1.2.2.2 Frequency Domain Input

Rather than assign the filter weights directly, the user can describe the frequency response desired and the appropriate filter matrix will be computed. Each of the two versions, BAF1 and BAF2, provides its own method of specifying the frequency response.

1.2.2.2.1 BAF1 Frequency Domain Description

In BAF1, the user specifies the elements of the $\overline{G}$ matrix defined in the transform pair (3.4) of the appendix. Imposing radial symmetry in the $\overline{G}$ matrix ensures radial symmetry in the $\overline{g}$ matrix, as is desired. Fig. 4 illustrates the assignment of values to the $\overline{G}$ matrix. The program will request the values by the horizontal and vertical frequency components, which correspond to the elements numbered in the figure. (Decimal point must be included in input values.) The remainder of the $\overline{G}$ matrix will be assigned values according to the symmetry of Fig. 2. The $\overline{g}$ matrix will be computed according to the transform (3.4) of the appendix, resulting in a filter matrix with the frequency response as specified.

1.2.2.2.2 BAF2 Frequency Response Description

Often one is concerned only with radial frequency response. BAF2 enables the user to specify a radial frequency response curve. The program will sample this curve to obtain values for the $\overline{G}$ matrix which will then be processed as described for BAF1 above.

The curve produced for the radial frequency response is segmented. The user specifies the endpoints of the segments. Fig. 5 gives an example. In this case, the user has specified three points. To form the response curve, consecutive points are joined with line segments. (The program orders the points according to the x-axis value, i.e., radial frequency.) The program will sample this curve to obtain values for the $\overline{G}$ matrix.*

*Given u, the horizontal frequency, and v, the vertical frequency, the radial frequency is $\sqrt{u^2 + v^2}$.

2. Frequency Domain Characteristics

The relationship between a given filter matrix $\overline{g}$ and its spatial frequency response $\overline{G}$ is developed in the appendix and restated here:

$$G_{k,n} = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j\frac{2\pi}{N}(ak+bn)}$$

$$g_{a,b} = \frac{1}{N^2} \sum_{k=-M}^{M} \sum_{n=-M}^{M} G_{k,n}\, e^{-j\frac{2\pi}{N}(ak+bn)}$$

for $-M \leq a, b, k, n \leq M$ and $N = 2M+1$

Each element $G_{k,n}$ corresponds to the frequency response of the filter at the frequency $(\frac{k}{NT}, \frac{n}{NT})$, where N is the filter size (matrix dimensions) and T is the sampling period of the digitized image. We have imposed via radial symmetry that

$$G_{k,n} = G_{-k,n} = G_{-k,-n} = G_{k,-n}$$

and $$g_{a,b} = g_{-a,b} = g_{-a,-b} = g_{a,-b}$$

Observe that $^k/N$ corresponds to the horizontal digital frequency in cycles per sample and $^n/N$ to the vertical digital frequency.

2.1 Filter Size Considerations

The choice of size of the square filter matrix to be used has considerable effect on the type of frequency response curve that can be obtained.

The number of elements in the frequency response matrix $\overline{G}$ is equal to the number of elements in the corresponding fulter matrix $\overline{g}$. Thus for a filter dimension of N there are N^2 elements in each matrix. However, radial symmetry allows us to specify only 1/2 (M+1)(M+2) unique values in either matrix, where $M = \frac{N-1}{2}$. Increasing N thus increases the number of points we can specify in the frequency response curve of the filter.

The interval between adjacent points in the $\overline{G}$ frequency response matrix is $^1/NT$. In terms of digital frequency this is $^1/N$ cycles per sample. The highest digital frequency that can be specified horizontally or vertically is $M/N = \frac{N-1}{2N}$.

For N = 3 this frequency is $^1/3$; for N = 19, this frequency is $\frac{9}{19} = 0.474$. As N is made larger, this upper frequency specification approaches 1/2. This is in keeping with the Nyquist criteria which places the highest digital frequency component of a properly sampled signal at 1/2.

3. Method of Solution

Fig. 6. illustrates the procedure used in deriving the filter and performing the operation.

