1-1-1981

Image processing for thermal images from an infrared camera.

Arthur Jacob Schenck

IMAGE PROCESSING FOR THERMAL IMAGES FROM AN INFRARED CAMERA

by

Arthur Jacob Schenck

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1981

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

This thesis is accepted and approved in partial
fulfillment of the requirements for the degree of Master of Science

April 9, 1981
(date)

Professor in Charge

Chairman of Department

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

LIST OF FIGURES

LIST OF FIGURES

ABSTRACT

This paper describes the application of computer image processing techniques to thermal images produced by an infrared temperature measurement camera. The camera was used to measure the temperature of materials inside an iron-producing blast furnace. The objective of computer processing of the images was to provide image enhancement - that is, improved contrast and reduced noise. Gray-level histogram calculations were performed and it was shown that contrast could be enhanced by manipulation (equalization) of the image histogram. Three types of 2-dimensional digital filtering techniques were demonstrated and applied to the thermal images. These were (1) convolution filtering, (2) discrete transform filtering (Fourier and Walsh-Hadamard) and (3) windowed finite impulse response filtering. For convolution filtering, both noise smoothing and edge enhancement masks were convolved with the original images. Even though both masks were kept relatively small to minimize computation time, they yielded very desirable results. High frequency noise was eliminated with the first mask, and edges (delineating zones of constant temperature) were defined with the second mask. Discrete transform filtering was performed by computing the 2-dimensional Fourier and Walsh-Hadamard transforms of the images, setting the smallest coefficients to zero, and then inverting the transforms to produce the filtered images. Although

the discrete filtered images contained more distortion than those filtered by convolution, this technique did reveal the potential for very efficient image coding. As many as 50 percent of the transform coefficients were eliminated without significantly affecting the quality of the transform filtered images. The best filtering results were produced with the more formal finite impulse response (FIR) technique. Both rectangular and Hanning windows were used to truncate the low pass and high pass filter impulse responses. The best results were obtained using the Hanning window, which resulted in the least distortion in the filtered images. The FIR technique has the important advantage that the filter characteristics can be controlled more precisely than with the other two techniques. A description is given of the software used to implement the image processing functions and program flowcharts are provided in the appendix.

INTRODUCTION

Image processing can be broadly defined to include four main functions - (1) image acquisition and correction, (2) image enhancement, (3) feature extraction, and (4) pattern recognition. This paper describes image enhancement in the context of an application where acquisition and correction are performed as preprocessing functions.

An infrared camera is used to measure the temperature of materials in an iron-producing blast furnace. The camera provides a two-dimensional temperature map (thermal image) of the material surface. These thermal images are sampled and quantized and are fed to a computer where they are corrected for geometric distortion and camera detector nonlinearity. The images are then classified based on several process variables and are stored for later viewing.

This paper describes the application of computer image enhancement techniques to the thermal images before they are classified and stored. The enhancement techniques provide (1) better contrast for easier interpretation, (2) noise removal for more accurate representation of temperatures, and (3) edge enhancement that could be used in automatic feature extraction. Each of the enhancement techniques described in the paper was implemented off-line using actual thermal images from the camera.

The paper is divided into four main sections. The Tutorial Description Section includes a detailed description of image sampling and interpolation, contrast enhancement, two-dimensional transforms, and two dimensional filtering. Thermal images and transform results are presented as 3-dimensional graphic plots.

The Software Section contains a detailed description of the computer programs used to implement the image processing functions. Program flowcharts are included in the appendix. The Application Section describes the infrared camera and data acquisition system, and describes the benefits provided by computer image processing. A method for incorporating the image enhancement techniques into an existing system is also provided in this section.

The Results Section reviews the usefulness of computer image enhancement and discusses the potential for future implementations in industrial remote sensing systems. It is assumed that the reader understands the fundamentals of one-dimensional signal processing techniques including sampling, convolution, Fourier transforms, and digital filtering.

Chapter 1

TUTORIAL DESCRIPTION OF INFRARED IMAGE PROCESSING

1.1 Image Sampling and Interpolation

The first step in any digital image processing application is the sampling and quantization of an analog image from a scanner. Figure 1.1 shows a 2-dimensional signal (image) overlayed with a 2-dimensional sampling grid, where x_1 and x_2 are the sample spacings in orthogonal directions. These spacings can be thought of as occuring in either the time or space domain. If the time domain is used, then the corresponding frequency domain units should be in radians per second. Similarly, if sampling is considered in terms of distance, or space between samples, then the spatial frequency domain should be used.

The two-dimensional sampling theory essentially states that a function of two variables $f(x_1, x_2)$ whose two dimensional Fourier transform is equal to zero for $W_1 > W_{1m}$ and $W_2 > W_{2m}$ is uniquely determined by the values taken at uniformly spaced sample points in the x_1, x_2 plane, if the sample spacings satisfy:

$$x_1 \leq \frac{\pi}{W_{1m}} \quad , \quad x_2 \leq \frac{\pi}{W_{2m}} \quad .$$

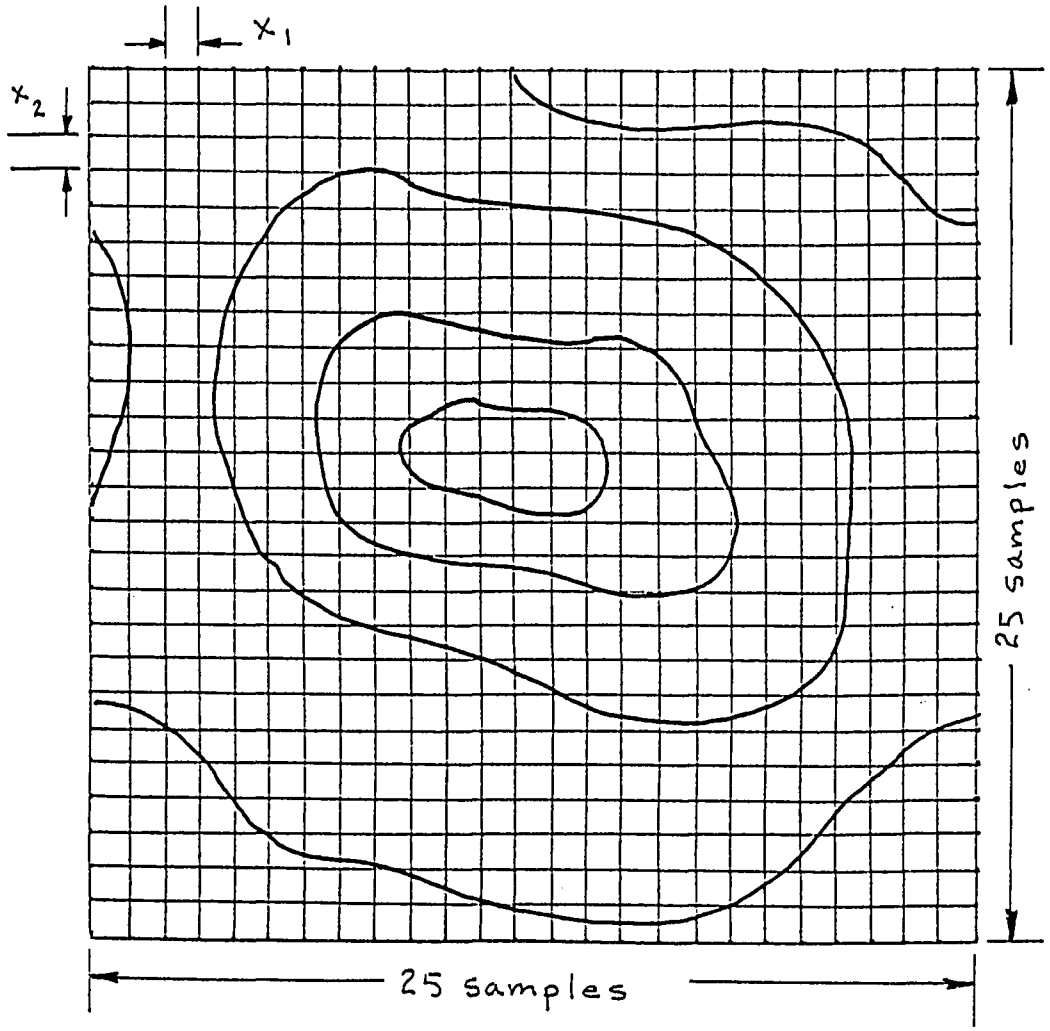


Figure 1.1 2-D Signal and Sampling Grid

Figure 1.2 shows a projection of the two-dimensional spectrum of the bandlimited signal in Figure 1.1. The crosshatched area is the region in the frequency plane containing non-zero samples. Notice that in two-dimensional systems, two frequency axes must be considered, corresponding to spatial frequency variations in both dimensions of the two dimensional signal.

The spectrum of the sampled signal $x(n_1x_1, n_2x_2)$ can be expressed as:

$$x_p(jw_1,jw_2) = \frac{1}{x_1}\frac{1}{x_2} \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} X(jw_1-jn_1w_{s1},jw_2-jn_2w_{s2})$$

where $X(jw_2,jw_2)$ is the continuous two-dimensional signal spectrum and:

$$w_{s1} = \frac{2\pi}{x_1} \quad , \quad w_{s2} = \frac{2\pi}{x_1}$$

As in the one dimensional case, the two dimensional sampled signal spectrum is periodically repeated, and is obtained by superimposing an infinite number of replicas of the continuous signal spectrum, each of which is centered at a point from the set k_1w_{s1}, k_2w_{s2}. Thus if $x(x_1,x_2)$ is bandlimited and it is sampled sufficiently fast, it is possible to recover the spectrum of the continuous two-dimensional signal, using the ideal two-dimensional low pass filter.

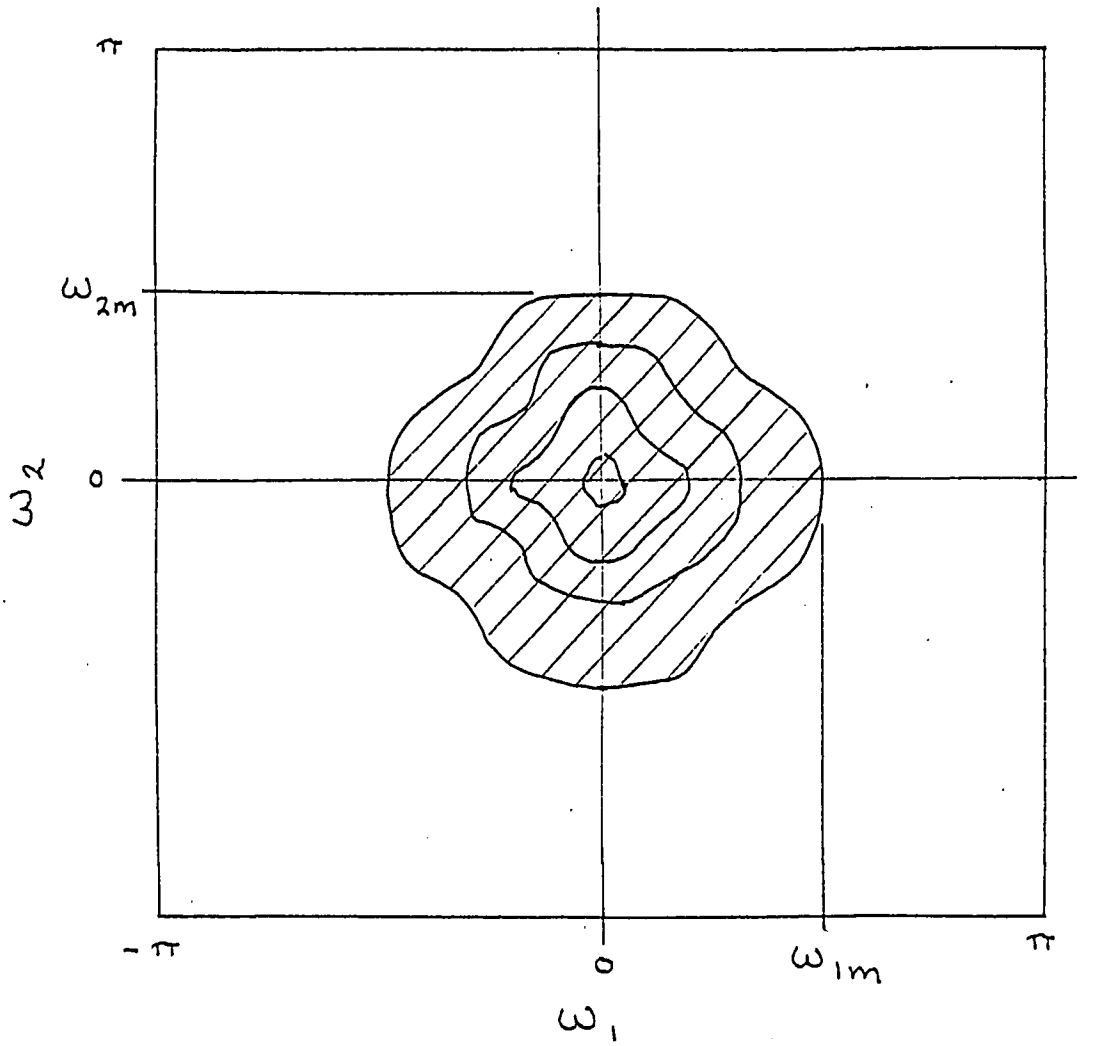


Figure 1.2 2-D Fourier Transform

The reconstruction low pass filter could have a transfer function equal to 1 in a rectangular region with dimensions W_{1m} and W_{2m},

$$H(jw_1,jw_2) = \text{rect}(j\frac{w_1}{w_{1m}})\ \text{rect}(j\frac{w_2}{w_{2m}})$$

where H is the filter transfer function and

$$\text{rect}(x) = \begin{cases} 1, \text{ for } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0, \text{ elsewhere} \end{cases}$$

The spectrum of $x(x_1,x_2)$ can be obtained by convolving the filter function with the periodic spectrum:

$$X(jw_1,jw_2) = X_p(jw_1,jw_2) \bullet H(jw_1,jw_2)$$

or in the space domain:

$$x(x_1,x_2) = x(n_1x_1,n_2x_2) * h(x_1,x_2)$$

where,

$$h(x_1,x_2) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \text{rect}\left(j\frac{w_1}{w_{2m}}\right)\ \text{rect}\left(j\frac{w_2}{w_{2m}}\right)\ e^{j(w_1x_1+w_2x_2)}dw_1dw_2$$

$$h(x_1,x_2) = \frac{1}{x_1}\ \frac{1}{x_2}\ \frac{\sin(w_{s1}x_1/2)}{w_{s1}x_1/2}\quad\frac{\sin(w_{s2}x_2/2)}{w_{s2}x_2/2}$$

Convolving this impulse response with the sampled two-dimensional signal,

$$x(x_1,x_2) = \sum_{n_1=-\infty}^{\infty}\ \sum_{n_2=-\infty}^{\infty} x(n_1x_1,n_2x_2)\frac{\sin(w_{s1}(x_1-n_1x_1)/2)}{w_{s1}(x_1-n_1x_1)/2}\cdot \frac{\sin(w_{s2}(x_2-n_2x_2)/2)}{w_{s2}(x_2-n_2x_2)/2}$$

which is the reconstruction relation for the two-dimensional case[1], i.e., the two dimensional form of the Sampling Theorem.

As will be described later, the infrared images from the blast furnace camera are essentially bandlimited, containing spatial frequencies that are relatively low. Thus, in practice, the two-dimensional sampling criterion was fairly easy to meet. Figure 1.3 is a 3-D plot of a sampled thermal image from the infrared camera. The sampling grid (N1,N2) is 25 by 25 points. The sample points are connected for a more pleasing presentation, and to create an effect similar to that which one would observe on a 2-D television type display through visual interpolation (smoothing).

ORIGINAL IR IMAGE

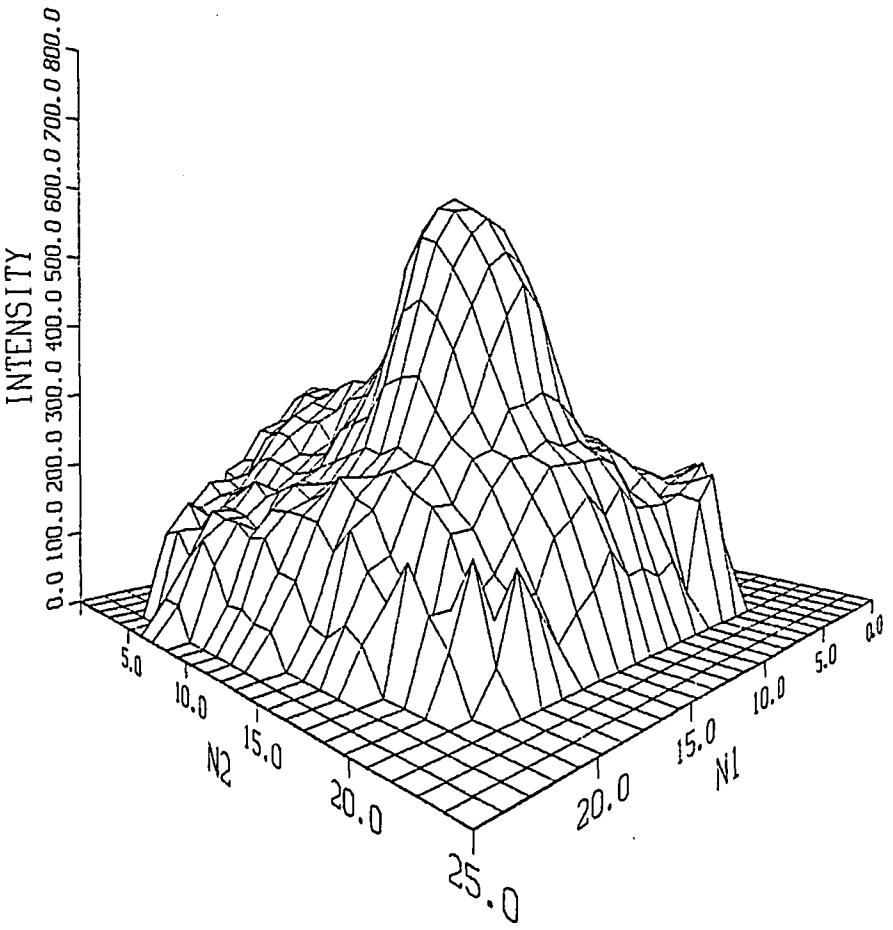


Figure 1.3 Original 25 X 25 Infrared Image

After sampling, each point is quantized with 8 bits, resulting in 256 discrete gray levels. Each of the gray levels represents a specific temperature. For convenience, temperature, rather than gray level, has been plotted on the vertical axis. With infrared images, heat intensity (temperature) is analagous to light intensity in visible light (vidicon) images. To keep the discussions general, image intensity, rather than temperture will be used throughout this section.

A 25 by 25 sampling grid results in a sampled image that appears relatively coarse. In most cases it is desirable to use more sampling points, if only to provide a more pleasing appearance. In addition, a 25 by 25 point function does not directly lend itself to many of the fast algorithms for computing transforms, since those algorithms require array dimensions that are an integer power of 2. The 25 by 25 point image would be more useful if it were transformed to a 32 by 32 point image. There are a number of techniques useful for this kind of interpolation. The simpler ones involve substituting an element in the new (more dense) array with the average of the spatially coincident points in the original array. This, however, can result in significant distortion in the resulting image, depending on how well points from the original image coincide with points in the resultant image. A technique that does not have this limitation is called bilinear interpolation.[2]

In bilinear interpolation, the original sampling grid is overlayed with the new one, and the exact spatial relationship between points in both grids is considered. A point in the new grid generally lies somewhere between 4 points in the original grid. Using the spatial relationship of the new point to the original ones, and the value (intensity) of the four original samples, the intensity of the new sample is computed.*

This is illustrated in Figure 1.4 where f(x,y) is the intensity of the new sample. A bilinear equation of the form:

$$f(x,y) = ax + by + cxy + d$$

is used in the computation. Using the four original points, the computation is carried out in three steps as follows:

(1) $f(x,0) = f(0,0) + x\,[f(1,0) - f(0,0)]$

(2) $f(x,1) = f(0,1) + x\,[f(1,1) - f(0,1)]$

(3) $f(x,y) = f(x,0) + y\,[f(x,1) - f(x,0)]$

Putting f(x,y) in the form of a bilinear equation:

$$f(x,y) = [f(1,0) - f(0,0)]\,x + [f(0,1) - f(0,0]y + [f(1,1) + f(0,0) - f(0,1) - f(1,0)]\,xy + f(0,0).$$

* The most accurate interpolation scheme, assuming there was no aliasing in the original sampling, would be to reconstruct a "nearly continuous" image using the reconstruction formula for the 2-D sampling theorem. This new image could then be resampled at the desired frequency, i.e., 32 by 32 points. This additional accuracy (which is computationally expensive) is difficult to justify for the infrared images since they contain mostly low frequency information.

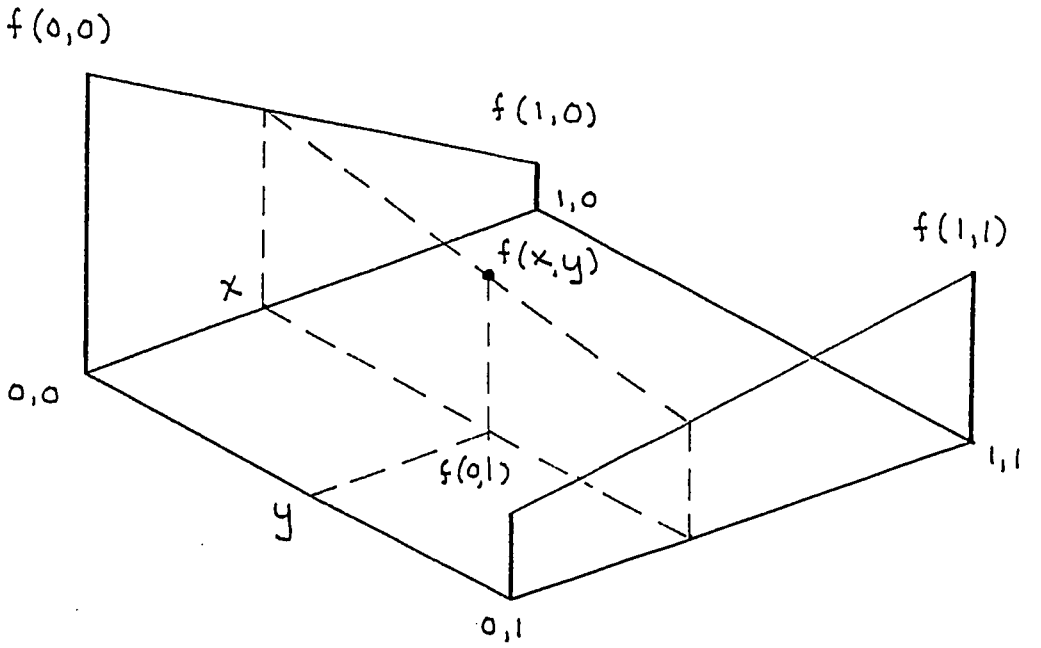


Figure 1.4 Bilinear Interpolation

Any interpolation process has the effect of smoothing (low pass filtering) the original data. However, because the image spatial frequencies are so low, this effect is minimal. Figure 1.5 shows the image in Figure 1.3 interpolated as a 32 by 32 point image. This new image will be used throughout the remainder of the paper in the computation of histograms and transforms.

1.2 Subjective Image Enhancement

1.2.1 Gray Level Histogram

The technique of "image enchancement" can involve many image processing functins, to meet a variety of goals. For example various types of digital filtering can be used to remove noise from images, or to enhance certain features such as edges. One of the most frequently desired objectives in image processing is contrast enhancement. This section describes a very useful technique to improve contrast called histogram equalization.

The gray level histogram of an image is defined as a function representing, for each gray level, the number of image samples having that gray level.[5] Gray level histograms are usually presented by plotting the number of pixels at each gray level against the full range of gray levels. Figures 1.6a and 1.6b show examples of gray level histograms having undersirable characteristics. Figure 1.6a shows a histogram from an image having relatively low contrast. This is reflected in the histogram as regions at both

IMAGE INTENSITY FUNCTION

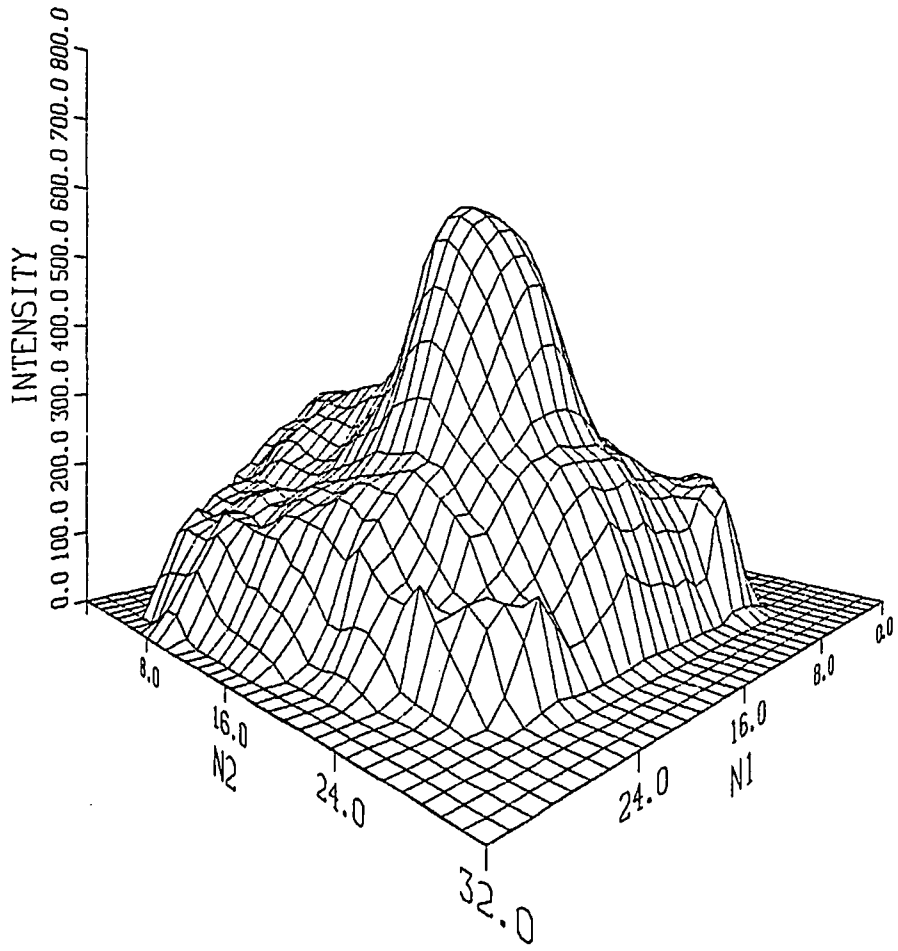


Figure 1.5 Interpolated 32 X 32 Infrared Image

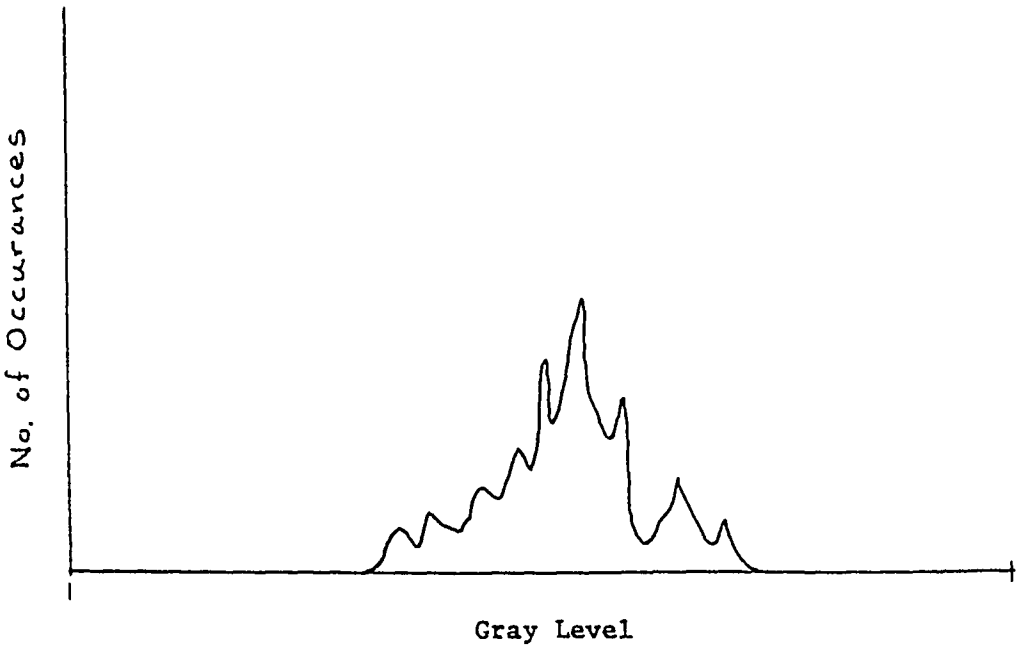


Figure 1.6 a Gray Level Histogram - Low Contrast

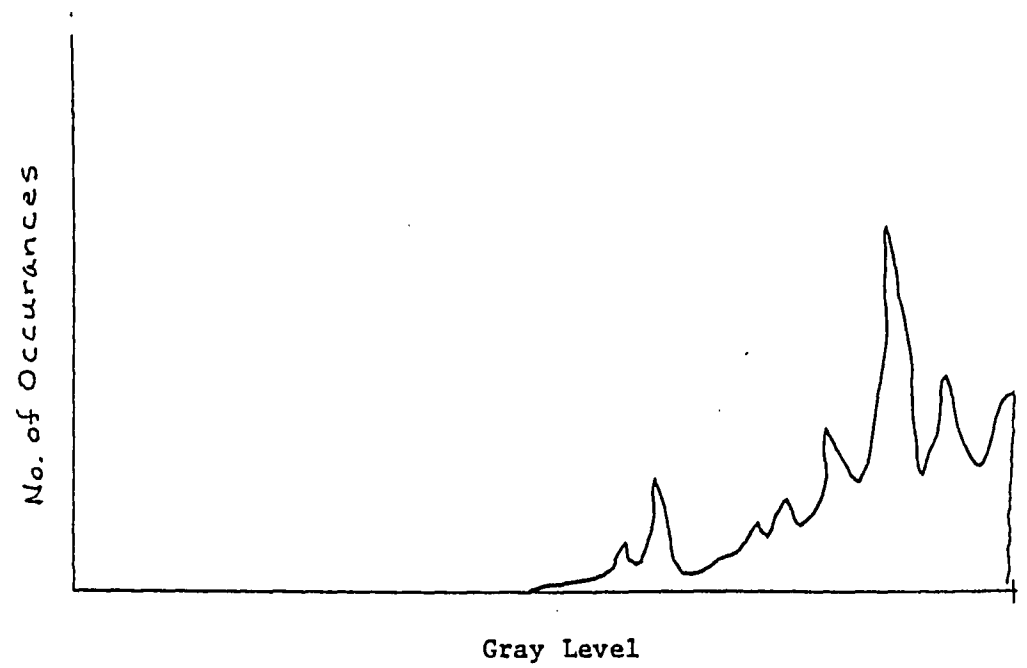


Figure 1.6 b Gray Level Histogram - Clipped

ends of the gray scale that have zero value. Thus the full dynamic range of gray level values was not utilized. In Figure 1.6b the histogram is clipped at the high end of the scale, again reducing the visible dynamic range. Both of these conditions could be the result of improper adjustment of the image quantization electronics. Hence the image histogram can provide information, not only on the images themselves, but also on the image sampling equipment. However, assuming the equipment is operating properly, the overall "shape" of the histogram is a function of the image characteristics. It is often desirable, in this case, to modify the histogram to increase the apparent dynamic range and/or improve the apparent contrast.

Although there are several well developed histogram modification techniques, the one that is most commonly used is histogram equalization.[4] This is illustrated graphically in Figure 1.7. In this example, the number of discrete output gray levels is one-half the number of input gray levels, but the histogram has the desired characteristics that it is relatively flat, and covers the full dynamic range of values.

The histogram transformation problem can be stated as follows: determine a mapping function T, that transforms an input image histogram $f(x,y)$ to an output histogram $g(x,y)$ having some desired characteristic.

$$g(x,y) = T[f(x,y)].$$

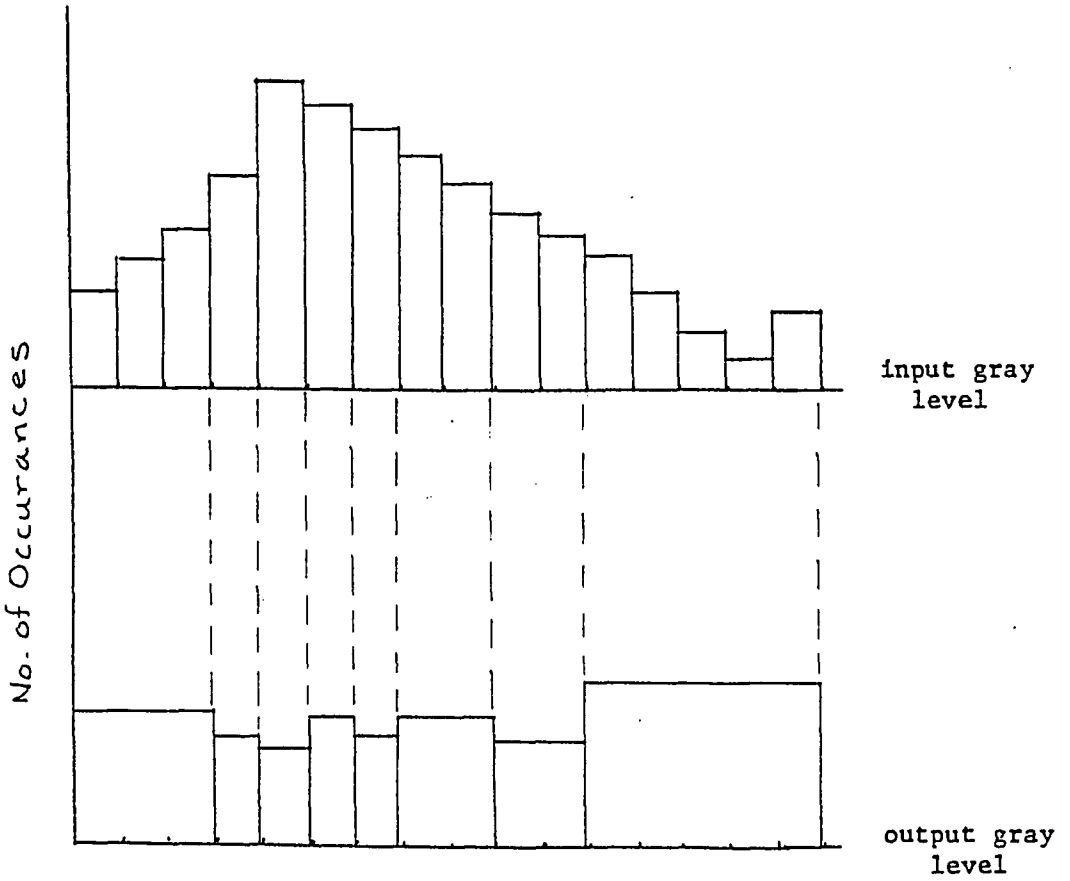


Figure 1.7 Histogram Equalization

A simple linear transformation could involve scaling the original histogram, with gray levels m through M, to cover the full range of gray levels n through N. This scaling could be expressed as:

$$g(x,y) = [(f(x,y)-m)/(M-m)]\ (N-n)$$

This type of mapping would be employed, for example when the image scanning or quantizer was not adjusted to cover its full dynamic range.

An input image f and output image g can be considered to be continuous random variables. The histograms for each of these functions can be normalized to produce the probability density functions $P_f(x)$ and $P_g(x)$ respectively. For histogram equalization, the objective is to determine a mapping function T, such that g(x,y) will have a flat density function.

A basic theorem from probability theory states that if X is a continuous random value with range

$$a \leq X \leq b$$

and distribution function

$$F_X(x) = \text{Prob}\ (X \leq x),$$

then the random variable defined by

$$Y = F_X(X)$$

has a uniform (0,1) distribution, i.e.:

$$F_Y(Y) = \begin{cases} 1, & 1 \leq y \\ y, & 0 \leq y < 1 \\ 0, & y < 0 \end{cases}$$

The theorem states that any absolutely continuous random variable, may be transformed into a uniformly distributed random variable. Since the theorem does not hold if X is not absolutely continuous, the use of this transformation for digital functions is based on the approximation of an absolutely continuous function by a finely quantized function. (3)

The theorem shows how to obtain a uniform (equalized) histogram with a gray level transformation. In general, a uniform distribution of gray levels makes equal use of each quantization level and tends to enhance low contrast information. This transformation is implemented by

(a) computing the image histogram,
(b) summing the histogram values to obtain a distribution curve, and
(c) using the distribution curve as the mapping function T.

The function T is the desired mapping function except for scaling. Since the original gray level values are integers and the distribution function varies between 0 and 1, some method of scaling must be used. A solution is to first convert the input gray levels to normalized values between 0 and 1. These values may then be mapped

directly using the distribution mapping. The output values are then scaled to integers as follows:

$$J = \text{Int}[(N-n)[(f(x,y)-m)/(M-m)]+0.5]$$

where the 0.5 term is added for symmetrical rounding.

Histogram equalization was performed on the infrared image in Figure 1.5. Figures 1.8a and 1.8b show the original and equalized image histograms. The transformed image is shown in Figure 1.9. Notice that the image intensity covers the full dynamic range, and that low contrast detail, barely visible in the original, is now visible.

There are two characteristics of histogram equalization that may produce undesired results. First, although equalization enhances low contrast detail, it does not discriminate between low contrast information and noise. Second, such equalization may impair further machine processing of the images - as in absolute intensity measurement or in segmentation for pattern recognition. Hence contrast enhancement is usually employed for the sole purpose of improving the subjective quality of images.

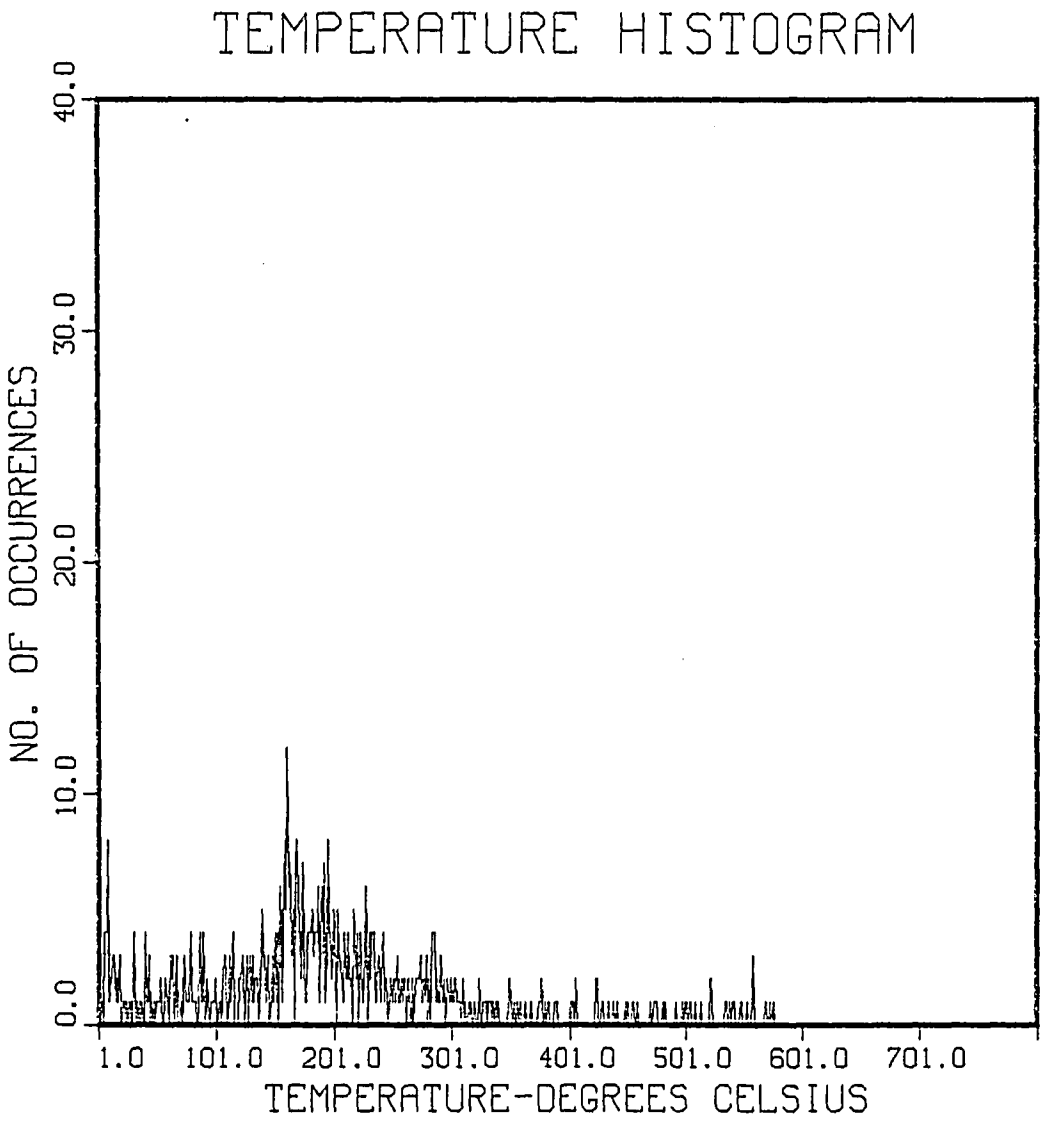


Figure 1.8 a Original Histogram of Infrared Image

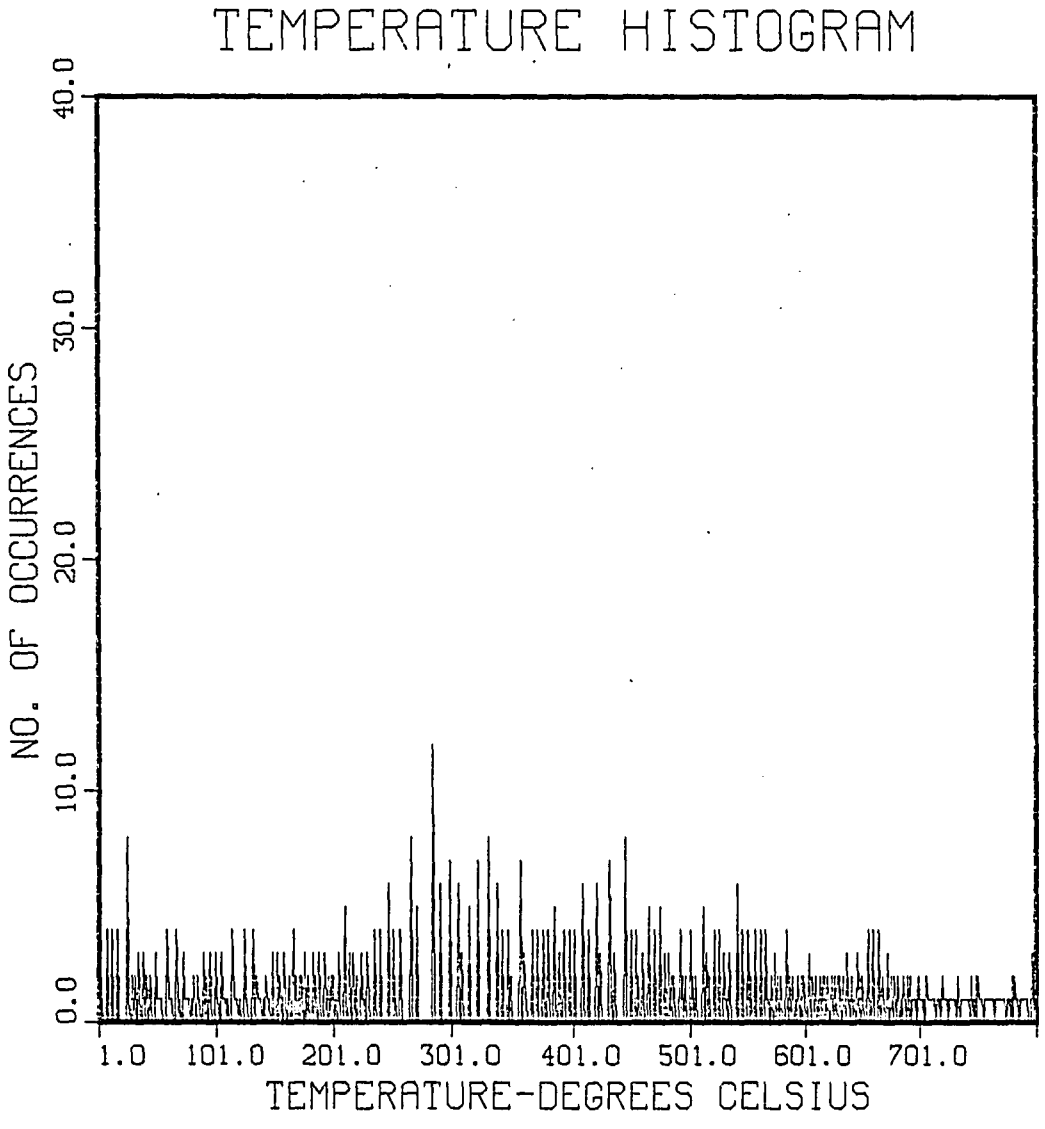


Figure 1.8 b Equalized Histogram of Infrared Image

IMAGE INTENSITY FUNCTION

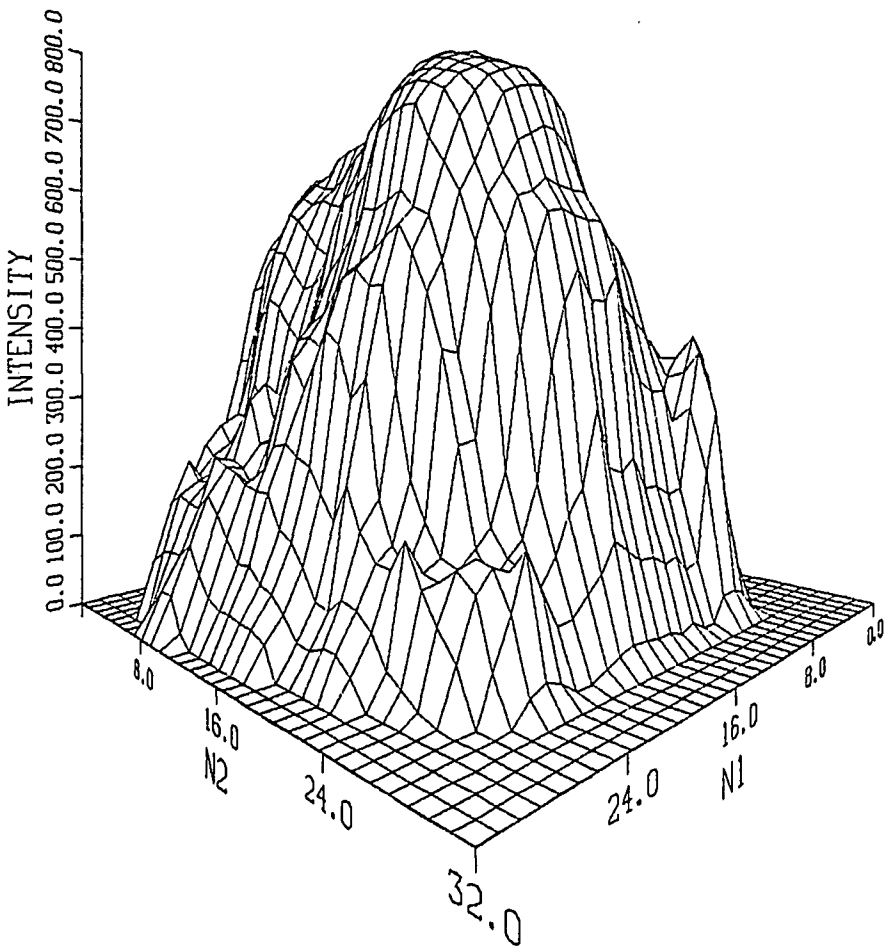


Figure 1.9 Infrared Image with Improved Contrast

1.2.1 Mask Filtering

Probably the most widely used technique for subjective improvement of image quality is spatial filtering using low pass or high pass masks. The masks are actually sampled impulse responses, or point spread functions, of two-dimensional filters with the desired characteristics. In most cases the impulse response is severely truncated to produce a very small mask. The filtering is accomplished by convolving the mask with the original image to produce an output image that is, in some sense, "enhanced". Low pass masks can be used to remove random noise from images, and high pass masks can be used to enhance edges in the image.

The spatial convolution of two functions corresponds to the frequency domain multiplication of the transforms of the two functions. Hence, mask filtering can be carried out in the frequency domain. However, for small masks, spatial convolution is generally simpler to implement and is computationally faster than the frequency domain approach.