The user enters the size of the filter desired as well as the input and output channels of video memory. If the filter is to be described in the frequency domain, the $\overline{G}$ matrix is input, either directly in BAF1 or via the curve sampling technique of BAF2. The $\overline{G}$ matrix is transformed to the $\overline{g}$ matrix. If the filter is to be described in the spatial domain, the $\overline{g}$ matrix is input directly.

At this point, the floating point matrix $\overline{g}$ is scaled by an appropriate power of two to bring it into a range that can be represented with maximum precision by a 16 bit integer array. The scale factor is saved for later restoration and the scaled matrix $\overline{g}$ is fixed to 16 bits. A bias value is computed which when summed to the output of the filter operation places the expected value of the output at 128 so as to be compatible with the 8 bit video display memory. The filter operation is implemented with the fixed 16 bit representation of $\overline{g}$. 32 bit fixed arithmetic is used for the intermediate multiplications and summations so that no precision is lost in this stage. (Each pixel value is represented by 8 bits and the corresponding filter weight by 16 bits.) The fixed point output values from this operation are shifted right or left the proper number of times to compensate the scale factor. These values are each summed to the bias value to produce the values that are displayed. Negative display values are forced to 0; values greater than 255 are forced to 255. All output values are thus compatible with the video display memory.

FIG. 1

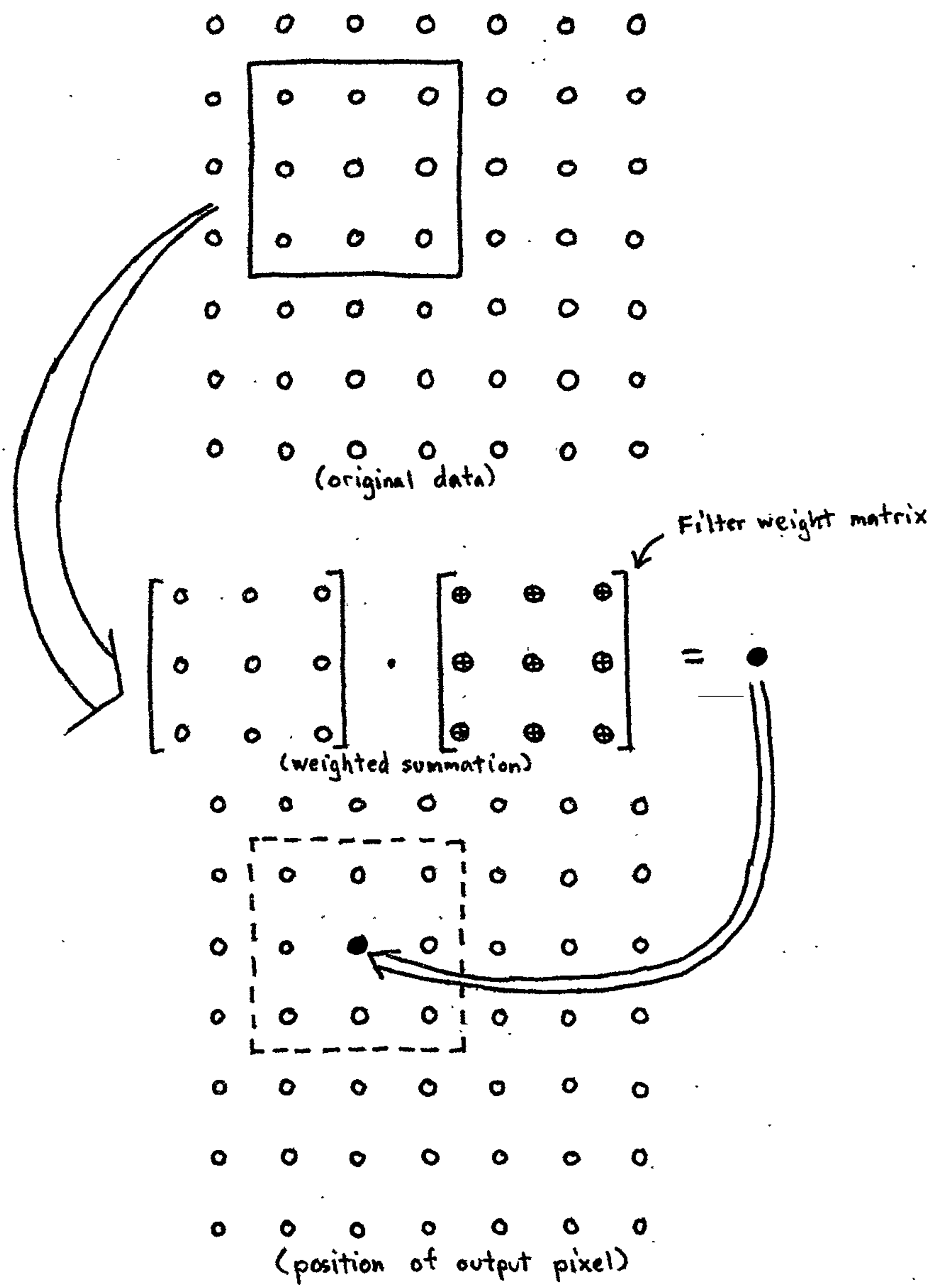

(original data)
Filter weight matrix
(weighted summation)
(position of output pixel)