The spatial convolution can be expressed mathematically as:

$$g(x,y) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(x-n,y-m) \cdot h(n,m)$$

where g(x,y) is the output image function, f(x,y) is the input function, and h is the 2-dimensional mask matrix of dimensions M by N. If the input image and mask matrix are real, as is usually the case, then the filtering can be carried out using only real multiplications and additions. As an example, a 32 by 32 point infrared image filtered with a 3 by 3 mask, requires less than 10,000 multiplications. The same filtering, done in the frequency domain, would require over 50,000 multiplications, even if Fast Fourier Transform algorithms were used.

Low frequency masks are used for removal of random noise from images. Random noise can give rise to extreme pixel to pixel variations in intensity, rather than the smooth variations normally observed in natural scenes or processes. For noise removal, a technique that detects the extreme variations can be used, and the affected points can be replaced by a local, weighted average. An example of a low pass mask to compute a local, weighted average is given below:

$$h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

This mask is symmetrical and is normalized to unit weighting. Figure 1.10 shows the image of Figure 1.5 filtered with the above mask. Clearly, the image intensity variations are much smoother in the filtered image. This image would lend itself to automatic feature extraction more readily than the original, because local random variations are much more subtle.

In addition to noise removal, masks can be used for edge enhancement. In this case, a high pass mask would be used to increase the visibility of low contrast edges. One of the simplest edge enhancement techniques is the digital equivalent of photographic "unsharp masking".[6] The essence of this method is to subtract a blurred representation of an image from the original image. The unsharp masking operation is equivalent to the convolution of the original image with a filter (mask) whose point spread function is of the form:

$$h = \begin{bmatrix} -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ -\frac{1}{8} & 1 & -\frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \end{bmatrix}$$

IMAGE INTENSITY FUNCTION

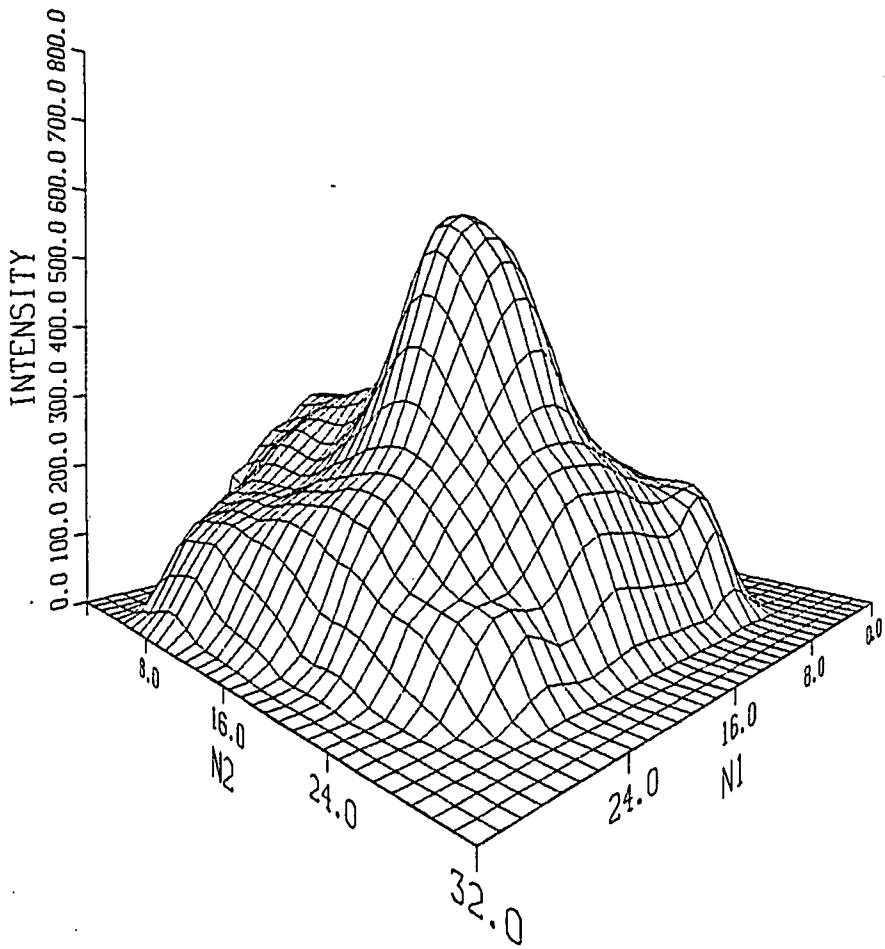


Figure 1.10 Convolution Filtered Image (low pass)

To illustrate the effect of this filter, assume an image contained a horizontal edge of the form (..., 0; 0, 1, 1, 0, 0, ...) with all zeroes elsewhere. When this portion of the image was convolved with the center of the above filter, the result would be (..., 0, -1/8, 7/8, 7/8, -1/8, 0, ...). Scaling could then be performed, for example by multiplying by 8, to produce (..., 0, -1, 7, 7, -1, 0, ...) This result more clearly defines the location of the edge.

The high pass mask above was convolved with the infrared image in Figure 1.5, and the result is shown in Figure 1.11. Notice that the mask has unity weighting and is symmetrical. The symmetry of the mask results in equal edge sensitivity in both the horizontal and vertical directions. The original infrared image contained very few sharp edges, except in the regions where the intensity changed from 0 to some positive value. The regions show up in the high pass filtered image as peaks, or regions of higher amplitude. Edge enhancement is most useful in processing images that contain very well defined shapes or objects. In fact, unsymmetrical high pass masks are sometimes used to enhance edges in specific directions. Two examples of such directional gradient filters are:

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad , \quad h = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

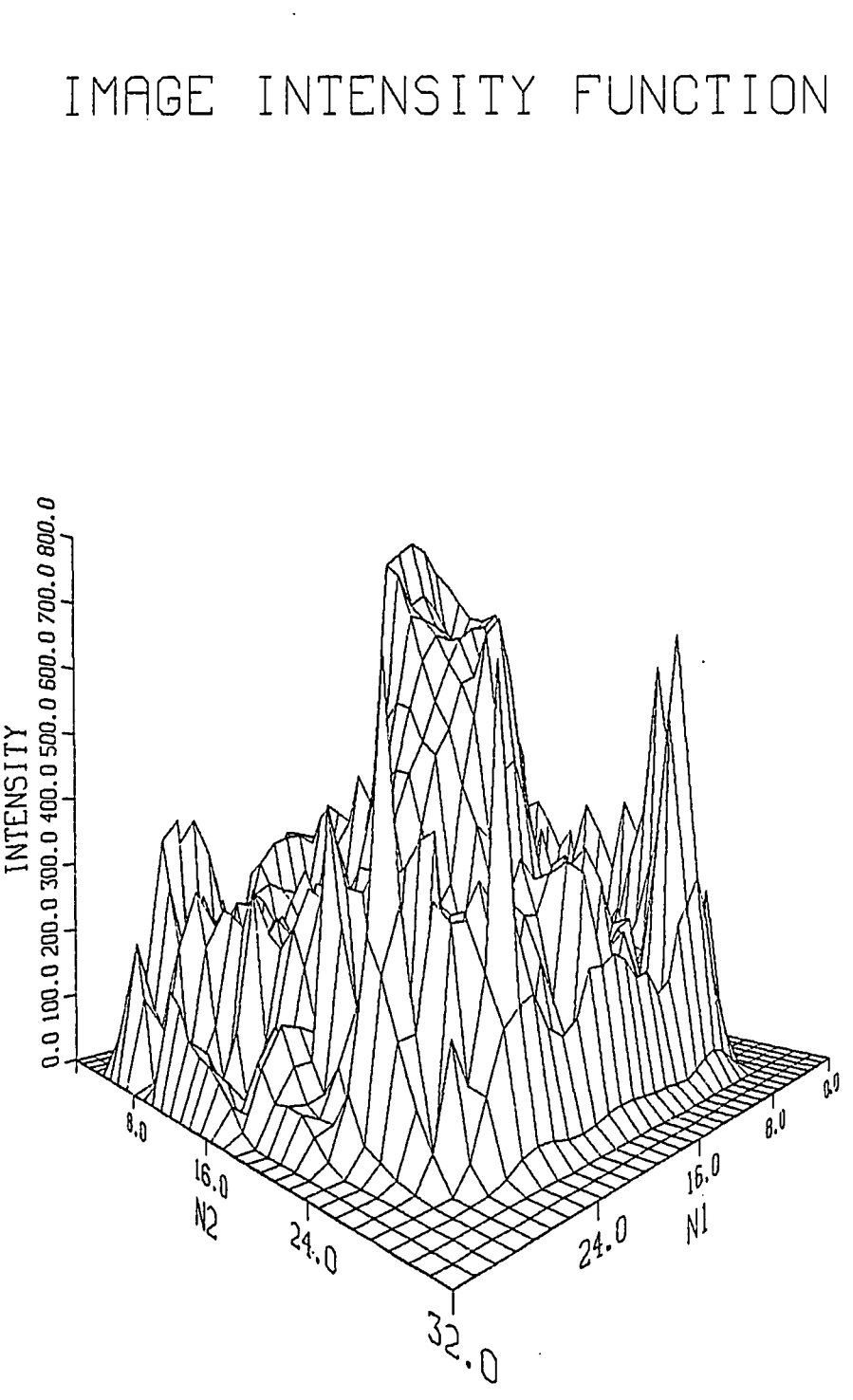


Figure 1.11 Convolution Filtered Image (high pass)

These masks would be most sensitive to edges oriented at 45 and 135 degrees respectively.

To summarize, spatial filtering using masks provides a convenient means of improving the subjective quality of images. However, the specific characteristics of the mask must usually be determined empirically, and there is no quantitative means of determining whether the "optimum" mask has been found. Therefore in many cases, it is desirable to employ more formal or analytical filter design techniques. Several of these techniques will be described later, but first, two dimensional transform processing will be illustrated.

1.3 Two-Dimensional Transforms

This section reviews two of the most commonly used and well documented two dimensional image transforms - Fourier and Walsh Hadamard. Both of these transforms were used on the original infrared image and the results are described in this section.

1.3.1 Fourier Transform

The continuous two dimensional Fourier transform is described by the transform pair:[7]

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

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

The significance of this transform in image processing is enhanced by the fact that the Fourier transform of an image field is equal to the far field or Fraunhoffer diffraction pattern of the image. For example the twinkling of a distant light or star at night is due to the observation of the Fourier transform. This property is also the basis for the computation of the Fourier transform using an optical system.

In both the continuous and discrete cases the Fourier transform is a complex function, i.e.:

$$F\ (u,v) = R\ (u,v) + j\ I\ (u,v)$$

where R and I represent the real and imaginary components respectively. F (u,v) can be represented by its magnitude and phase spectrum where:

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)}$$

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

$$\phi(u,v) = \tan^{-1} [I(u,v)/R(u,v)]$$

The human eye responds to the magnitude of an image field, therefore it is difficult to directly observe the phase spectrum. However, it is now widely known that the phase spectrum contains most of the information about the position of edges in an image. This fact will be demonstrated later.

The two-dimensional discrete Fourier transform (DFT) for a square N by N image is defined by the transform pair: (8)

$$F(k,l) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}(km+ln)}$$

$$f(n,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l)\, e^{j\frac{2\pi}{N}(km+ln)}$$

The significance of the DFT is that it may be computed on sampled data using a fast Fourier transform (FFT) algorithm.

Important properties of the DFT such as linearity, symmetry, shift of position, modulation and convolution are similar to those properties of the continuous transform. The only differences result from the fact is that the discrete transform is periodic as a result of the sampling of the image. As an example, the DFT computed on an image with a dark background and a high intensity area in the center, will have the DC or average value term at the origin of the array, and the largest magnitude terms occur at the corners. This is the case with all casual two-dimensional signals or images, and does not present any special difficulties in computer processing. It is worth mentioning that in some literature, origin centered transforms are computed. Origin centered transforms correspond directly to optically computed transforms, and using the shift property of the DFT, can be produced by multiplying each element of the image array by the factor $(-1)^{i+j}$. This will center the transform at the point (1/2N, 1/2N).

Observation of the DFT of an image reveals several important features of the image. The magnitude spectrum contains information about overall intensity and the frequency of intensity variations in the horizontal and vertical directions. As stated earlier, the phase spectrum contains information about the location of edges in

the image. Figure 1.12 shows the DFT magnitude spectrum of the infrared image of Figure 1.5. Since the original image intensity was fairly symmetric, the magnitude spectrum coefficients are also relatively symmetric. In addition the spectrum shows that the image contains primarily low frequency information. In fact, based on the magnitudes, there would appear to be relatively few significant coefficients. This infers that there is potential for efficient image coding, and also that low pass filtering could be done very effectively. Both of these ideas will be illustrated in the section on filtering.

Finally, the importance of the phase spectrum is illustrated in Figure 1.13, where the "inverted phase" spectrum has been plotted. The inverted phase spectrum was computed by setting the coefficient magnitudes to 1, and using the phase coefficients to compute the real and imaginary DFT coefficients:

$$R(u,v) = \text{Cos}(\phi(u,v))$$

$$I(u,v) = \text{Sin}(\phi(u,v))$$

These coefficients were then used to compute the inverse DFT. Clearly the inverted phase spectrum reveals the location of edges in the original image, and is qualitatively similar to a high pass filtered image.

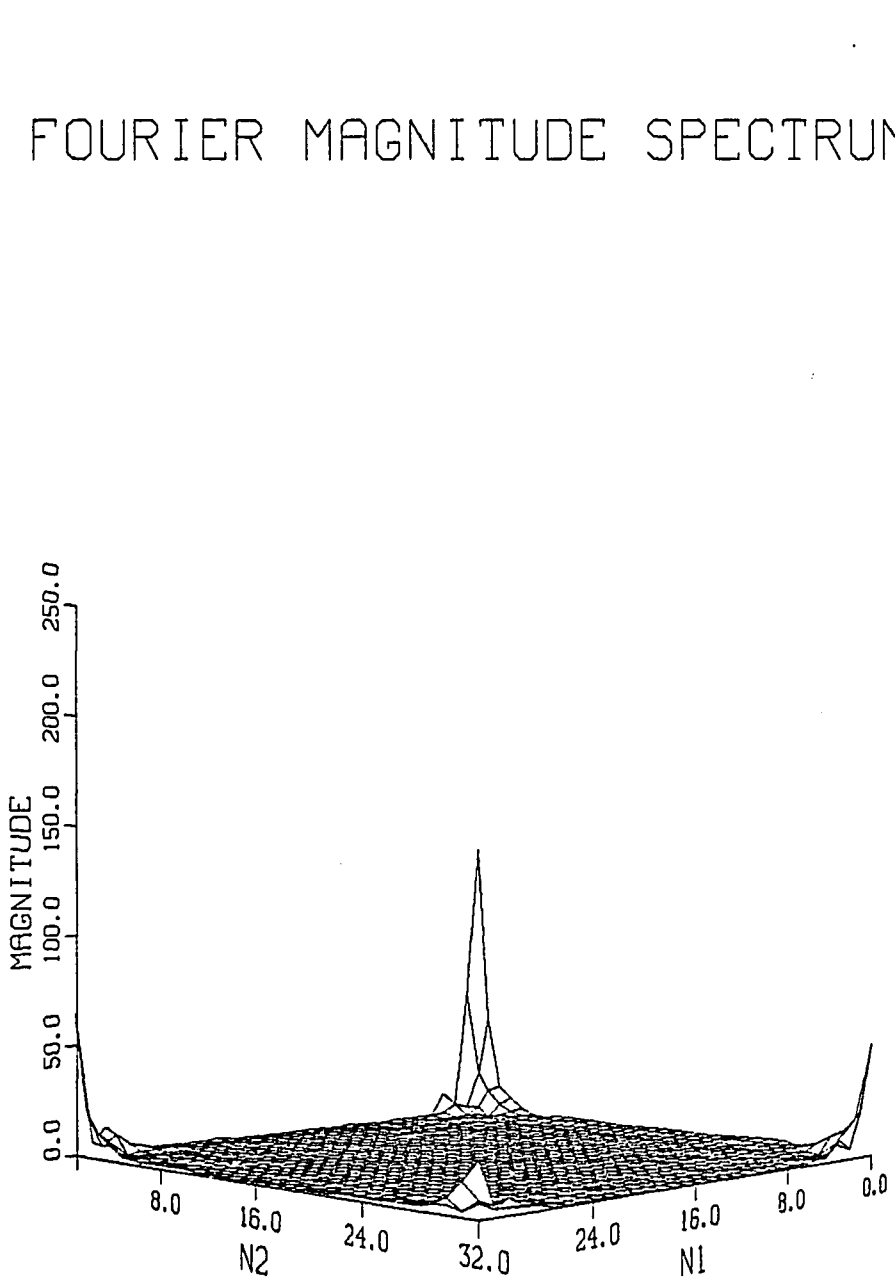


Figure 1.12 Discrete Fourier Transform Coefficients

INVERSE TRANSFORM

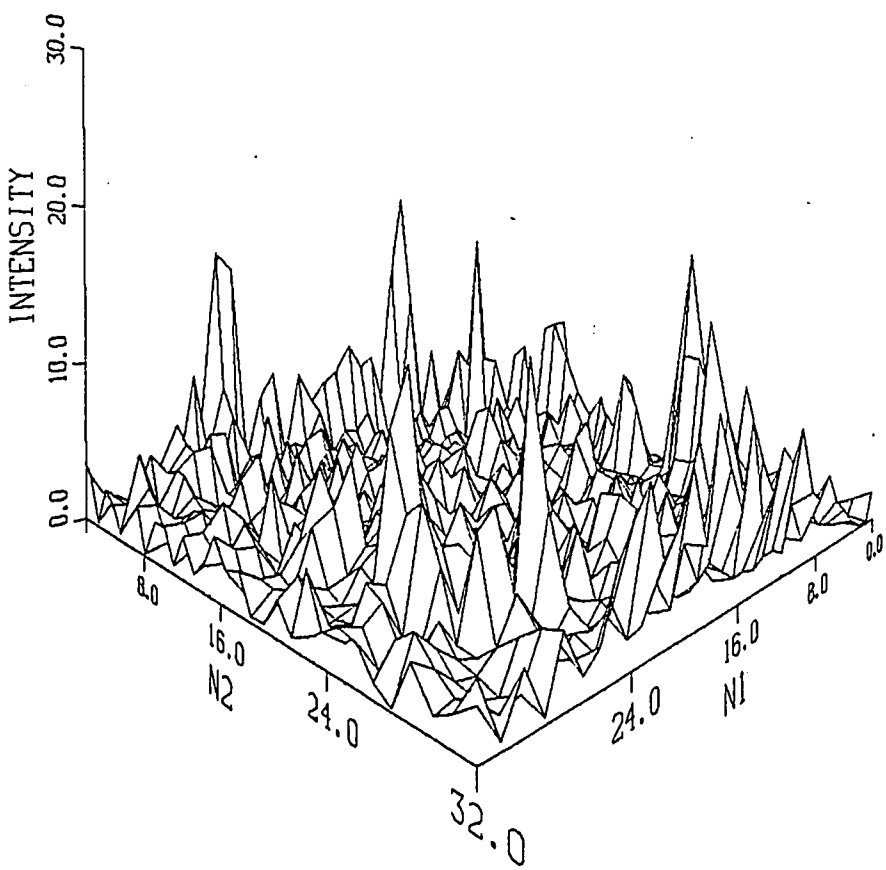


Figure 1.13 Inverted Phase Spectrum

1.3.2 Walsh-Hadamard Transform

Another useful two-dimensional transform is the Walsh-Hadamard transform. The Walsh-Hadamard transform is computed using Walsh functions as basis vectors, the first 8 of which are shown in Figure 1.14.[9] Sequency of the Walsh functions is the analog of frequency of the Fourier sine and cosine basis vectors. The functions of Figure 1.14 are in sequency and not Hadamard order. Sequency is defined as one-half the number of zero crossings of a function on a specified interval.

$$\text{sequency} = s_i = \begin{cases} 0 , & i=0 \\ i/2 , & i \text{ even} \\ (i+1)/2 , & i \text{ odd} \end{cases}$$

The CAL and SAL functions are even and odd respectively and are analogous to the sine and cosine functions. The most significant property of the Walsh functions is that they contain only terms whose values are 1 and -1. Thus no complex arithmetic is required to compute the transform. Another advantage, since there are no complex terms, is that the transform coefficients can be stored in one-half the space that the Fourier coefficients would require for a given size image.

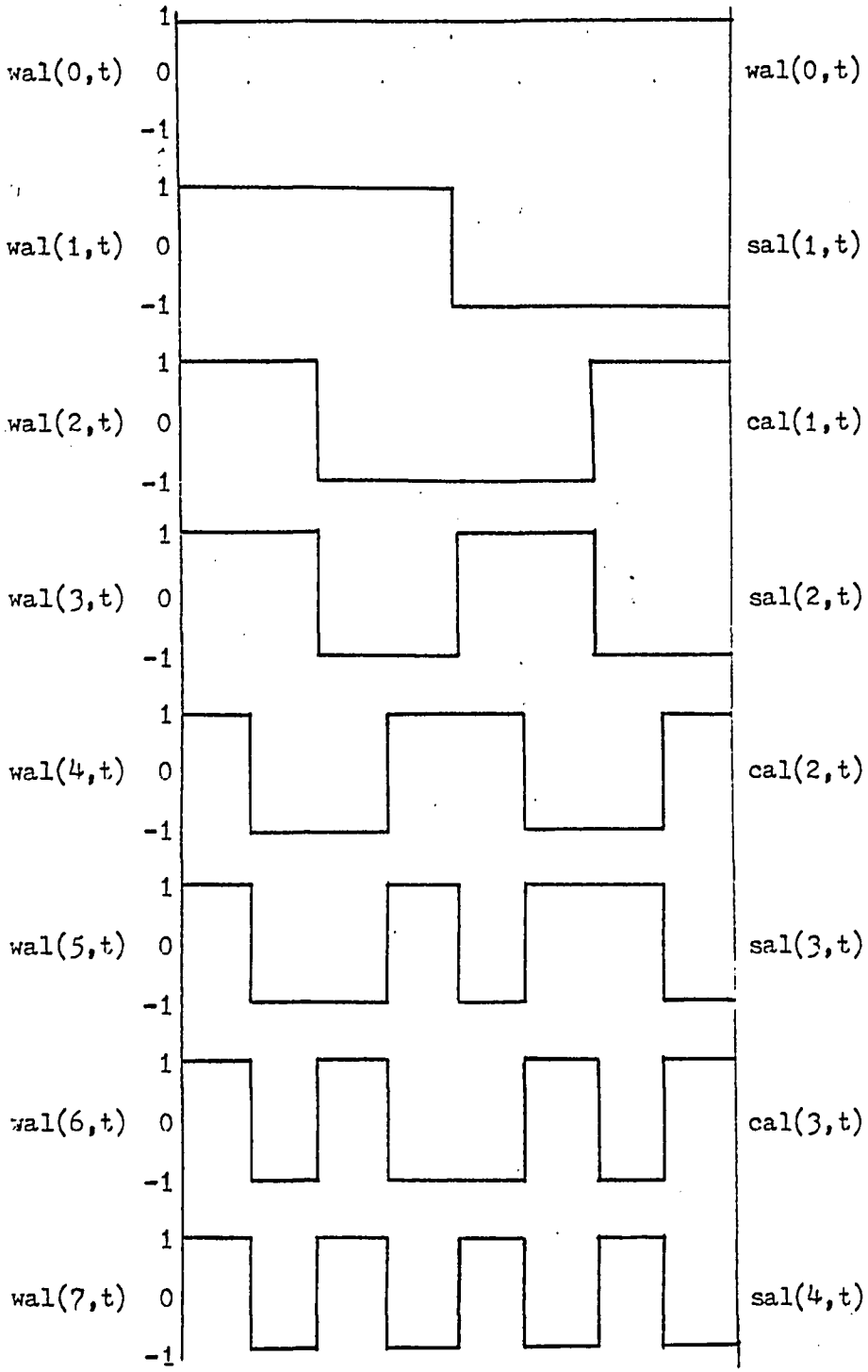


Figure 1.14 Walsh Functions

A one dimensional signal x(t) can be described using the Walsh functions as:

$$x(t) = \sum_{k=0}^{\infty} d_k \, wal_w(k,t)$$

$$d_k = \int_0^1 x(t) \, wal_w(k,t) \, dt \, , \quad k=0,1,2,.....$$

or:

$$x(t) = a_o \, wal_w(0,t) + \sum_{k=1}^{\infty} [a_k \cdot cal(k,t) + b_k \cdot sal(k,t)]$$

where:

$$a_0 = d_0$$
$$a_k = d_{2k}$$
$$b_k = d_{2k-1}$$

The above equation shows that the Walsh representation of signals is analogous to the Fourier representation. Intuitively, this is what one would expect because of the resemblance between the Walsh functions and the Fourier sinusoids. Since the Walsh-Hadamard

transform does not include complex coefficients, the phase spectrum is not usually considered. However, a Walsh-Hadamard phase spectrum is described in at least one text[9]. An analogy with the DFT phase spectrum is used to define the WHT phase spectrum as:

$$\phi_w(s) = \tan^{-1} \frac{w_w(2s-1)}{w_w(2s)}$$

or:

$$\phi_w(s) = \tan^{-1} \frac{sal(k)}{cal(k)}$$

The difference is that for each Fourier phase coefficient, the real and imaginary parts of a single frequency term are used to compute the phase. Each phase component for the Walsh-Hadamard transform requires two sequency coefficients for definition.

Computation of the Walsh-Hadamard transform in two dimensions can be expressed, in matrix form, as:

$$[F(u,v)] = [H(u,v)]\,[f(x,y)]\,[H(u,v)]$$

where:

$[F(u,v)]$ = WHT coefficient matrix

$[H(u,v)]$ = Walsh-Hadamard basis matrix

$[f(x,y)]$ = input image

The inverse transform is computed as:

$$[f(x,y)] = \frac{1}{N^2}\,[H(u,v)]\,[F(u,v)]\,[H(u,v)]$$

for an N by N image.

The sequency ordered Walsh-Hadamard transform can be thought of as the summed product or non-normalized correlation of the corresponding basis matrix with the original image[10]. Thus the (1,1) element may be interpreted as the average value, the (2,1) element as the correlation with a single step horizontal edge, the (1,2) element as the correlation with a single step vertical edge, the (3,1) element as the correlation with a horizontal line, and so on. These correlation relationships contain the same kind of information that the Fourier phase spectrum contains, i.e: the locations of edges in the image.

A relationship analogous to Paserval's relation for the Fourier transform exists for the Walsh-Hadamard transform:

$$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \left|f(x,y)\right|^2 = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \left|F(u,v)\right|^2$$

The implication is that if a few of the WHT coefficients are large in magnitude, then many of the remaining coefficients must be of small magnitude. In many cases the smaller magnitude coefficients may be discarded to effect a bandwidth reduction. This is the basis for WHT image coding.

Figure 1.15 shows the Hadamard ordered WHT magnitude spectrum of the infrared image of Figure 1.5. Note that the coefficients are Hadamard, and not sequency, ordered. The lower sequency coefficients have the largest magnitude, and there are many very small coefficients. The sequency information presented follows the same pattern as the DFT frequency information. However the WHT contains more very small coefficients than the DFT and hence more efficient data compression would be possible, in this case, using the WHT. This will be illustrated in the section on digital filtering.

1.4 Two Dimensional Filtering

Filtering of images is done for a number of reasons depending on the application. Among the most common uses of filtering are removal of random noise, removal of noise caused by the process being sensed, and enhancement of certain features such as edges. It was shown previously that filtering could be accomplished by convolving various types of masks with the image to be filtered. This type of filtering is generally employed to improve the subjective quality of images for improved viewing. Where filtering is required for the removal of specific image frequency components (noise) or the enhancement of certain features, it is necessary to use more quantitative and exact filter techniques.

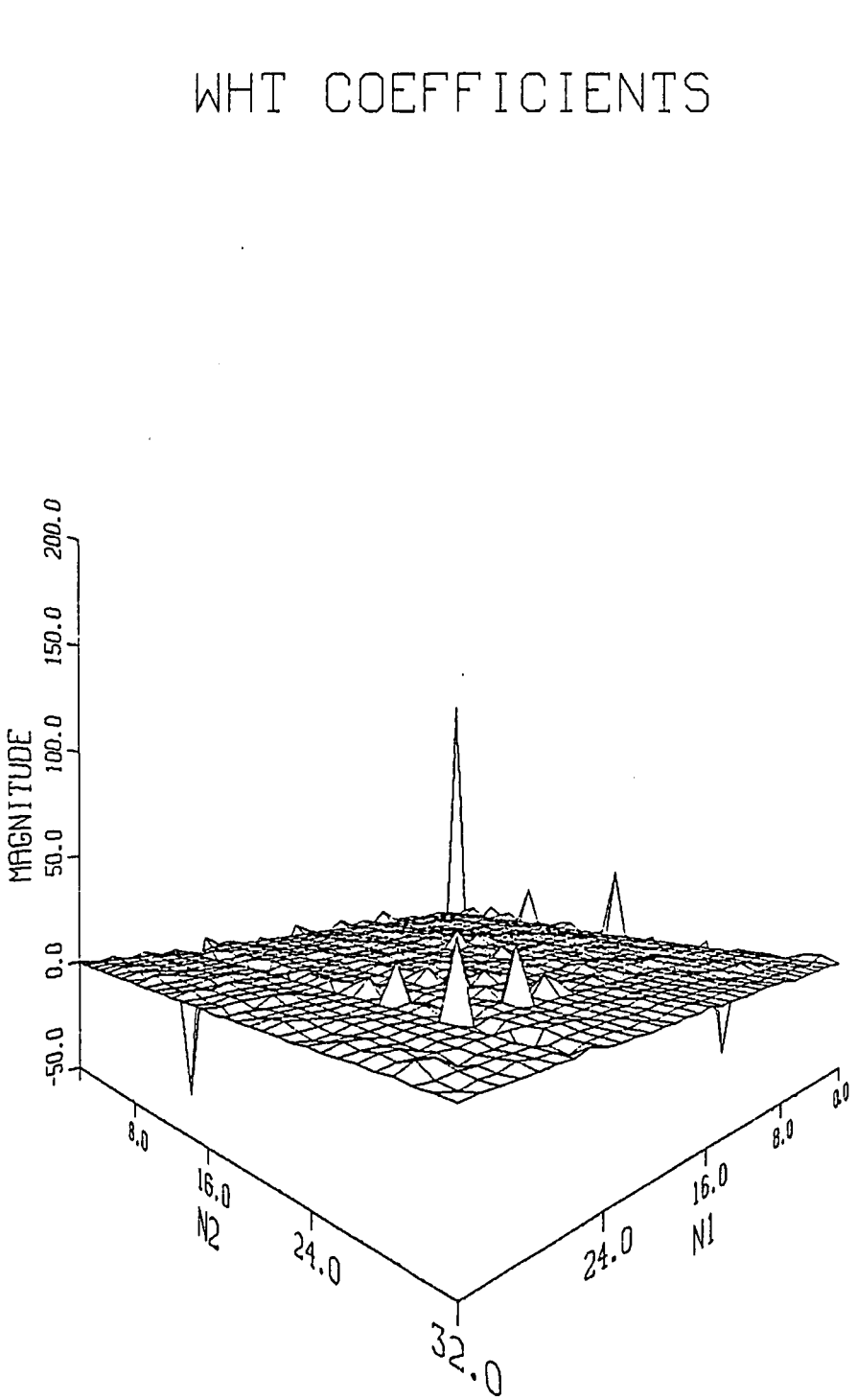


Figure 1.15 Walsh-Hadamard Transform Coefficients

1.4.1 Frequency Domain Representation of Images

A discrete two-dimensional system with impulse response $h(m_1,m_2)$ has an input/output relation of the form:[12]

$$y(n_1,n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(m_1,m_2)\ x(n_1-m_1,n_2-m_2)$$

where x is the input and y is the output. If

$$x(n_1,n_2) = e^{jn_1w_1}\ e^{jn_2w_2},$$

$$y(n_1,n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(m_1,m_2)\ e^{j(n_1-m_1)w_1}\ e^{j(n_2-m_2)w_2}$$

$$= e^{jn_1w_1}\ e^{jn_2w_2}\ H(e^{jw_1},e^{jw_2})$$

where:

$$H(e^{jw_1},e^{jw_2}) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(m_1,m_2)\ e^{-jn_1w_1}\ e^{-jn_2w_2}$$

H is, by definition, the frequency response of the system. H is periodic in both W_1 and W_2 with period 2π .

$$H(e^{jw_1},e^{jw_2}) = H(e^{jw_1+j2k\pi},e^{jw_2+j2l\pi})$$

Recalling that convolution in the spatial domain (or in time) is equivalent to multiplication in the frequency domain;

$$Y(e^{jw_1},e^{jw_2}) = H(e^{jw_1},e^{jw_2})\cdot X(e^{jw_1},e^{jw_2})$$

This relationship is the basis for fast, frequency domain filtering. The relationship above is implemented by computing the complex DFT of both the filter frequency response and the input image, multiplying to generate the output image DFT, and then performing the inverse DFT to generate the filtered output image. Later in this section, techniques will be presented for deriving the filter frequency response, but first a more direct transform filtering technique will be illustrated.

1.4.2 Direct Transform Filtering

Direct transform filtering is accomplished by computing the Fourier or Walsh-Hadamard transform of an image, systematically forcing certain coefficients to zero, and then computing the inverse transform to produce the filtered image. This is equivalent to time domain filtering with an ideal (but unrealizable) filter having unity magnitude in the passband, an infinitely sharp transition band, and zero magnitude in the stopband.

There are two main advantages to this type of filtering. First, it is extremely simple to implement since it involves only the image function - no filter function is involved. Second, since certain coefficients are set to zero (or discarded) it provides a direct means of data compression. That is, the image can be stored using the remaining transform coefficients at a significant reduction in required storage space.

A disadvantage to this type of filtering is that the filtered image is distorted by the elimination of some transform coefficients. This effect is not the result of aliasing per se, but is similar to time domain distortion that would be caused by the elimination of certain Fourier series coefficients. Another disadvantage to this method is that it has limited application. It should be used only with images whose spectra contain components with very large or

very small magnitudes. For images with slow transitions from the very large to very small magnitude coefficients, application of direct transform filtering may result in significant distortion in the filtered image.

This technique can be applied, with reasonably good results, to the infrared images. Figures 1.16 and 1.17 show the infrared image of Figure 1.5 after filtering with a low pass, direct DFT and direct WHT technique respectively. The distortion in the filtered images manifests itself as "ripples" in the image, especially in the low intensity regions. In this example, filtering with either transform appears to have produced approximately the same result. However, only 50 percent of the DFT coefficients were set to zero, while over 70 percent of the WHT coefficients were set to zero. Thus the WHT filtering was considerably more efficient.

1.4.3 Analytical Two-Dimensional Filter Design

The mathematical basis for digital filter design is the study of the z-transform. Z transforms for filter functions are computed, and the characteristics of the filters can be studied by observing the poles and zeroes in the z-plane. Z-transform techniques are generally easy to apply to one-dimensional systems because the

INVERSE TRANSFORM

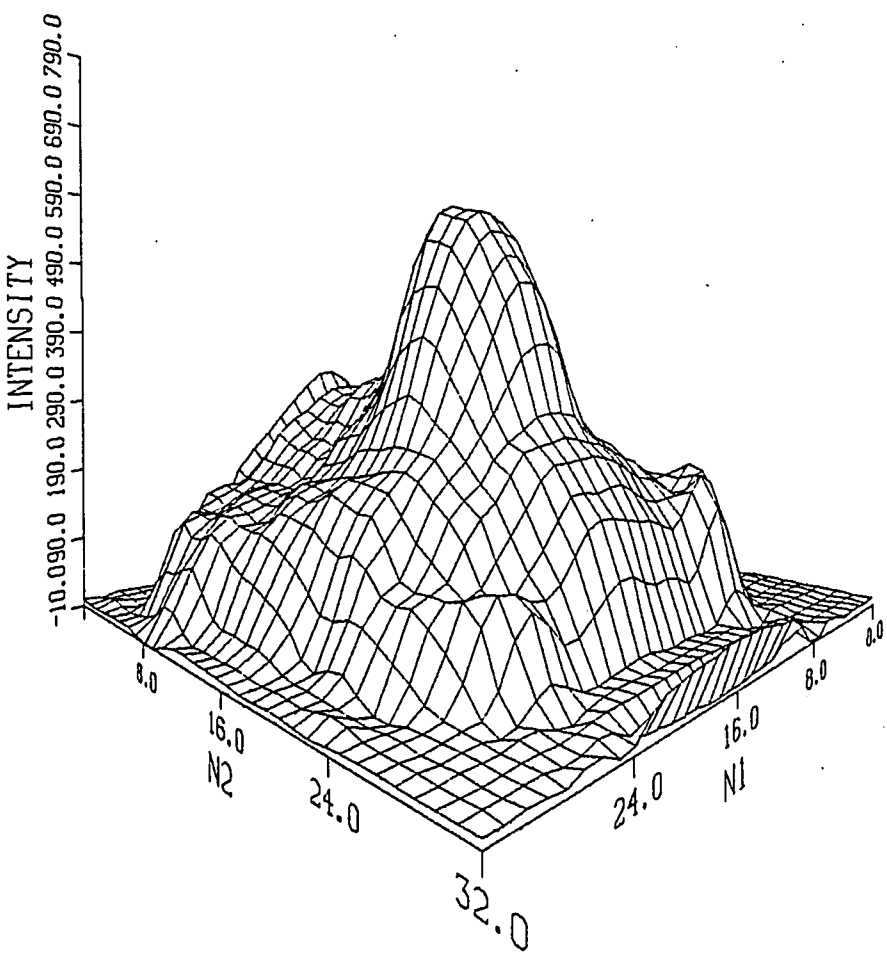


Figure 1.16 Image Filtered with Direct DFT Technique

INVERSE WHT

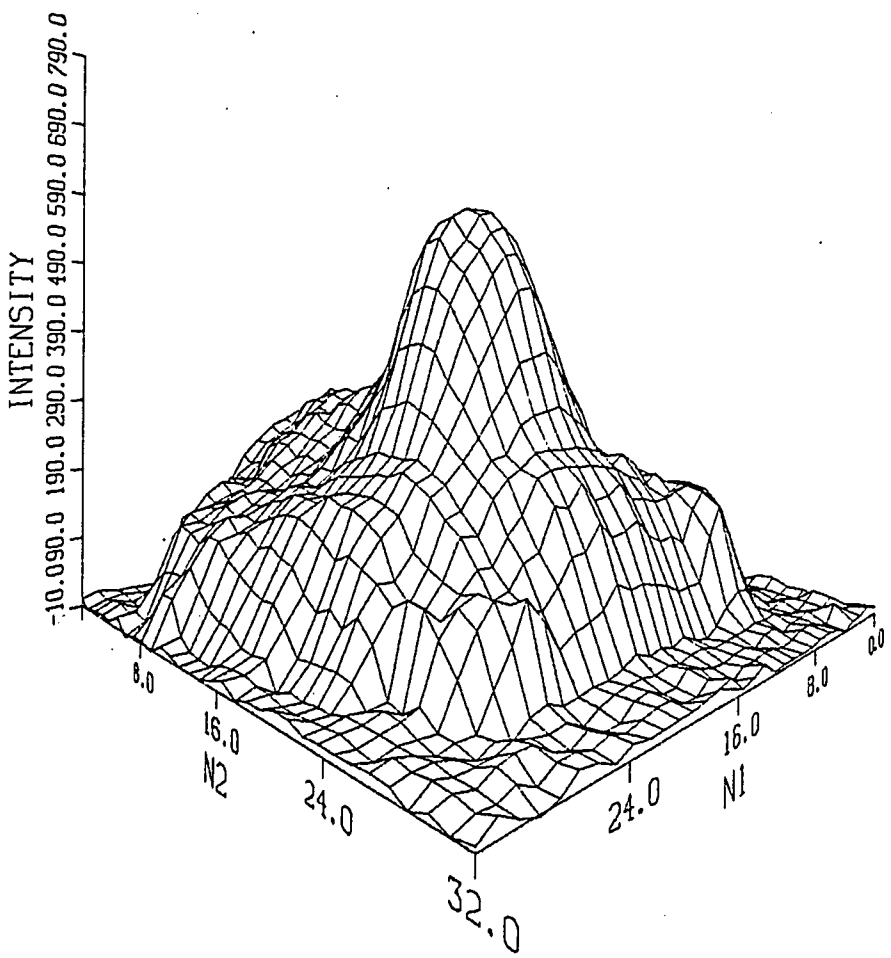


Figure 1.17 Image Filtered with Direct WHT Technique

convergence of the transform can be easily evaluated. The 2-D z-transform of a sequence $x(n_1, n_2)$ is defined as:

$$X(z_1,z_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1,n_2)\, z_1^{-n_1}\, z_2^{-n_2}$$

The inverse relation is:

$$x(n_1,n_2) = (1/2\pi j)^2 \oint_{C_1} \oint_{C_2} X(z_1,z_2)\, z_1^{n_1-1}\, z_2^{n_2-2}\, dz_1\, dz_2$$

where C_1 and C_2 are closed, contours encircling the origin and within the region of convergence.

As stated above it is very difficult to study in detail the convergence of the 2-D z transforms, because the $z_1 - z_2$ space is actually four-dimensional. This difficulty is alleviated in the case of finite length sequences which are bounded. For finite length sequences, the z-transforms converge everywhere in the z-plane except perhaps at $z_1 = 0$, $z_2 = 0$, or $z_1 = \infty$, $z_2 = \infty$. It is for this reason that finite impulse response (FIR) filters are almost always used to filter images (or two-dimensional signals). Infinite impulse response (IIR) filters are generally not used because their stability, which is directly related to the conver-

gence of the z-transform, is difficult to predict. The remainder of this section will deal with the design of 2-D FIR filters.

The finite impulse response design technique involves defining a desired frequency response D, and then computing the filter coefficients (or impulse response $h(n_1, n_2)$. In general:[13]

$$D(e^{j2\pi f_1}, e^{j2\pi f_2}) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} h(n_1,n_2)\, e^{-j2\pi n_1 f_1}\, e^{-j2\pi n_2 f_2}$$

$$h(n_1,n_2) = \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} D(e^{j2\pi f_1}, e^{j2\pi f_2})\, e^{j2\pi f_1}\, e^{j2\pi f_2}\, df_1\, df_2$$

This analytical expression for the filter coefficients $h(n_1,n_2)$ can be difficult to evaluate unless certain assumptions are made about the symmetry of the desired frequency response. Generally, either rectangular (or square) or circular symmetry is assumed. Of these, circular symmetry is more popular because one-dimensional impulse responses and window functions can be extended to two dimensions using straightforward trigonometric techniques. Even though, for sampled systems, the DFT coefficients repeat with rectangular symmetry, very good approximations can be attained in practice using circular symmetry.

The two most popular FIR design techniques in two dimensions are windowing and frequency sampling. Optimum design using Chebyshev approximations is not often used because it is not possible to generalize multiple exchange algorithms such as the Remez exchange algorithm. This is because the alternation problem is not directly generalizable in the 2-D case.[14] The technique of windowing is the most straightforward to apply, and in image processing produces excellent results.

As in the one-dimensional case, the object of windowing is to produce a smooth truncation of the filter impulse response. Assuming the impulse response is windowed to length N_1 by N_2:

$$H(e^{j2\pi f_1}, e^{j2\pi f_2}) = \sum_{n_1=-\bar{N}_1}^{\bar{N}_1} \sum_{n_2=-\bar{N}_2}^{\bar{N}_2} h(n_1,n_2)\, e^{-j2\pi f_1 n_1}\, e^{-j2\pi f_2 n_2}$$

where:

$$\bar{N} = \frac{N-1}{2}$$

For circular symmetry:

$$h(n_1,n_2) = h(n_1,-n_2) = h(-n_1,n_2) = h(-n_1,-n_2)$$

To obtain a causal response, $h(n_1, n_2)$ is shifted $(N_1-1)/2$ samples in the positive n_1 direction and $(N_2-1)/2$ samples in the positive n_2 direction. This corresponds to a linear phase term:

$$e^{-j(N_1-1)/2w_1} \; e^{-j(N_2-1)/2w_2}$$

For N_1 and N_2 even, this phase shift term corresponds to a delay that is a non-integer number of samples. The impulse response requirements for linear phase in the 2-D case are directly analogous to the requirements in the 1-D case.[15] To preserve the impulse response symmetry, the window must have even symmetry. The "windowed" frequency response can be expressed as:

$$\hat{D}(e^{j2\pi f_1}, e^{j2\pi f_2}) = \sum_{n_1=-\bar{N}_1}^{\bar{N}_1} \sum_{n_2=-\bar{N}_2}^{\bar{N}_2} \hat{h}(n_1,n_2) \; e^{-j2\pi f_1 n_1} \; e^{-j2\pi f_2 n_2}$$

where

$$\hat{h}(n_1, n_2) = h(n_1, n_2) \cdot w(n_1, n_2)$$

and w is the window function.

$$\hat{D}(e^{j2\pi f_1}, e^{j2\pi f_2}) = D(e^{j2\pi f_1}, e^{j2\pi f_2}) * W(e^{j2\pi f_1}, e^{j2\pi f_2})$$

where W is the frequency response of the window. Thus the frequency response is the desired response smeared by the response of the window. The frequency transform of a good window will:

(1) be a circularly symmetric function,
(2) have a large volume under its main lobe, and
(3) have a small volume under the secondary side lobes.

In addition, the window response should be narrow with respect to the desired frequency response. If this is the case, the actual filter response will "look like" the desired response. This, however, requires a long duration window containing many samples. In practice the window function is chosen as short as possible to minimize computation time while having a frequency response as narrow as possible to faithfully reproduce the desired response. The height of the side lobes in the window response can be made smaller by tapering the window smoothly to zero at both ends. But this causes a wider main lobe and a wider transition band in the filter frequency response. There are several well developed window functions that are designed to minimize oscillations and smearing at discontinuities in the filter response. Among these are the Bartlett, Hamming, Hanning, Blackman, and Kaiser windows. In general, if w(n) is a good 1-D window and if W(w) is narrow with

respect to the bands of the desired filter, then the 2-D circularly symmetric window obtained from rotating w(n) as:

$$w(n_1,n_2) = w(n_1^2 + n_2^2)^{1/2}$$

is also a good window.[16]

The usual process for windowed filter designs is to define the desired filter response and then calculate the impulse response using the integral relationship. A useful technique for doing this when circular symmetry is used is the Hankel transform.[17] For circular symmetry:

$$f(x,y) = f_r(r)$$
$$r^2 = x^2 + y^2$$

then:

$$\int_{-\infty}^{\infty}\int f(x,y)\, e^{-j2\pi(ux+vy)}\, dx\, dy = \int_0^{\infty}\int_0^{2\pi} f_r(r)\, e^{-j2\pi qr\cos(\theta-\phi)} \cdot r\, dr\, d\theta$$

where:

$$x + jy = re^{j\theta} \quad \text{and} \quad u + jv = qe^{j\phi}$$

Angle ϕ can be dropped because the integral is taken over one full cycle of the cosine:

$$F(\ f(x,y)\ ) \int_0^{\infty} f_r(r) \int_0^{2\pi} e^{-j2\pi qr\ \cos\theta}\ d\theta\ r\ dr$$

And:

$$J_0(z) = \frac{1}{2\pi} \int_0^{2\pi} e^{-jz\ \cos\theta}\ d\theta$$

where J_o is the Bessel function of the first kind of order 0.

$$F(\ f(x,y)\ ) = 2\pi \int_0^{\infty} f_r(r)\ J_0(2\pi qr)\ r\ dr$$

$$F(u,v) = F_r(q)$$

$$q^2 = u^2 + v^2$$

Thus the Fourier transform of a 2-D circularly symmetric function can be expressed as a function of only a single radial frequency q.

$$F_r(q) = 2\pi \int_0^\infty f_r(r)\, J_0(2\pi qr)\, r\, dr$$

$$f_r(r) = 2\pi \int_0^\infty F_r(q)\, J_0(2\pi qr)\, q\, dq$$

This special case of the 2-D Fourier transform is called the Hankel transform of order zero. A useful table of Hankel transforms in given in reference (17).

Where the original integral cannot be evaluated and the Hankel Transform cannot be used, an approximation to the impulse response can be obtained by sampling the desired frequency response and then calculating an inverse DFT. The greater the number of samples, the better the approximation to the desired response.