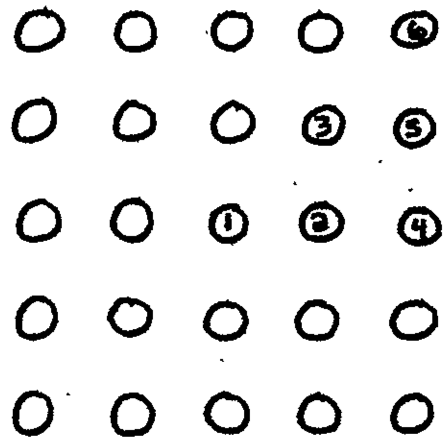
FIG. 2

6	5	4	5	6
5	3	2	3	5
4	2	1	2	4
5	3	2	3	5
6	5	4	5	6

RADIAL SYMMETRY ABOUT CENTER WEIGHT

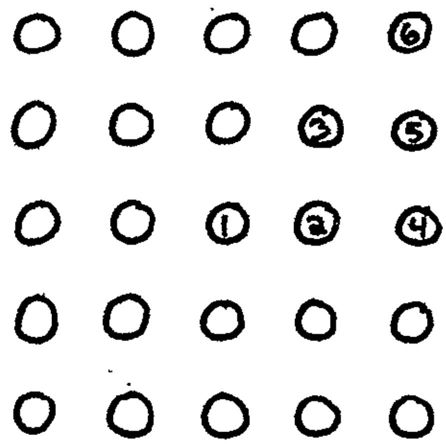
ORIGINAL PAGE IS
OF POOR QUALITY

FIG. 3



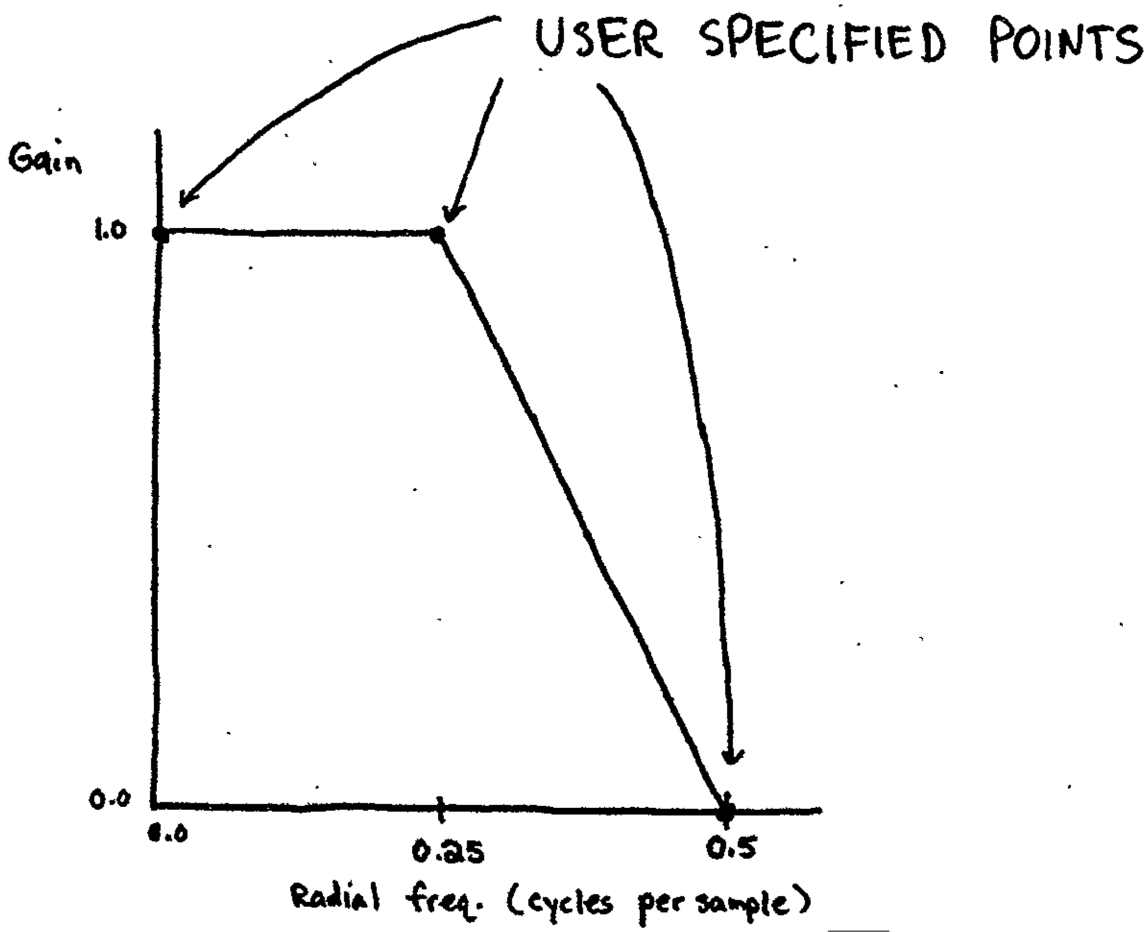
X	Y	PIXEL #
0	0	1
1	0	2
1	1	3
2	0	4
2	1	5
2	2	6

FIG. 4



Horizontal Freq.	Vertical Freq.	Element Number
.0000	.0000	1
.2000	.0000	2
.2000	.2000	3
.4000	.0000	4
.4000	.2000	5
.4000	.4000	6