1.4.4 Implementation of 2-D Filters

Two dimensional digital filters designed by any method can be applied using either direct convolution in the time domain or frequency domain multiplication. With direct convolution the number of multiplications required is:

$$N_1 \times N_2 \times M_1 \times M_2.$$

where N_1 and N_2 are the filter dimensions and M_1 and M_2 are the image dimensions. Filtering in the frequency domain is accomplished by augmenting the filter and image arrays with zeroes to allow buffer space for the effect of circular convolution. The dimensions of both the new filter and image arrays are L_1 and L_2. Filtering is accomplished by:

(1) Computing the image 2-D FFT:

$$X(k_1, k_2);\ k_1 = 0, \ldots L_1 - 1$$
$$k_2 = 0, \ldots L_2 - 1$$

of $x(n_1, n_2)$ where

$$x(n_1, n_2) = x(n_1, n_2) \quad n_1 = 0, \ldots M_1 - 1$$
$$n_2 = 0, \ldots M_2 - 1$$
$$x(n_1, n_2) = 0 \quad n_1 = M_1 \ldots L_1 - 1$$
$$n_2 = M_2 \ldots L_2 - 1$$

where L_1 and L_2 are the minimum powers of 2 greater than $M_1 + N_1 - 1$, $M_2 + N_2 - 1$.

(2) Computing the filter 2-D FFT:

$$H(k_1, k_2);\ k_1 = 0, \ldots L_1 - 1$$
$$k_2 = 0, \ldots L_2 - 1$$

of $h(k_1, k_2)$ where

$$h(n_1, n_2) = h(n_1, n_2) \quad n_1 = 0, \ldots N_1 - 1$$
$$n_2 = 0, \ldots L_1 - 1$$
$$h(n_1, n_2) = 0 \quad n_1 = N_1, \ldots L_1 - 1$$
$$n_2 = N_2, \ldots L_2 - 1$$

(3) Computing the complex product:

$$Y(k_1, k_2) = X(k_1, k_2) \cdot H(k_1, k_2)$$

$$k_1 = 0, \ldots\ldots L_1 - 1$$

$$k_2 = 0, \ldots\ldots L_2 - 1$$

(4) and then computing the inverse FFT of $Y(k_1, k_2)$ which is the convolution of the filter function with the image.

The number of multiplication using the frequency domain approach is:

$$L_1 L_2 (\log_2 L_1 + \log_2 L_2).$$

Usually the input matrix is real, therefore the DFT is symmetric with the origin and therefore with the L/2 point. Also, the imaginary part of the DFT is antiysmmetric with respect to the same point. Thus the real part has L/2 + 1 independent values and the imaginary part has L/2 -1 independent values (since the 0 and L/2 points are zero for antisymmetry). So the transform of the L point sequence is specified by L real values, which reduces the number of required computations.

Figure 1.18 shows the magnitude response of a 17 by 17 point circularly symmetric low pass filter, windowed with a rectangular window. The cutoff frequency is $\pi/2$. The ripples in the passband and stopband are caused by the rectangular window. Figure 1.19 is

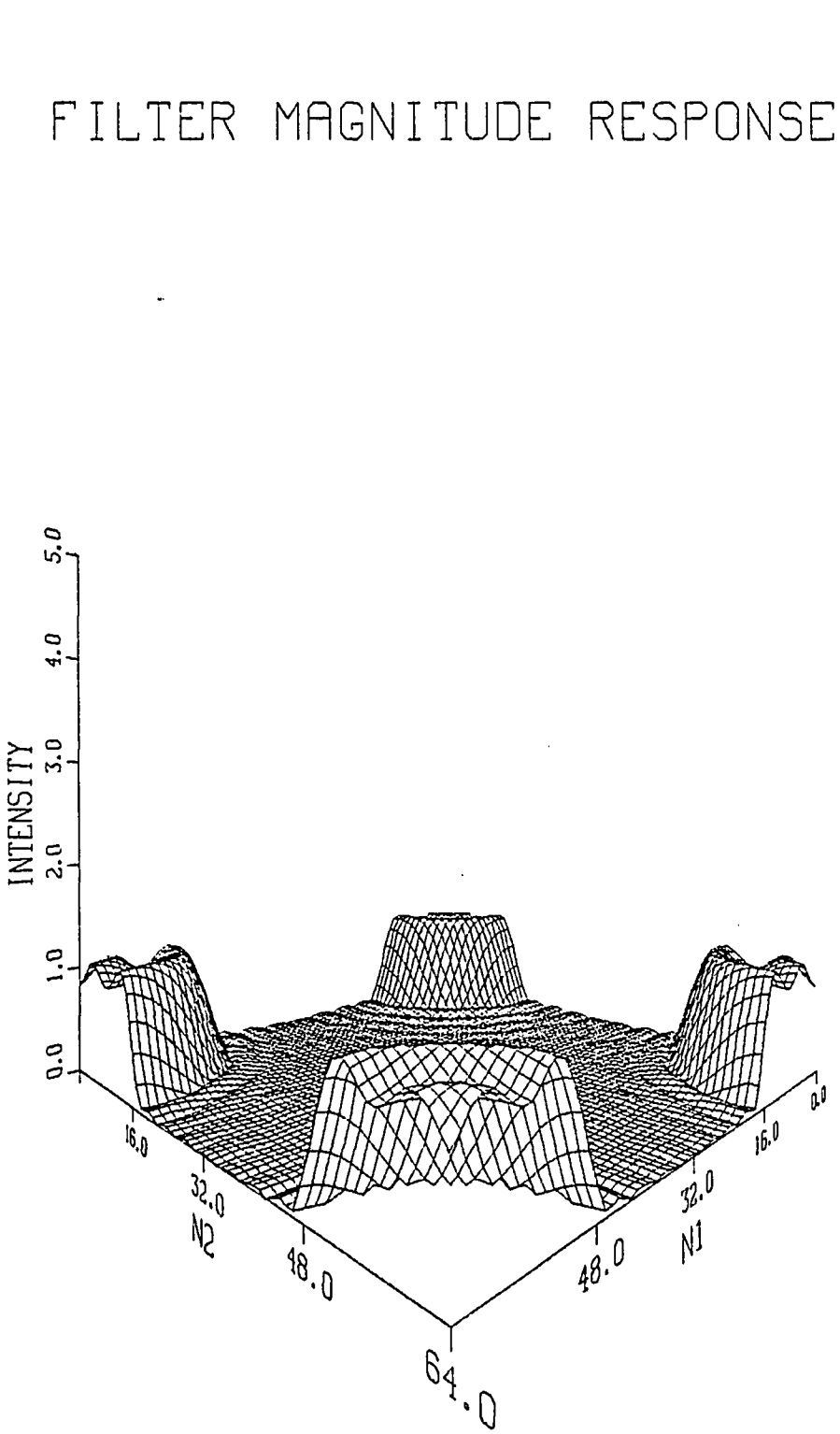


Figure 1.18 Magnitude Response of FIR Low Pass Filter (rectangular)

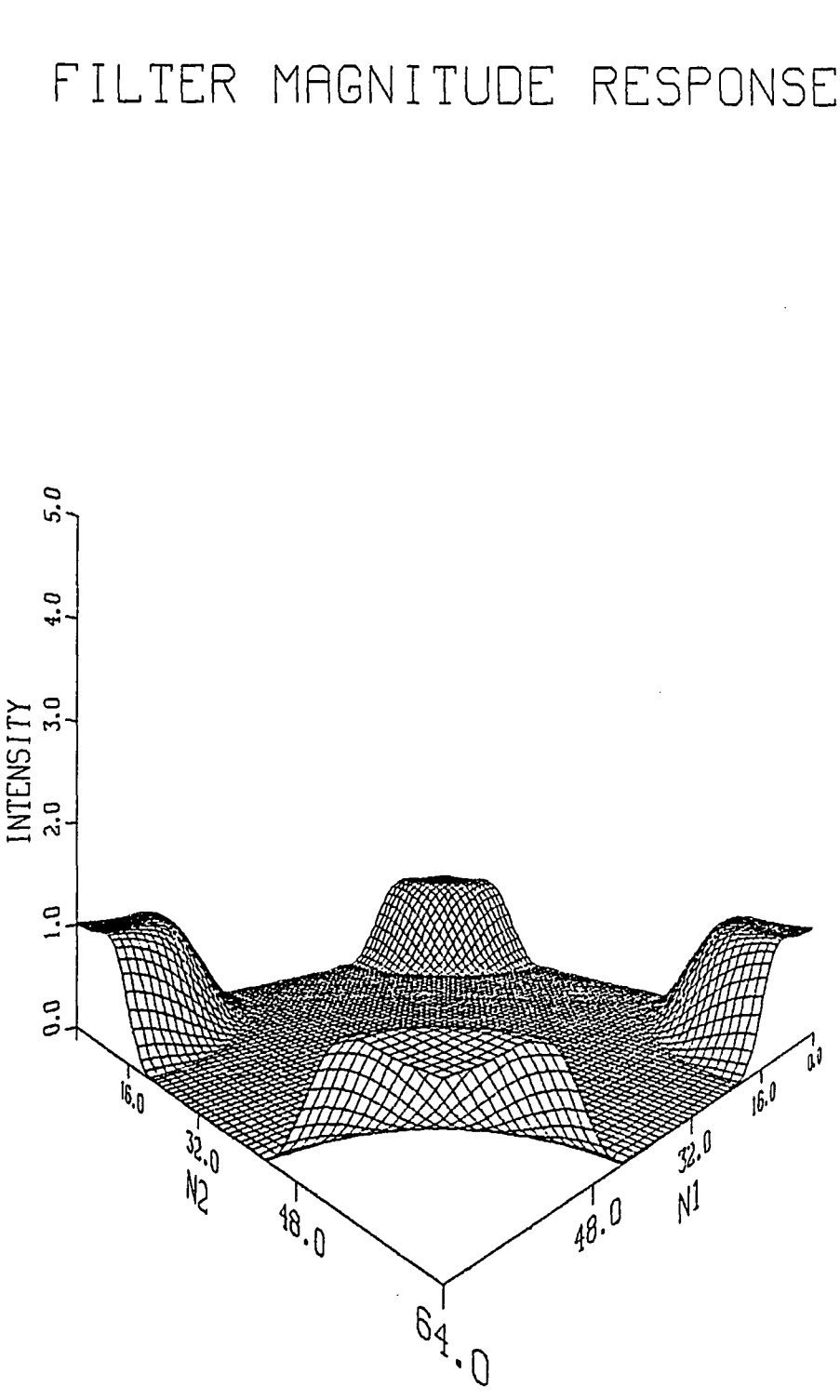


Figure 1.19 Magnitude Response of Fir Low Pass Filter (Hamming)

the magnitude response of the same basic filter, windowed with a Hamming window. Here the passband and stopband have very little ripple, however the transition band is slightly wider than it was with rectangular windowing. Figure 1.20 and 1.21 are the infrared image of Figure 1.5 filtered with Hamming windowed filters having cutoff frequencies of $\pi/2$ and $\pi/4$ respectively. Both images are "smoother" than the original with the second, narrower band, filter producing a very dramatic effect.

Figure 1.22 shows the magnitude response of a 17 by 17 point circularly symmetric high pass filter, windowed with a rectangular window. The cutoff frequency is $\pi/2$. As in the low pass example, the passband and stopband contain ripples caused by the rectangular window. Figure 1.23 is the magnitude response of the same basic filter, windowed with a Hamming window. Again the passband and stopband are much smoother but the transition band is wider. Figure 1.24 is the original infrared image filtered with this Hamming windowed high pass filter. The filtered image reveals edges or spatial discontinuities in the original.

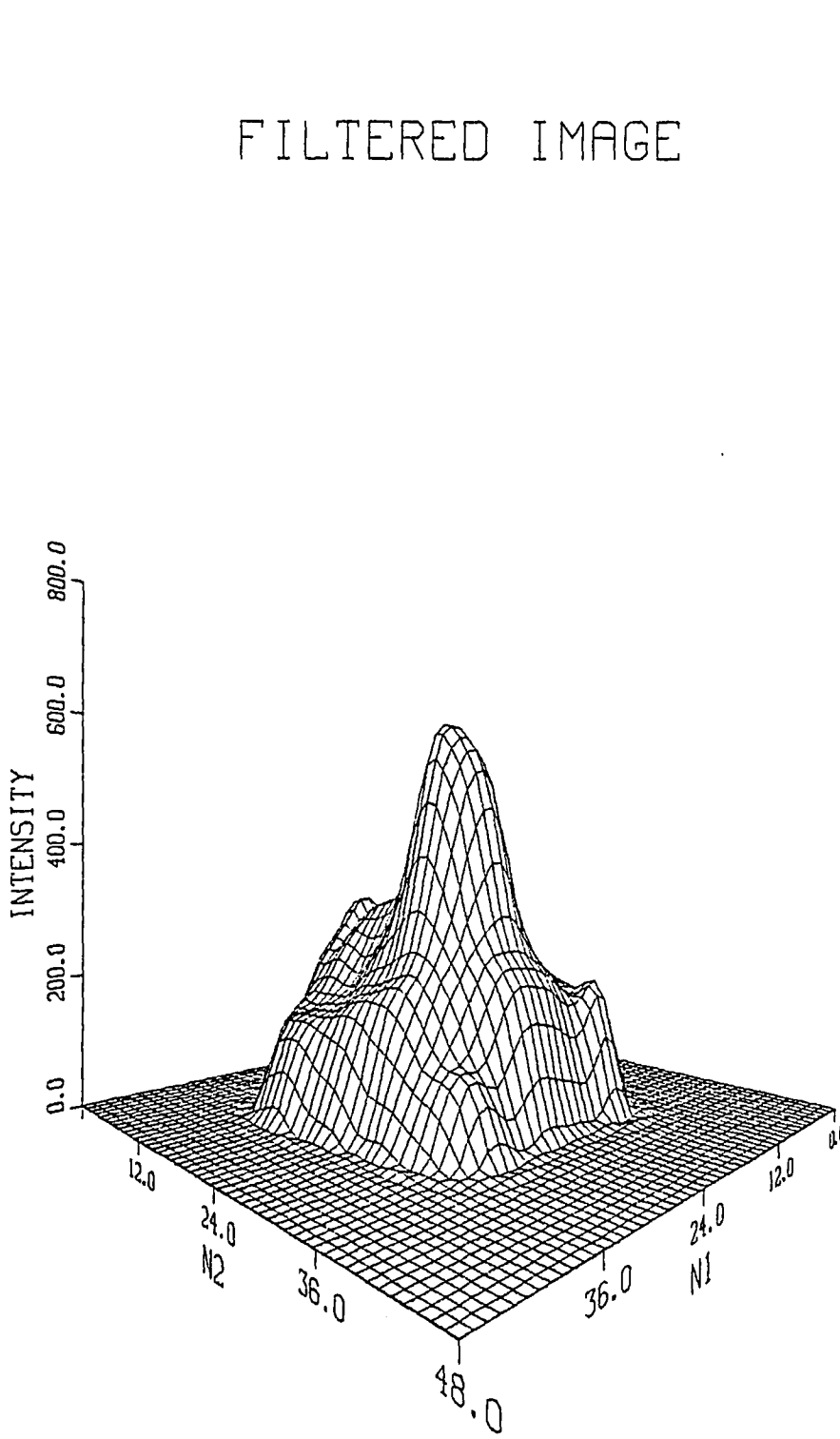


Figure 1.20 Low Pass Filtered Infrared Image (w_c = PI/2)

FILTERED IMAGE

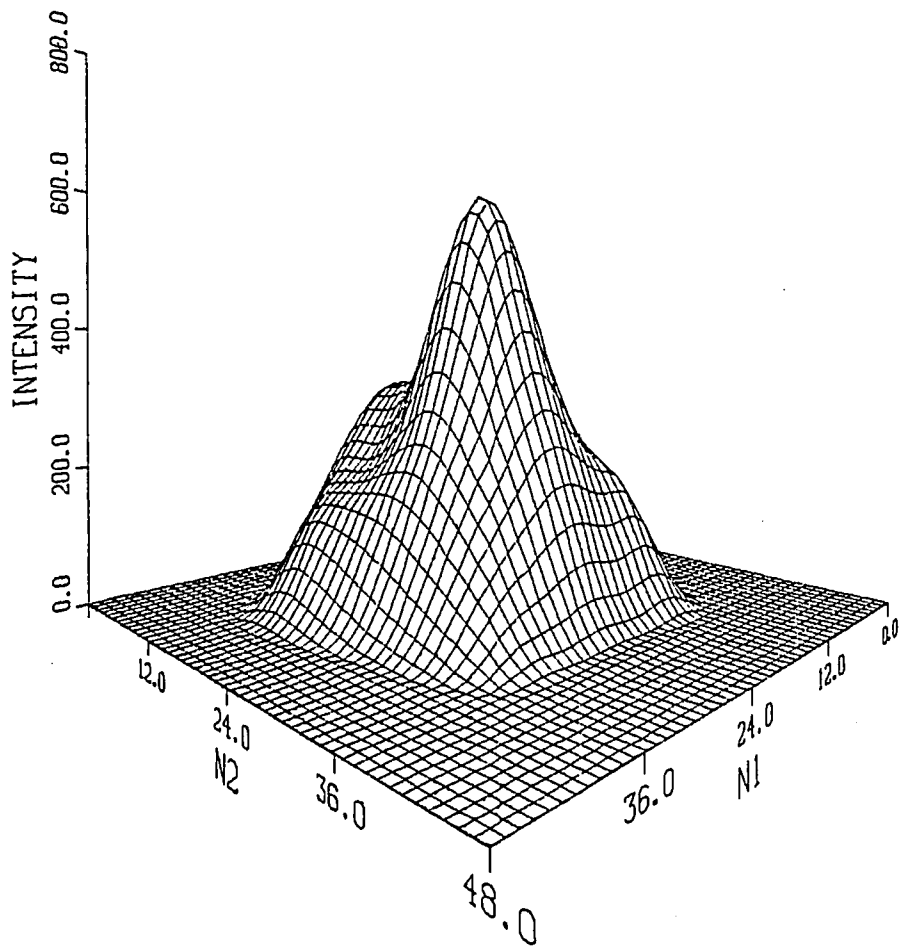


Figure 1.21 Low Pass Filtered Infrared Image (w_c = PI/4)

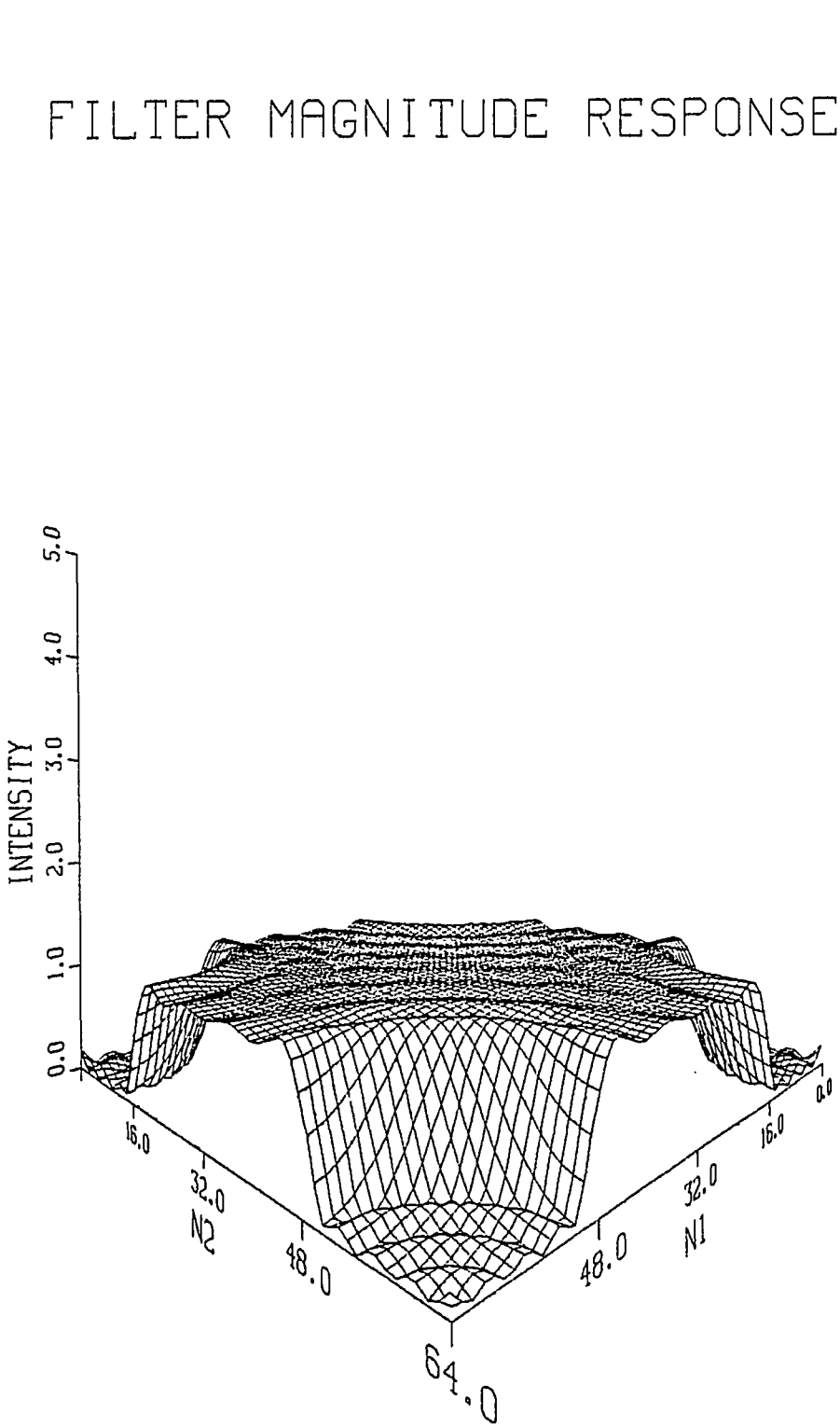


Figure 1.22 Magnitude Response of FIR High Pass Filter (rectangular)

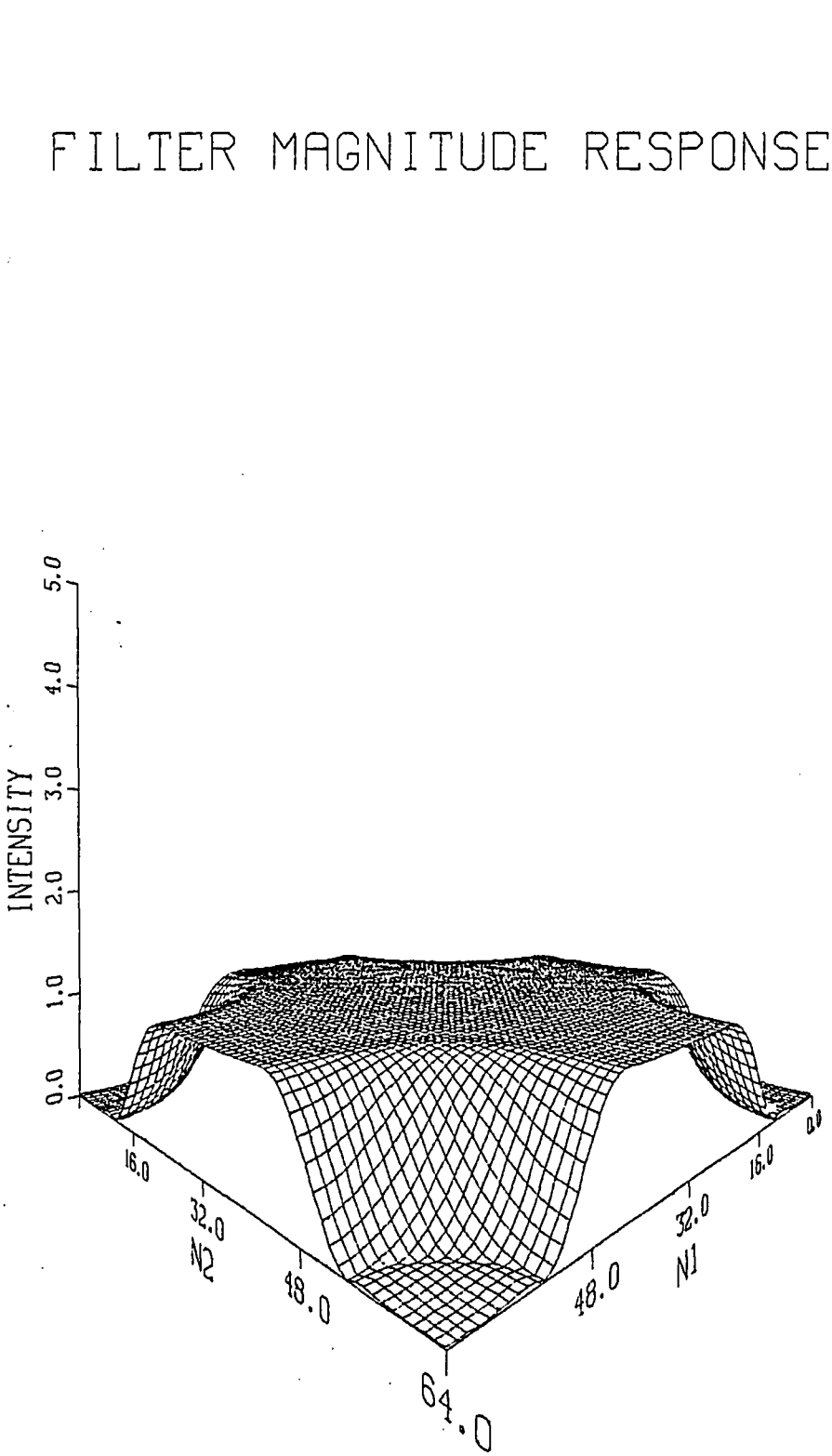


Figure 1.23 Magnitude Response of FIR High Pass Filter (Hamming)

FILTERED IMAGE

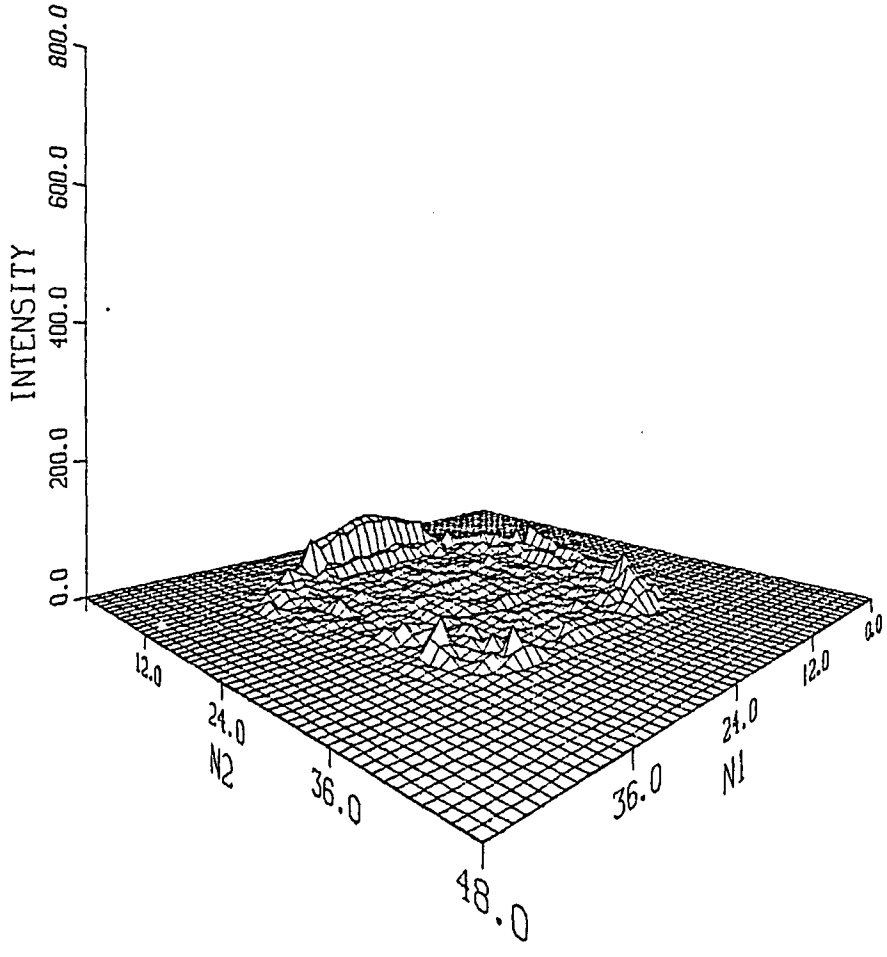


Figure 1.24 High Pass Filtered Infrared Image (w_c = PI/2)

Chapter 2

DESCRIPTION OF THE IMAGE PROCESSING SOFTWARE

Each of the image processing functions described in the Tutorial Description was implemented in software using programs written in FORTRAN programming language. The programs were run interactively on an IBM 3032 computer using actual thermal image data from the infrared camera. The two and three dimensional plots were generated on a VERSATEC plotter using the "DISSPLA"[18] software package.

Flowcharts for each of the image processing programs are provided in the Appendix. A brief description of each of the programs is given below, with the program name in parenthesis.

2.1 Bilinear Interpolation (INTPOL)

INTPOL transforms an N by N image array to an M by M image array using bilinear interpolation. The details of bilinear interpolation were described in the Tutorial Description section.

To make this program as general as possible and to eliminate overlap problems at the image edges, the borders of the interpolated image are defined to line up with the borders of the original image. Hence, each point in the first and last row and column of the new image are filled in by interpolating only points in that row or column.

The remainder of the new image is filled in, point by point as follows. For each point, the four closest points in the original image are determined, and then the spatial relationship in the X direction is determined. Interpolation in X is performed, for two adjacent rows. Next the spatial relationship in the Y direction is determined and interpolation in y is performed using the two results of the previous calculation. This process is repeated until the value for each point in the new image is computed.

2.2 Gray Level Histogram Equalization (HIST)

It was shown in the Tutorial Description that the gray level histogram of an image could be transformed (equalized) to improve contrast.

This program computes the histogram of an image by scanning the image array, using the gray level of each pixel as an index into the histogram array, and adding one to the histogram at that index (gray level). The length of the histogram array is equal to the number of gray levels. Each element contains the number of pixels having that gray level. Finally the histogram is output to a disc file for plotting.

The first step in equalizing the histogram is normalization of the histogram. This is performed by dividing each element of the histogram array by the number of pixels. The result is the pro-

bability distribution function. Next the cumulative probability distribution is computed by calculating a running sum on the initial distribution function. (The cumulative probability distribution is the mapping function). The gray level of each pixel in the original image is used as an index into this mapping function array to derive a new, normalized gray level. The normalized gray level is then scaled and substituted back into the same location in the image array. The new image is then output to a disc file for plotting.

Finally the equalized histogram is verified by computing the gray level histogram of the new (contrast enhanced) image.

2.3 Mask Filtering (IFILT)

IFILT performs spatial domain convolution filtering by sliding a 3 by 3 mask (low pass or high pass) across the image function and performing a convolution sum at each point. The mask type is operator selectable during program execution.

The filtered image is generated as follows. At each pixel, the values of the eight surrounding points and of that pixel are multiplied by the values at the corresponding points of the mask function. These nine products are then added together and divided by the mask normalizing value. This quantity is entered into the new image array at the corresponding pixel location. The process

is repeated at each pixel in the image. As noted in the Tutorial Description, this filtering technique is very effective, yet is computationally very simple to implement and requires relatively little storage space.

2.4 Walsh-Hadamard Transform (WHT)

WHT computes the forward or inverse Walsh-Hadamard transform of an image. The image size and direction of the transform are selected by the operator at run time. A special feature of the program is that, for the forward transform, the operator can input a threshold value and force any coefficients less than that value to zero. The inverse transform can then be computed using this new set of coefficients, and data compression can be optimized.

WHT uses the same algorithm for the forward and inverse transforms, performing in-place computations. The two dimensional transform is computed by performing fast one-dimensional Walsh-Handamard transforms on each row of the image, and then on each column of the row transformed array. In the program this is accomplished by computing the 1-D transform of each row, transposing the matrix, and then transforming each new row. For the forward transform, each coefficient is divided by the number of pixels in the image. After the forward transform is computed the operator can optionally enter a threshold value and force the smallest coefficients to zero. The coefficients of the forward or inverse transform are then output to a disc file for plotting.

As mentioned in the Tutorial Description, the Walsh-Hadamard transform is easy to compute and relatively efficient in terms of storage because no complex numbers are involved. In addition, the same program statements can be used to compute the forward and inverse transforms.

2.5 Fast Fourier Transform (FFTRAN)

FFTRAN computes the forward or inverse two-dimensional fast Fourier transform (FFT). The FFT subroutine is an IMSL library function[19]. As with the 2-D Walsh-Hadamard transform, a one-dimensional transform is performed on each row, the matrix is transposed, and a 1-D transform is again performed on each new row. The image size and direction of the transform are entered by the operator at run time.

FFTRAN includes several experimental features that can be selected by the operator at run time. When computing the forward transform, the smallest coefficients can be forced to zero. The number of coefficients to be forced to zero is entered by the operator. In addition the operator can have the inverse transform computed using the phase spectrum only. This is accomplished by setting each coefficient in the magnitude spectrum to one and then using the phase spectrum to compute the real and imaginary coefficients.

Initially two arrays are defined; one for the real coefficients and one for the imaginary coefficients. For the forward transform, the

image is read into the real array and the transform is computed (with any of the options described previously) by calling the FFT subroutine. The coefficients are returned in a complex array, which is then separated into arrays containing the real and imaginary parts. These arrays are then used to compute both the magnitude and phase spectrum. The four arrays containing the real and imaginary coefficients and the magnitude and phase spectra are written to disc files for plotting.

For the inverse transform, the FFT coefficients are read from disc, a complex array is set up, and the inverse FFT is calculated. A complex array is returned and is separated into the real and imaginary part. Since the original image was real, the imaginary part of the inverse transform is zero and can be discarded. The real part is written to a disc file for plotting.

2.6 FIR Filter Design (FILT2D)

FILT2D is an interactive program used to design two-dimensional finite impulse response filters with the window technique. Either low pass or high pass filters can be designed using Hanning, Hamming, Blackman or rectangular windows.

The operator enters the filter type, window-type, filter cutoff frequency, and filter length, N. Next the one-dimensional window function is computed based on the window type. This one-dimensional

function is converted to a circularly symmetric two-dimensional function (in one quadrant only). The circularly symmetric filter impulse response is computed in one quadrant (using a Bessel function library routine)[19]. This impulse response is then windowed by multiplying by the one-quadrant window function. The remainder of the two-dimensional array is then filled in using mirror image values from the first quadrant. This step guarantees circular symmetry and saves a great deal of computation time.

The filter frequency response coefficients are then computed by using the impulse response as the real part of a complex array and then calling the complex FFT subroutine[19]. The FFT subroutine returns a complex array that is then separated into the real and imaginary parts. These are the coefficients that will be used for filtering in the frequency domain, and so they are written to a disc file. They are also used to compute the filter magnitude and phase spectra which are written to disc files for plotting. The filter impulse response is also written to a disc file for plotting.

2.7 Image Filtering (IMFILT)

IMFILT performs frequency domain filtering on an M by M image using the N by N filter function computed by FILT2D. Low pass or high pass filtering is performed depending on the coefficients in the disc files generated by FILT2D.

The real and imaginary filter coefficients are read from disc and are used to formulate a complex array. The real image array is read from disc, a complex image array is set up, and an FFT is performed on the image to generate a complex array of frequency domain coefficients. The complex filter and image coefficient arrays are multiplied to effect the spatial domain convolution. This new complex array is then inverse transformed to produce the filtered image. The real part of the inverse transform is then written to a disc file for plotting.

The overlap effects of circular convolution were circumvented by padding the filter and image arrays with a suitable number of zeroes before the transforms were computed. Frequency domain filtering is faster than conventional convolution filtering for large arrays, but does require considerably more storage space since several complex arrays must be stored. The greatest efficiency is achieved when the image and filter array dimensions add up to a number that is just less than an integer power of two. In the present example, the infrared image dimension was 32 and the filter dimension was 17, for a total of 49. This required that the complex array dimension be equal to 64. Thus only about 50 percent of the required space was utilized. A filter dimension of 31 could have been used with no additional computation time or storage space.

Chapter 3

DESCRIPTION OF THE APPLICATION

Throughout the previous sections of this paper, images from an infrared camera have been used to illustrate various image processing functions. The application of the camera was described, but only in very general terms. This section presents a more complete description of the application, describes the factors contributing to the degradation of the images, and outlines a method of incorporating image enhancement functions into an existing image processing system.

3.1 Blast Furnace Burden Temperature Measurement

In the steel industry, blast furnaces are used to produce pure iron from iron ore. This is accomplished chemically by combining a reducing gas containing carbon with the ore. Carbon atoms in the gas, produced by heating baked coal (coke) combine with oxygen atoms from the iron ore. The primary products of this reaction are carbon dioxide and pure iron. This process is performed in the furnace by charging alternate layers of coke and iron ore into the furnace from the top, and then forcing an extremely hot (2000F) blast of air up through these layers of material (burden) from the bottom of the furnace. The bottom layers melt, the chemical reactions take place, and pure iron is tapped from the bottom of the furnace.

The efficiency of iron production is determined largely by the flow pattern of the hot air blast up through the material layers. This flow pattern is in turn determined by the distribution pattern of the material layers. The hot blast flow characteristics can be monitored indirectly by measuring the surface temperature of materials at the furnace top. Temperatures will be highest where the volume flow rate is greatest.

The hot blast flow pattern can be optimized (strong central flow) by using the material surface temperature data to (manually or automatically) alter the material charging pattern. This can result in improved furnace efficiency and potentially significant cost savings.

One of the best methods for measuring the material surface temperatures at the furnace top is remote sensing using an infrared camera. An infrared camera is sensitive to infrared energy (heat), and is capable of providing a two dimensional thermal image. In most systems, the object whose temperature is being measured is optically or electronically scanned to produce a continuous electrical signal whose amplitude is a function of the heat intensity.

Bethlehem Steel Corporation uses such a camera on its largest blast furnace in Baltimore, Maryland. Installation of the camera on the furnace top is illustrated in Figure 3.1. The distribution chute

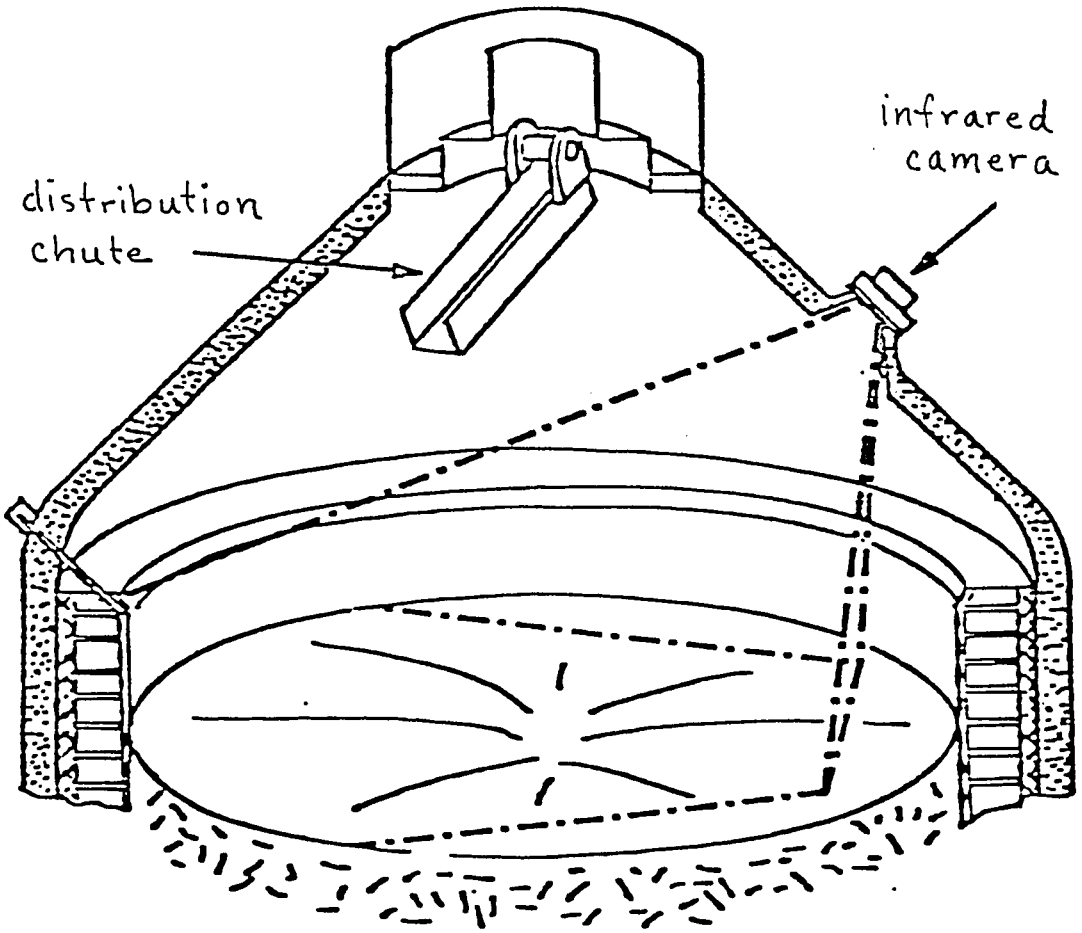


Figure 3.1 Installation of Infrared Camera on Blast Furnace

is used to charge materials into the furnace. The chute is rotated and its vertical angle is varied to deposit the materials in any desired pattern. The position of the infrared camera is also shown in Figure 3.1. The camera views the material surface at an oblique angle which results in geometric distortion of the thermal image.

Charging of materials into the furnace is not continuous. Typically, material is charged for 1-2 minutes and then there is a delay of 1-5 minutes before more material is charged. The average material temperature varies as a function of material charging as shown in Figure 3.2. The temperature drops when new (colder) material is charged into the furnace. After charging is complete, the temperatures rise gradually and reach a quasi steady state value after about 2 minutes. Not only does the average temperature change with time - the two dimensional thermal pattern also changes.

3.2 Factors Contributing to the Degradation of the Thermal Images

Most of the factors affecting the quality of the infrared images are related to the blast furnace ironmaking process itself. For example, as material is charged into the furnace a great deal of gas, steam, and dust is created in the furnace interior. Infrared radiation emitted from the material surface will not transmit through the gas and dust, and hence will not reach the camera. Therefore a good quality thermal image cannot be obtained during

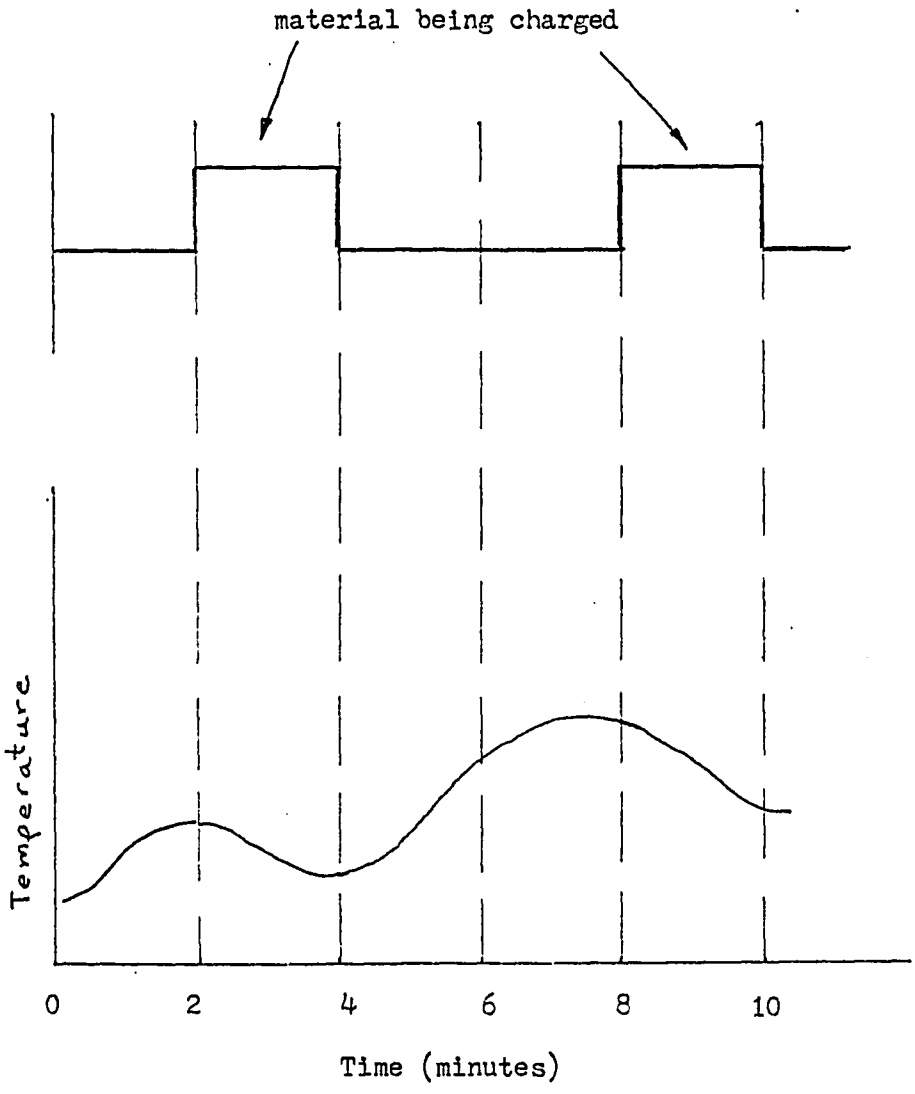


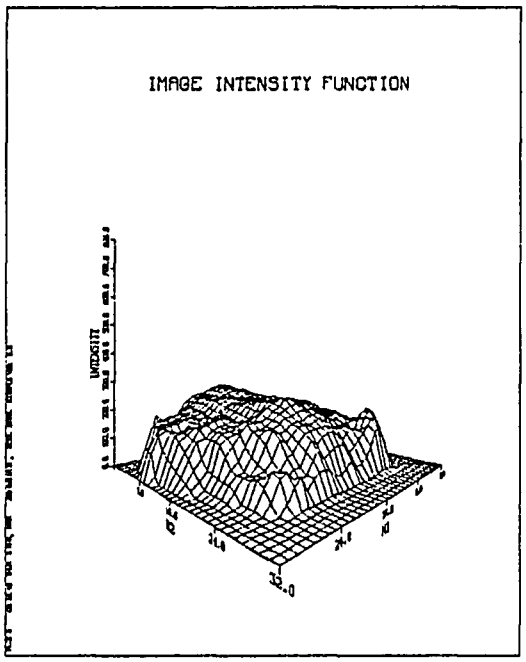
Figure 3.2 Material Temperature Variation with Time

material charging. When charging is complete, some residual dust and particulate matter are still present, however their density varies throughout the image field and with time in a randam manner.

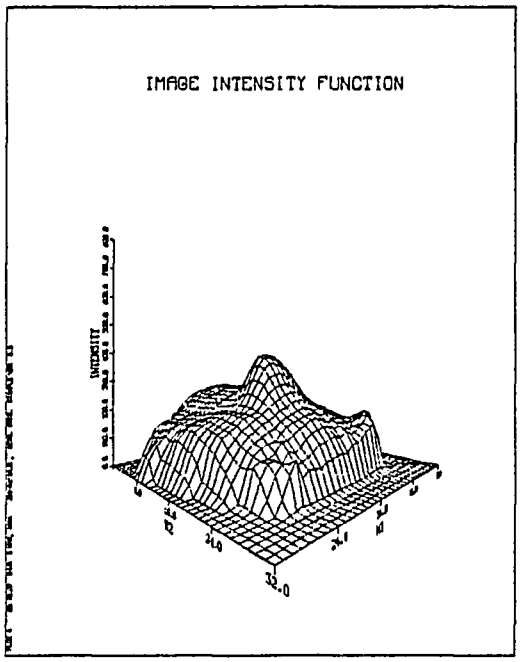
The highest velocity of the hot air blast occurs at the center of the blast furnace. This causes excess gas and dust at the top center of the furnace even after the effects of material charging have stabilized. This central area of gases and dust is referrred to as the "plume". This plume makes it virtually impossible for the camera to obtain a clear view of the material surface on the opposite side of the furnace.

Another factor affecting the quality of the images is the accumulation of dust, dirt, and moisture on the camera lens. As stated previously, these materials impede the transmission of infrared energy.