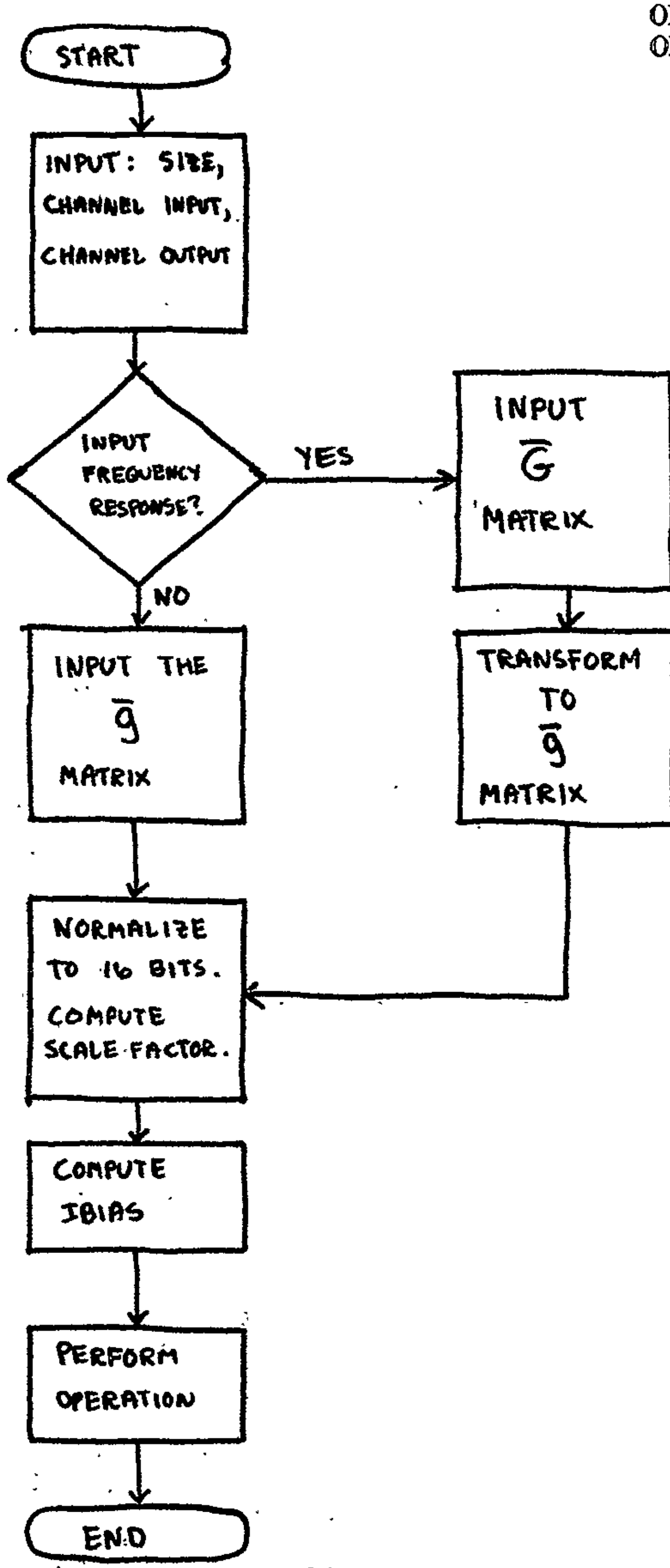
FIG. 5



Points entered:

Radial freq.	Gain
0.0	1.0
0.5	0.
0.25	1.0

FIG. 6


START
INPUT: SIZE, CHANNEL INPUT, CHANNEL OUTPUT
INPUT FREQUENCY RESPONSE?
YES
NO
INPUT $\overline{G}$ MATRIX
INPUT THE $\overline{g}$ MATRIX
TRANSFORM TO $\overline{g}$ MATRIX
NORMALIZE TO 16 BITS. COMPUTE SCALE FACTOR.
COMPUTE IBIAS
PERFORM OPERATION
END

APPENDIX

Mathematical developement of BAF spatial filter

I.

Let us define a two dimensional impulse function with properties analagous to those of the conventional single dimension impulse function. We define:

(1.1) (i) $\delta(x-x_0, y-y_0) = 0 \quad , x \neq x_0 \text{ or } y \neq y_0$

(ii) $\int_{x_1}^{x_2}\int_{y_1}^{y_2} \delta(x-x_0, y-y_0)\,dy\,dx = 1 ,$
$x_1 < x_0 < x_2$ and $y_1 < y_0 < y_2$

(iii) $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x-x_0, y-y_0)\,dy\,dx = f(x_0,y_0)$
for $f(x,y)$ continuous at (x_0,y_0).

Using this impulse function we can describe a sampled function of two variables, $h_s(x,y)$, such that

(1.2) $$h_s(x,y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(x,y)\,\delta(x-nT, y-mT)$$

where T is the sampling interval and $h(x,y)$ is the continuous function which is sampled.

The two dimensional Fourier transform of a function $f(x,y)$ is given by:

(1.3) $$\mathcal{F}\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi(ux+vy)}\, dy\, dx$$

The Fourier transform of our sampled signal is then:

$$\mathcal{F}\{h_s(x,y)\} = H_s(u,v)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_s(x,y)\, e^{-j2\pi(ux+vy)}\, dy\, dx$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(x,y)\,\delta(x-nT,\, y-mT)\right] e^{-j2\pi(ux+vy)}\, dy\, dx$$

$$= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\, e^{-j2\pi(ux+vy)}\,\delta(x-nT,\, y-mT)\, dy\, dx\right]$$

$$= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(nT,mT)\, e^{-j2\pi(unT+vmT)}$$

Thus, given a discretely sampled function $h_s(x,y)$, its (continuous) Fourier transform is given by:

(1.4) $$H_s(u,v) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(nT,mT)\, e^{-j2\pi(unT+vmT)}$$

II.