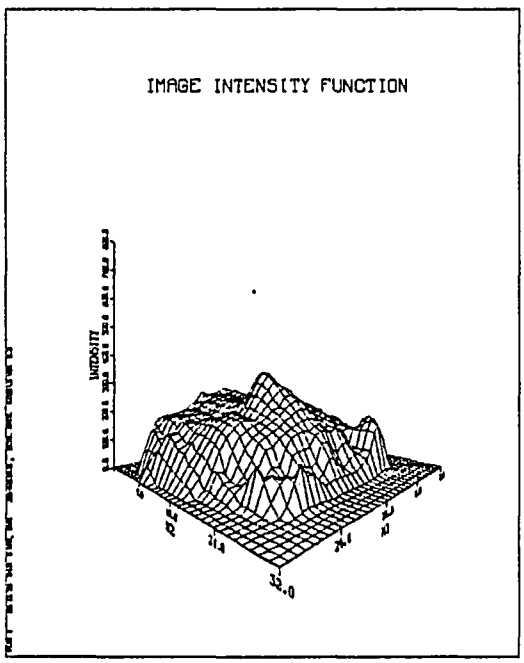
Figures 3.3a through 3.3p illustrate the variation of the thermal pattern of the material surface with time. These unprocessed images were taken at 15 second intervals following the completion of material charging. (Figure 3.3a is 15 seconds after completion of the charge, Figure 3.3b is 30 seconds after completion, and so on). These figures illustrate the effect of the material charging, random gases and dust, and the central gas plume. Clearly the



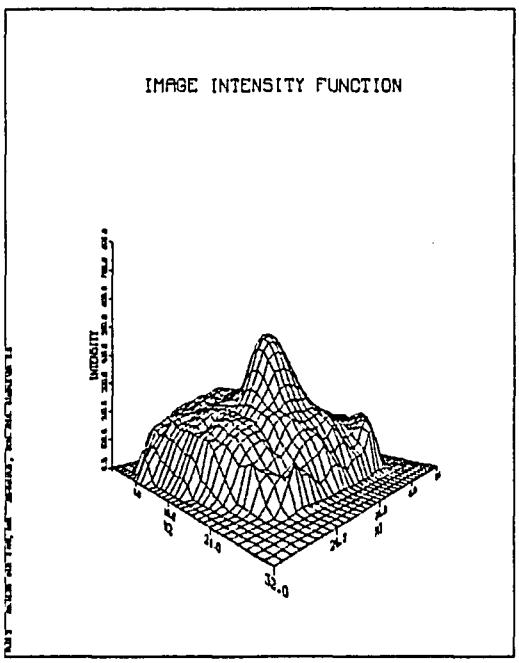
(b)



(d)



(a)



(c)

Figure 3.3 Infrared Images as a Function of Time

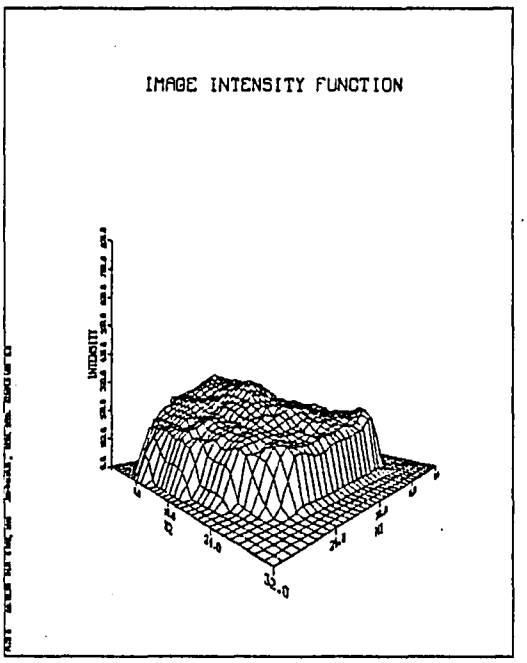


IMAGE INTENSITY FUNCTION

(f)

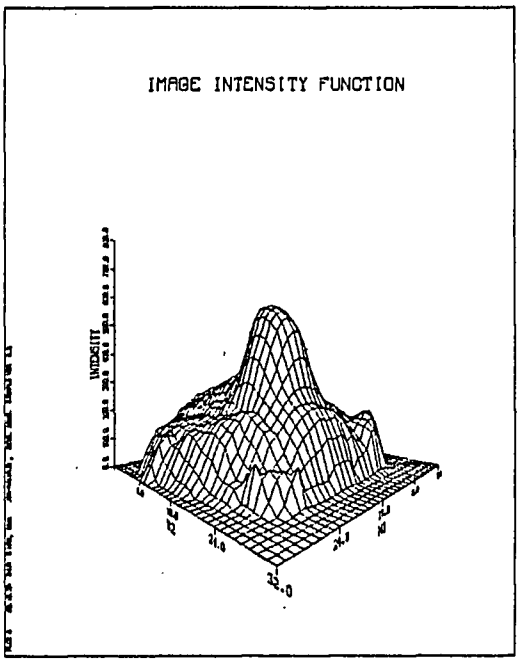


IMAGE INTENSITY FUNCTION

(h)

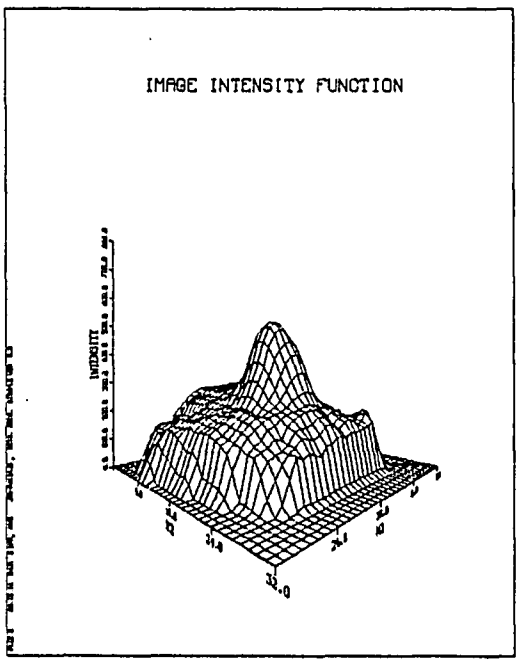


IMAGE INTENSITY FUNCTION

(e)

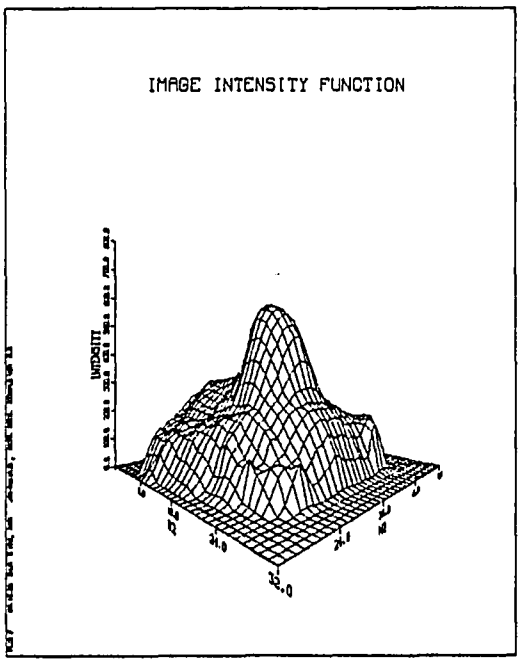


IMAGE INTENSITY FUNCTION

(g)

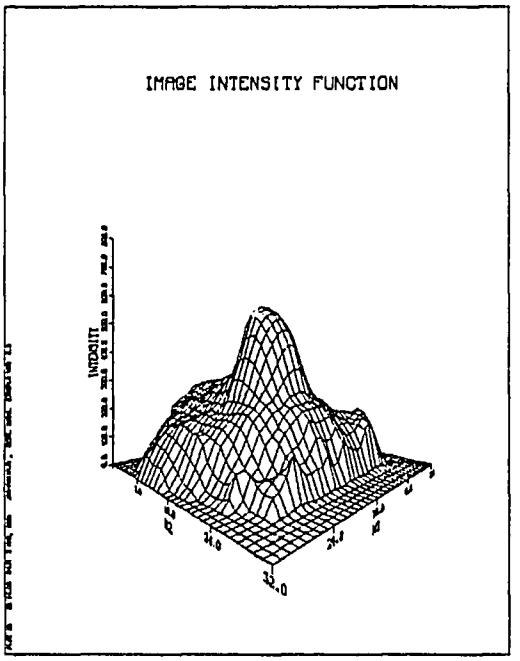


IMAGE INTENSITY FUNCTION

(j)

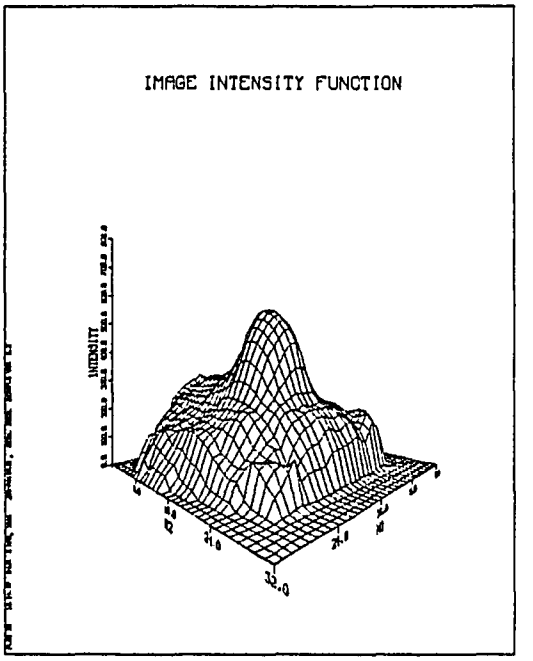


IMAGE INTENSITY FUNCTION

(l)

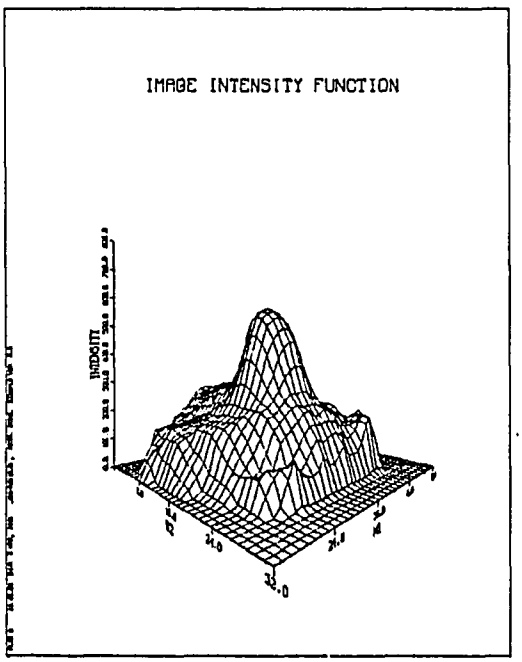


IMAGE INTENSITY FUNCTION

(i)

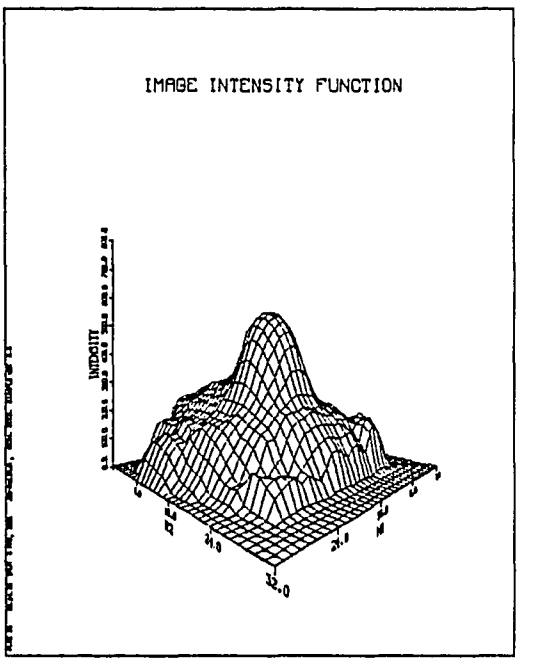


IMAGE INTENSITY FUNCTION

(k)

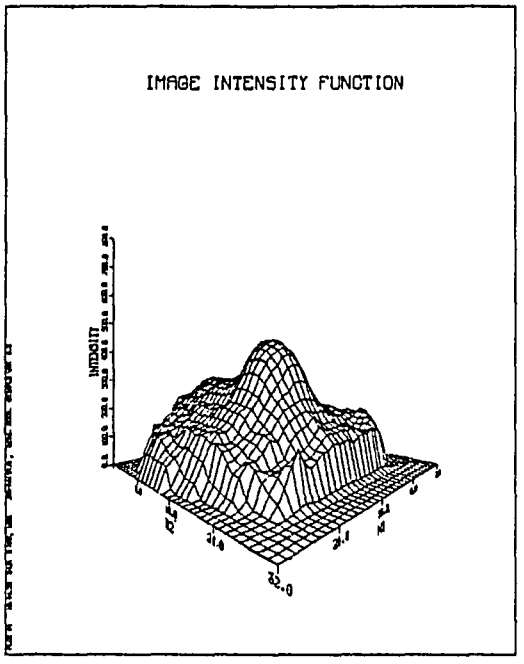


IMAGE INTENSITY FUNCTION

(n)

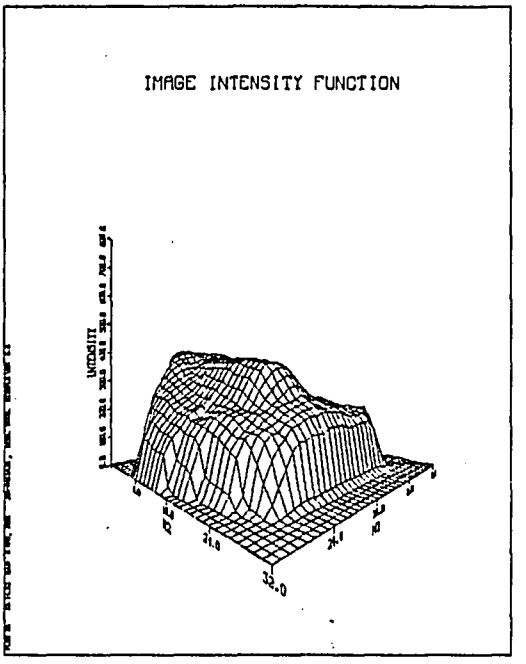


IMAGE INTENSITY FUNCTION

(p)

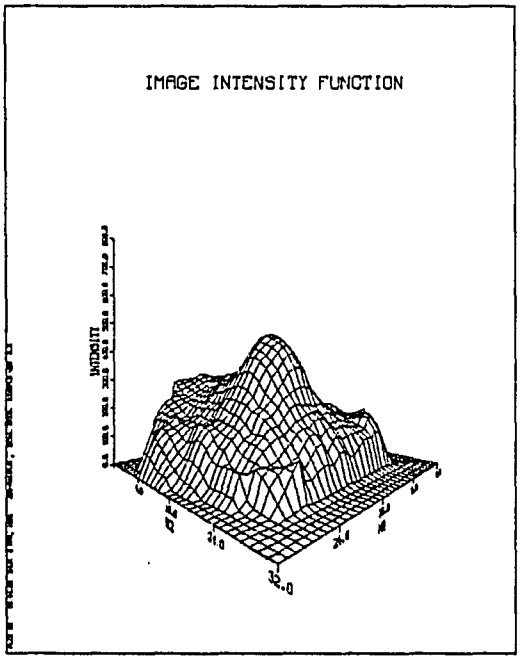


IMAGE INTENSITY FUNCTION

(m)

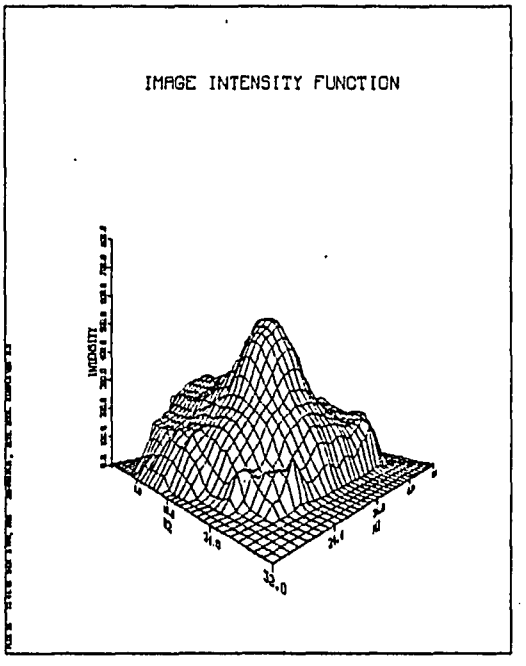


IMAGE INTENSITY FUNCTION

(o)

thermal pattern reaches a fairly steady state condition after about 120 seconds (Figure 3.3h). It is at this time that the thermal patterns are most representative of the flow of gases through the material layers. Just before the image of Figure 3.3p was acquired, new material was charged into the furnace. Hence Figure 3.3p resembles Figure 3.3a.

3.3 A Thermal Image Processing System

In order to provide automatic acquisition and analysis of images from the infrared camera, and to provide a tool to implement image processing functions, Bethlehem Steel developed a computerized image processing system for the infrared camera[20]. This system presently includes some rudimentary image processing functions as illustrated in Figure 3.4.

The analog image from the camera is periodically sampled and quantized using a high speed analog to digital converter. The sampled image is then input to the computer for preprocessing under software control. The preprocessing functions include:

(1) Linearization of the gray level of each pixel to compensate for the nonlinear infrared detector characteristic. (With all infrared detector materials the output signal is a known, but nonlinear, function of the infrared energy impinging on the detector).

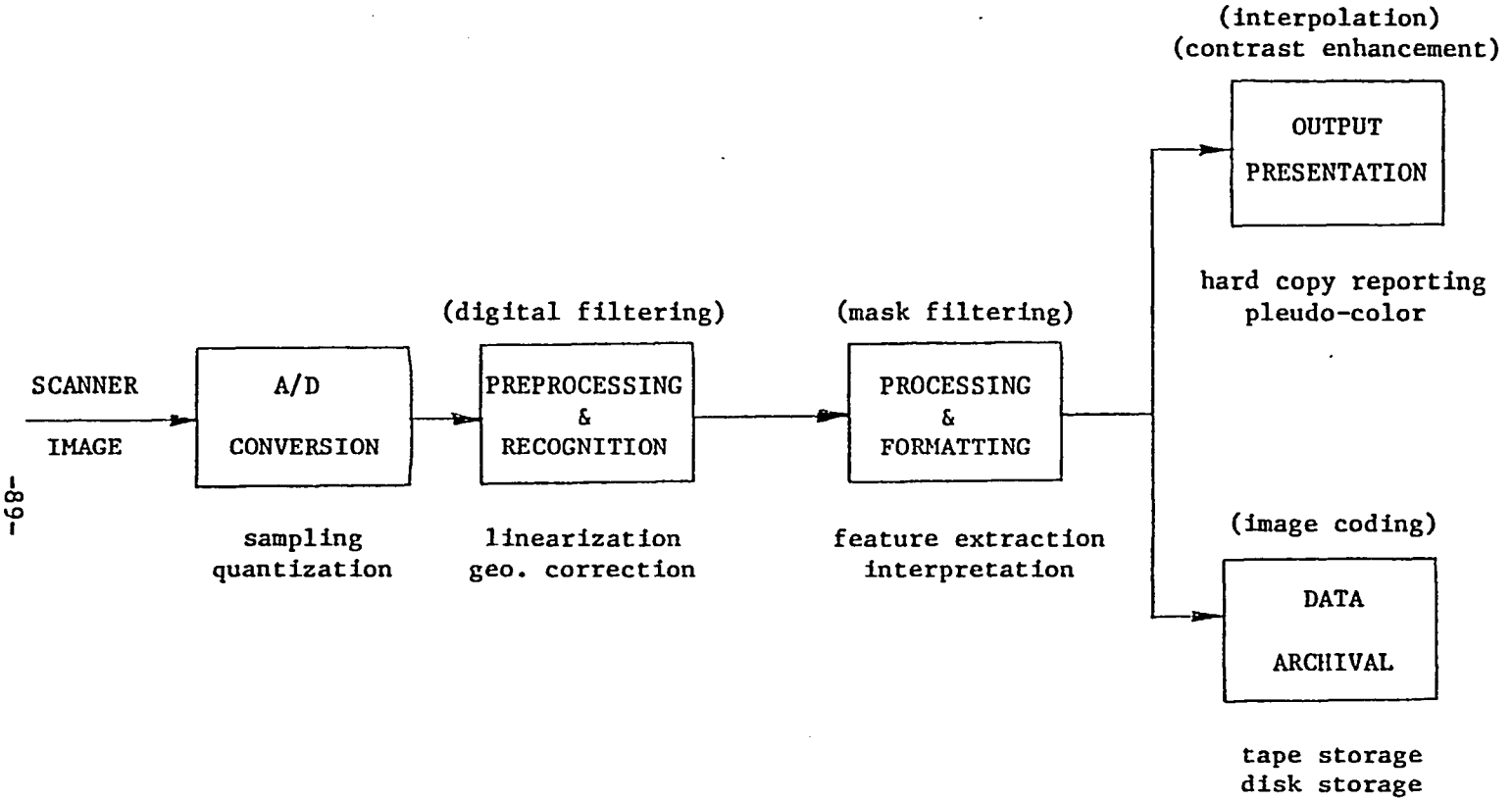


Figure 3.4 Image Processing System

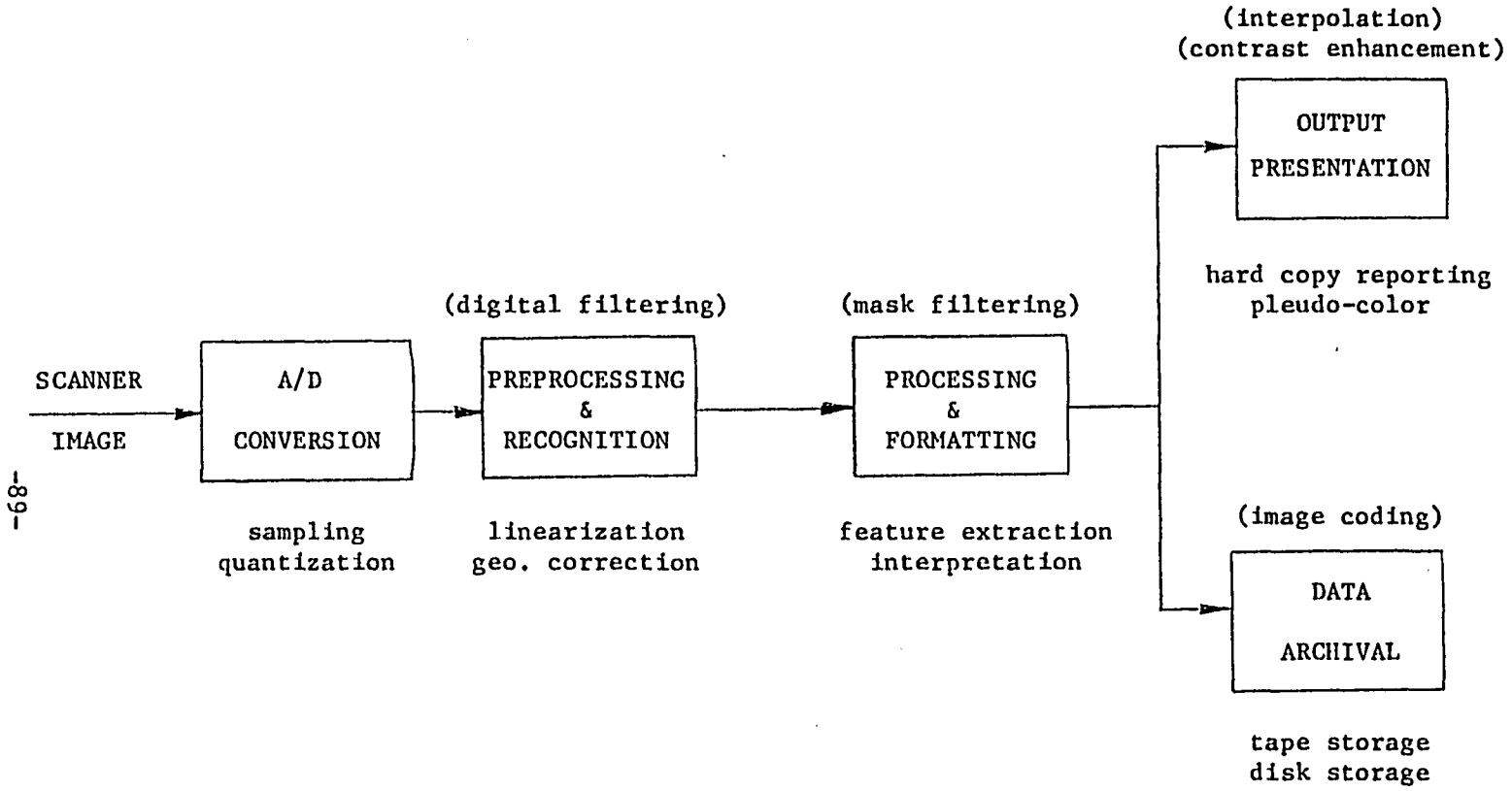


Figure 3.4 Image Processing System

(2) Geometric distortion correction to compensate for the oblique camera view angle and the varying level (depth) of materials in the furnace.