To describe the operation of the BAF filter, let us define several matrices:

$$\bar{f} = [f_{i,k}]_{512 \times 512}$$
$$\bar{g} = [g_{a,b}]_{N \times N}, \quad N \text{ odd}, \quad 3 \leq N \leq 19$$

Matrix $\bar{f}$ is the sampled image, 512 pixels horizontal by 512 pixels vertical.
Matrix $\bar{g}$ is the filter weight matrix, a square matrix of dimension N. We impose the restriction that N be an odd integer between 3 and 19, inclusive.

The filtering process is that of numerical convolution. Each element of the output matrix $\hat{f}$ is defined by:

$$(2.1) \quad \hat{f}_{i,k} = \sum_{a=1}^{N} \sum_{b=1}^{N} \left[(f_{i+a-p,\, k+b-p}) \, (g_{a,b}) \right], \quad p = \frac{N+1}{2}$$

[We shall say that $f_{i,k} = 0$ except for $1 \leq i \leq 512$ and $1 \leq k \leq 51$

For notational convenience, let us reassign the subscripts of $\bar{g}$.

Let $\bar{g} = [g_{a,b}]$
such that $-M \leq a \leq M$, $-M \leq b \leq M$
where $M = \frac{N-1}{2}$.

As an example, let $N=3$. Then $M=1$ and

$$\bar{g} = \begin{bmatrix} g_{-1,-1} & g_{0,-1} & g_{1,-1} \\ g_{-1,0} & g_{0,0} & g_{1,0} \\ g_{-1,1} & g_{0,1} & g_{1,1} \end{bmatrix} .$$

The filter operation of (2.1) becomes

$$(2.2) \qquad \hat{f}_{i,k} = \sum_{a=-M}^{M} \sum_{b=-M}^{M} f_{i+a,k+b} \, g_{a,b} \quad .$$

If we define a function $\hat{f}(x,y)$ such that

$$(2.3) \qquad \hat{f}(x,y) = \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \hat{f}_{i,k} \, \delta(x-iT, \, y-kT)$$

then from (1.4) the Fourier transform $\mathcal{F}\{\hat{f}(x,y)\}$ is given by

$$(2.4) \quad \hat{F}(u,v) = \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \hat{f}_{i,k} \, e^{-j2\pi(uiT+vkT)}$$

From (2.2) and (2.4):

(2.5)

$$\hat{F}(u,v) = \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \left[\sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, f_{i+a,k+b} \right] e^{-j2\pi(iuT + kvT)}$$

$$= \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \left[\sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, f_{i+a,k+b} \right] e^{-j2\pi T[u(i+a)+v(k+b)]}\, e^{-j2\pi T(-ua-vb)}$$

$$= \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j2\pi T(ua+vb)} \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f_{i+a,k+b}\, e^{-j2\pi T[u(i+a)+v(k+b)]}$$

$$= \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j2\pi T(ua+vb)} \sum_{i=1}^{512} \sum_{k=1}^{512} f_{i,k}\, e^{-j2\pi T(ui+vk)}$$

Now, if we define a function $f(x,y) = \sum_{i=1}^{512} \sum_{k=1}^{512} f_{i,k}\, \delta(x-iT, y-kT)$ then:

(2.6) $$\mathcal{F}\{f(x,y)\} = F(u,v) = \sum_{i=1}^{512} \sum_{k=1}^{512} f_{i,k}\, e^{-j2\pi T(ui+vk)}$$

Similarly, for $g(x,y) = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, \delta(x+aT, y+aT)$ then:

(2.7) $$\mathcal{F}\{g(x,y)\} = G(u,v) = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j2\pi T(au+bv)}$$

Notice that moving from left to right in the matrix $\bar{g}$ corresponds to moving in the direction of decreasing x for $g(x,y)$. Similarly, moving from top to bottom in matrix $\bar{g}$ corresponds to moving in the direction of decreasing y for $g(x,y)$.

From (2.5), (2.6), and (2.7) we thus conclude:

$$(2.8) \quad \hat{F}(u,v) = G(u,v) \cdot F(u,v)$$

That is, the spectrum of the output function $\hat{f}(x,y)$ is equal to the product of the spectrum of the input function $f(x,y)$ and the spectrum of the function $g(x,y)$. The frequency response of the filter described in (2.2) is thus $G(u,v)$ as given in (2.7).

III.

Consider the relationship

$$G(u,v) = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j2\pi T(ua+vb)}$$

In section II we establish that this expression is the (continuous) frequency response of the BAF filtering operation for a filter weight matrix $\bar{g}$.

Let us now evaluate $G(u,v)$ for certain values of u and v.

Let $u = \frac{k}{NT}$, $v = \frac{n}{NT}$ (recall that $N = 2M+1$).

Then $$G\left(\frac{k}{NT}, \frac{n}{NT}\right) = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j2\pi T\left(\frac{ak}{NT} + \frac{bn}{NT}\right)}$$

$$= \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j\frac{2\pi}{N}(ak+bn)}$$

Since $\frac{k}{NT} = \frac{k/N}{T}$, then $\frac{k}{N}$ represents digital frequency, that is, cycles per sample period. For a filter weight matrix $\bar{g} = [g_{a,b}]_{N \times N}$, then, the digital frequency response at intervals of $1/N$ is

$$(3.1) \qquad G_{k,n} = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j\frac{2\pi}{N}(ak+bn)}$$

Now,

(3.2) $$\sum_{k=-M}^{M}\sum_{n=-M}^{M} G_{k,n}\, e^{-j\frac{2\pi}{N}(ak+bn)}$$

$$=\sum_{k=-M}^{M}\sum_{n=-M}^{M}\left[\sum_{t=-M}^{M}\sum_{w=-M}^{M} g_{t,w}\, e^{j\frac{2\pi}{N}(tk+wn)}\right] e^{-j\frac{2\pi}{N}(ak+bn)}$$

$$=\sum_{t=-M}^{M}\sum_{w=-M}^{M} g_{t,w}\sum_{k=-M}^{M}\sum_{n=-M}^{M} e^{j\frac{2\pi}{N}(tk+wn)}\, e^{-j\frac{2\pi}{N}(ak+bn)}$$

Also,
$$\sum_{k=-M}^{M}\sum_{n=-M}^{M} e^{j\frac{2\pi}{N}(tk+wn)}\, e^{-j\frac{2\pi}{N}(ak+bn)}$$

$$=\sum_{k=-M}^{M}\sum_{n=-M}^{M} e^{j\frac{2\pi}{N}k(t-a)}\, e^{j\frac{2\pi}{N}n(w-b)}$$

$$=\sum_{k=-M}^{M} e^{j\frac{2\pi}{N}k(t-a)}\sum_{n=-M}^{M} e^{j\frac{2\pi}{N}n(w-b)}$$

(3.3) $$=\begin{cases} N^2, & t=a \text{ and } w=b \\ 0 & \text{otherwise}\end{cases}$$

From (3.1), (3.2), and (3.3) we conclude:

$$\sum_{k=-M}^{M}\sum_{n=-M}^{M} G_{k,n}\, e^{-j\frac{2\pi}{N}(ak+bn)} = N^2 g_{a,b}$$

We have therefore developed a transform pair:

(3.4) $g_{a,b} \Leftrightarrow G_{k,n}$ for $-M \leq a,b,k,n \leq M$

where $$G_{k,n} = \sum_{a=-M}^{M} \sum_{b=-M}^{M} g_{a,b}\, e^{j\frac{2\pi}{N}(ak+bn)}$$

$$g_{a,b} = \frac{1}{N^2} \sum_{k=-M}^{M} \sum_{n=-M}^{M} G_{k,n}\, e^{-j\frac{2\pi}{N}(ak+bn)}$$

$\overline{G}$ is an N by N matrix whose elements are samples of the frequency response of the filter described by algorithm (2.2). Each element $G_{k,n}$ corresponds to the frequency response of the filter at the frequency $(\frac{k}{NT}, \frac{n}{NT})$. The transform pair (3.4) enables us to compute the filter weights $\overline{g}$ from the sampled frequency response matrix $\overline{G}$, or compute the (sampled) frequency response matrix $\overline{G}$ from the filter weights $\overline{g}$.

Finally, we impose that:

$$g_{a,b} = g_{-a,b} = g_{a,-b} = g_{-a,-b} \quad ;$$

This symmetry about the center weight $g_{0,0}$ ensures that all elements of $\overline{G}$ will be real and of the same symmetry.