(3) Averaging of several images acquired in rapid succession to reduce the effect of random noise. (This assumes that the spatial registration of successive images is exact).

The digitization and preprocessing of the images is performed in real time.

After the image has undergone preprocessing it is further processed for interpretation and feature extraction. In the present implementation of the system the feature extraction includes:

(1) finding the value of the maximum temperatures,
(2) finding the location of the maximum temperture, and
(3) identifying the shape of the thermal profile along an arbitrary diameter of the furnace.

Each of the features is identified by simply searching the image array, point by point. No other machine interpretation is performed.

The processed images and/or key features are archived on magnetic tape and magnetic disk memory for later recall and observation. In

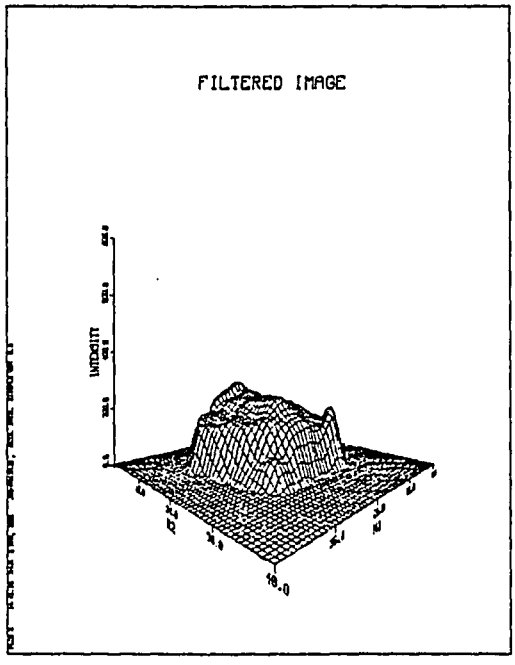
addition, the key image features are printed and are displayed using a psuedocolor display terminal, where colors are used to represent ranges of temperature.

3.4 Potential Improvements to the Image Processing System

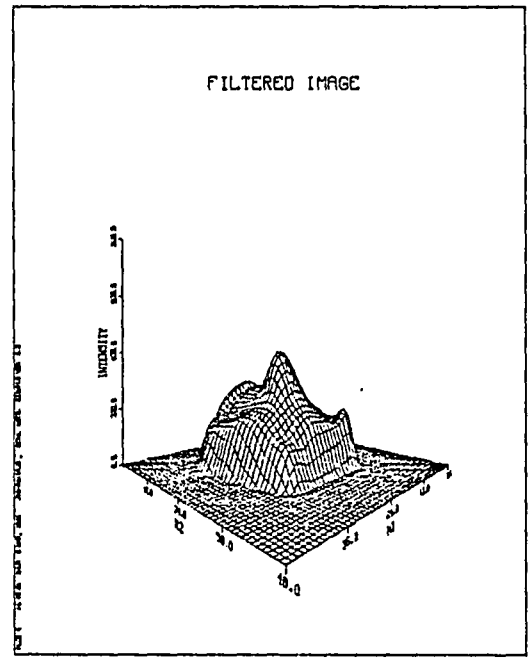
The image processing sytem just described would conveniently lend itself to many of the more advanced functions described in this paper. These functions are listed in parenthesis above the appropriate blocks in Figure 3.4.

The preprocessing operation could include some type of digital filtering for removal of random noise. This could be used to replace the averaging function, which requires extra time since multiple images must be processed, and also requires perfect spatial registration from frame to frame. Figures 3.5a through 3.5p are the original 16 images of Figure 3.3 filtered with a low pass Hanning window FIR filter. Clearly some of the effects of the process noise have been eliminated by the filtering.

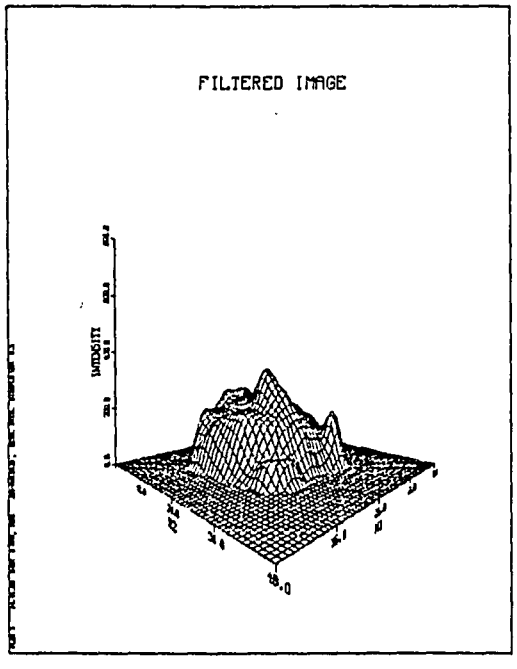
The feature extraction operation could include mask filtering for edge enhancement. This would reveal the areas where the temperature gradient is highest - an important parameter in the operation of the furnace. Figures 3.6a through 3.6p are the 16 original thermal images, filtered with a high pass mask. The intensity is largest in amplitude where the gradient is largest. Note how the temperature gradient changes with time.



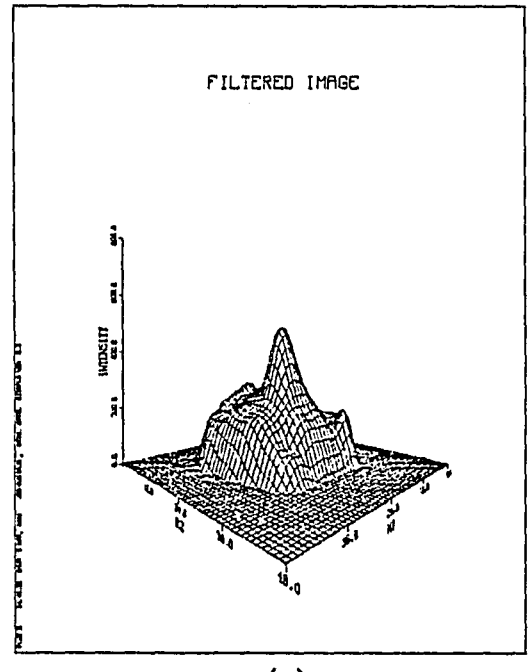
(b)



(d)

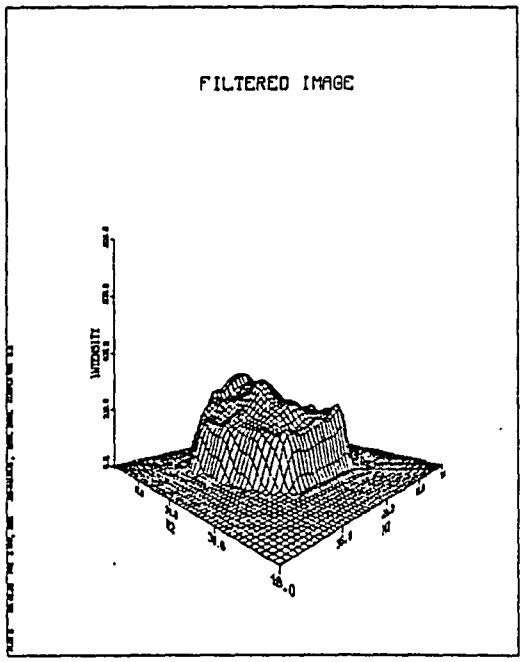


(a)

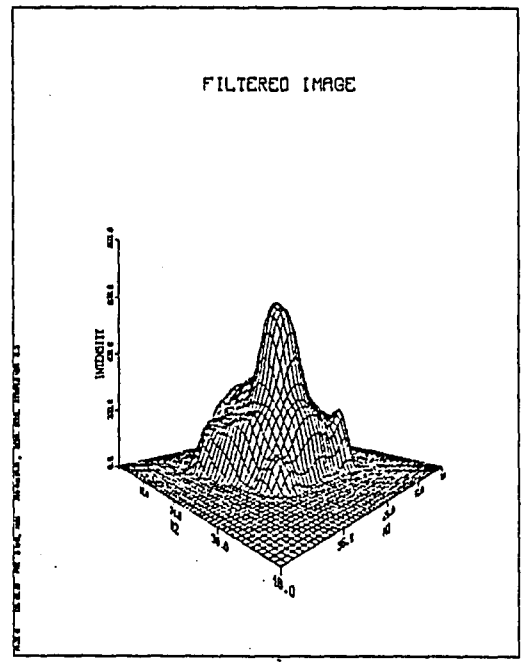


(c)

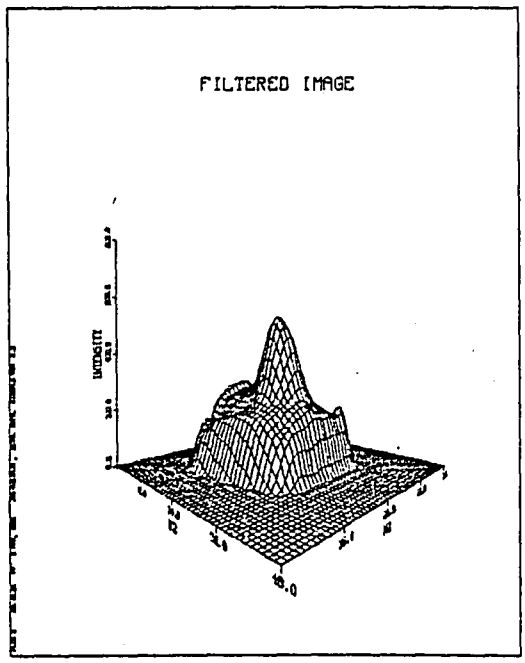
Figure 3.5 Low Pass FIR Filtered Infrared Images


FILTERED IMAGE

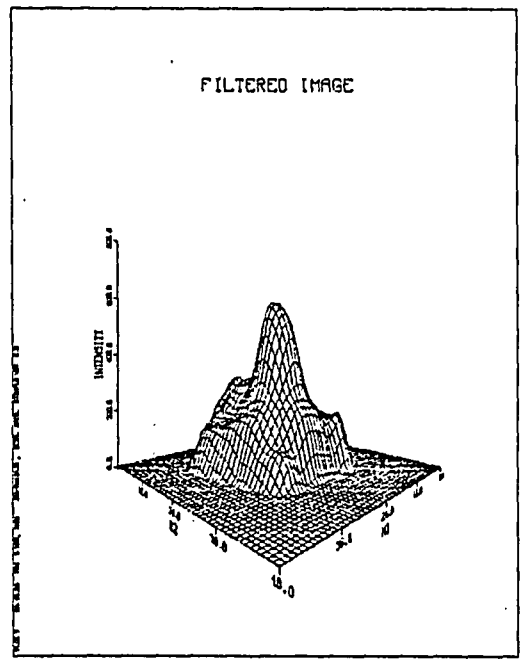
(f)


FILTERED IMAGE

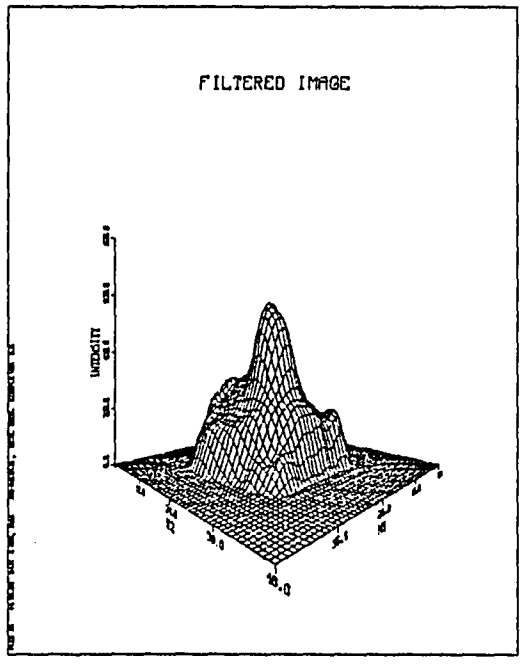
(h)


FILTERED IMAGE

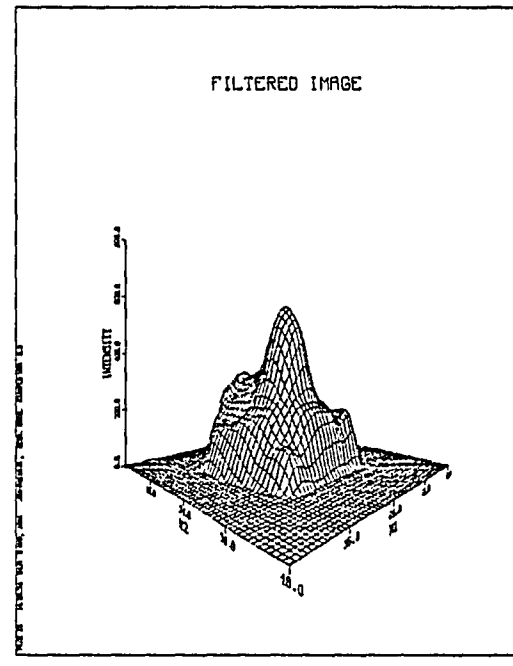
(e)


FILTERED IMAGE

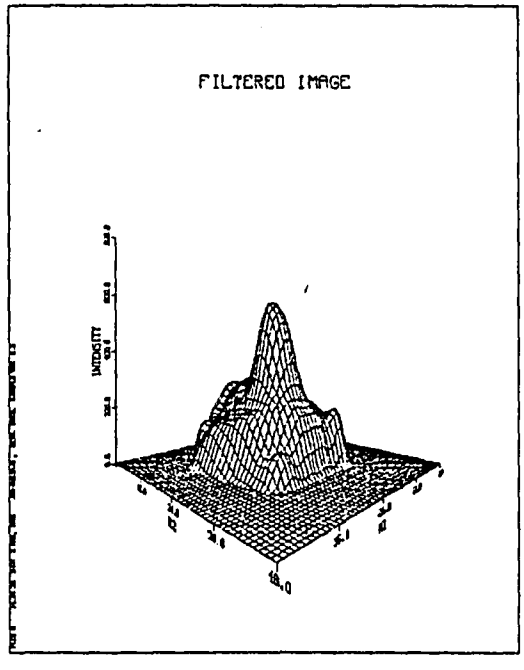
(g)


FILTERED IMAGE

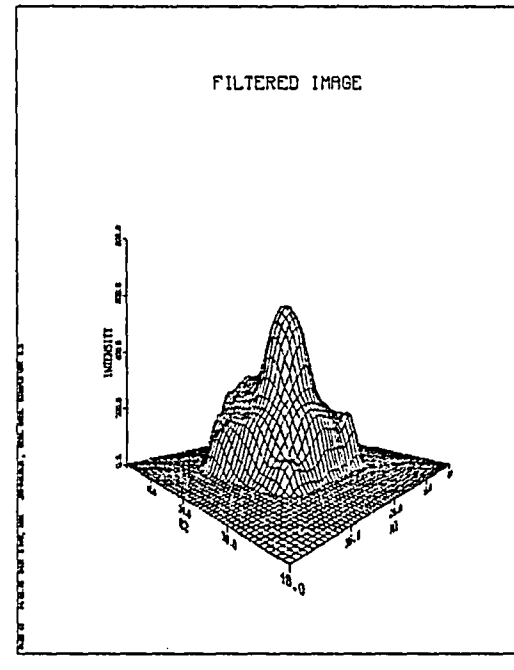
(j)


FILTERED IMAGE

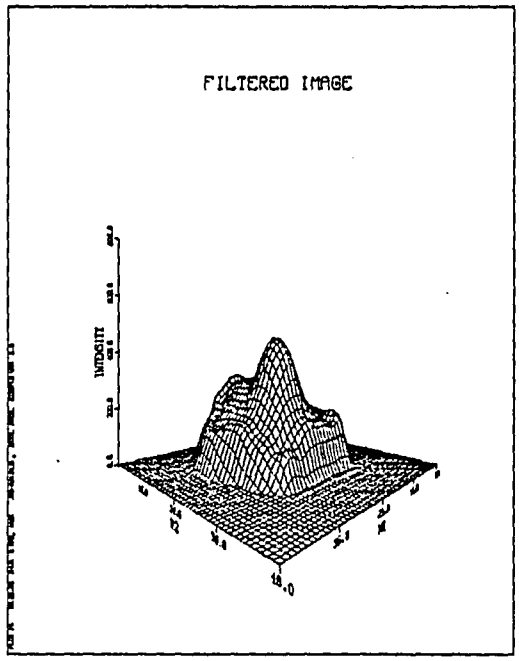
(l)


FILTERED IMAGE

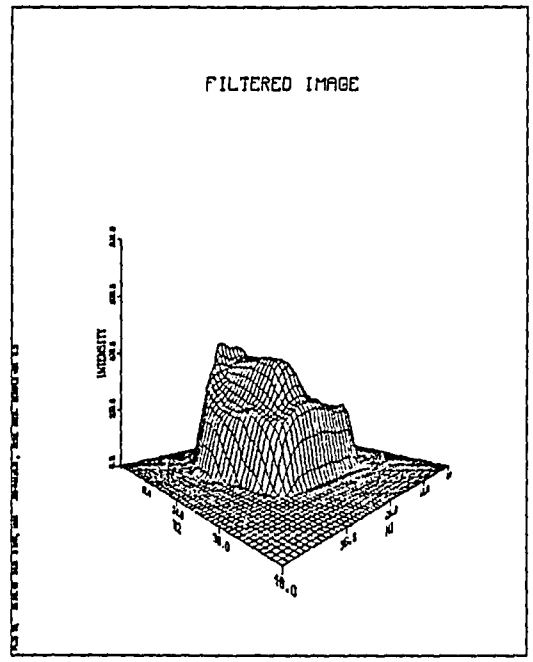
(i)


FILTERED IMAGE

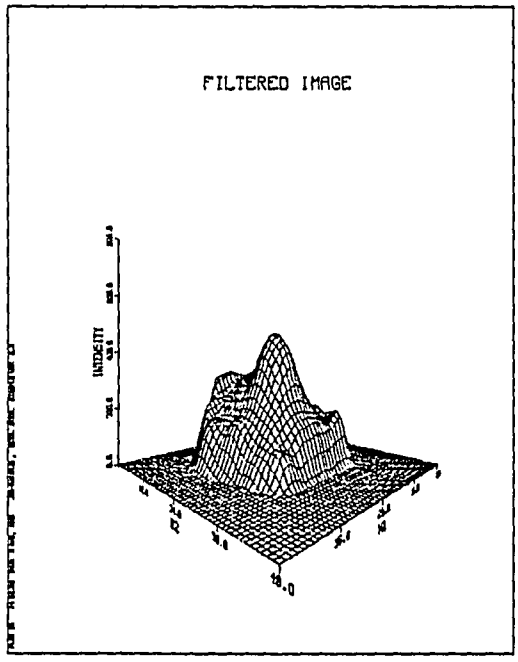
(k)


FILTERED IMAGE

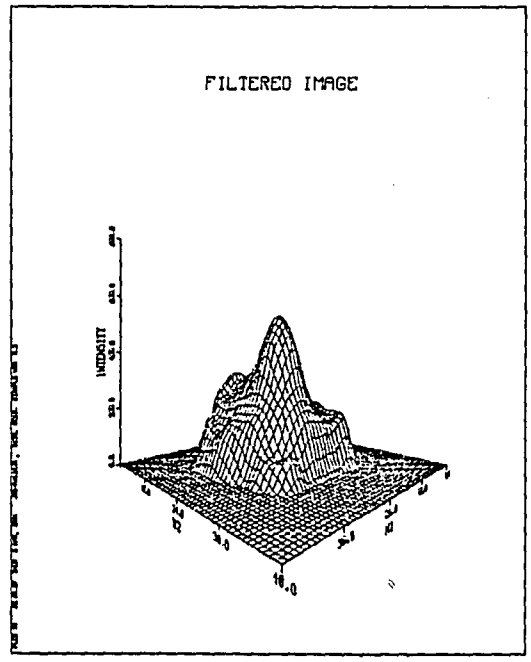
(n)


FILTERED IMAGE

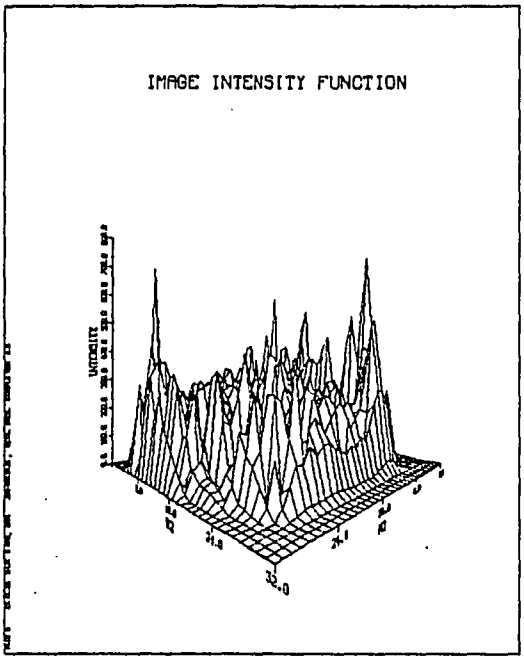
(p)


FILTERED IMAGE

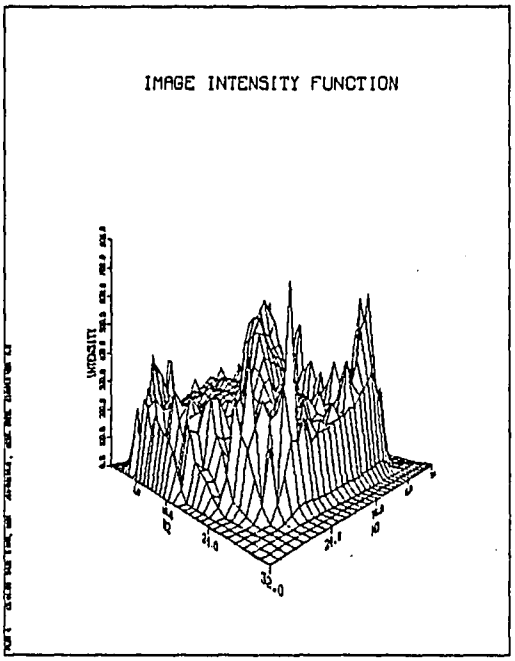
(m)


FILTERED IMAGE

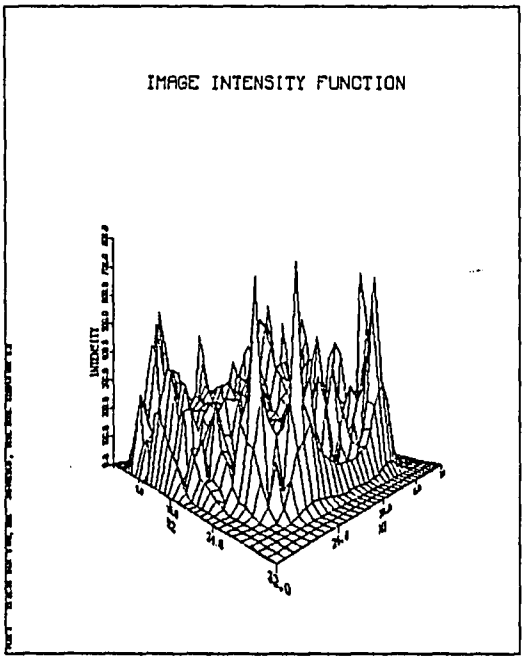
(o)



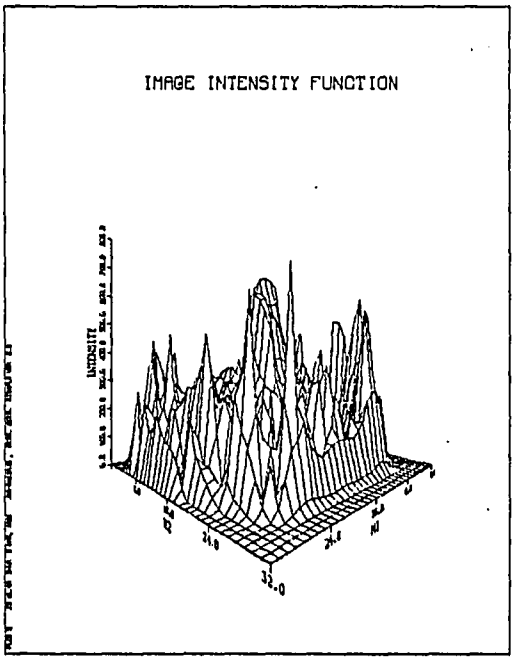
(b)



(d)



(a)



(c)

Figure 3.6 High Pass Mask Filtered Infrared Images

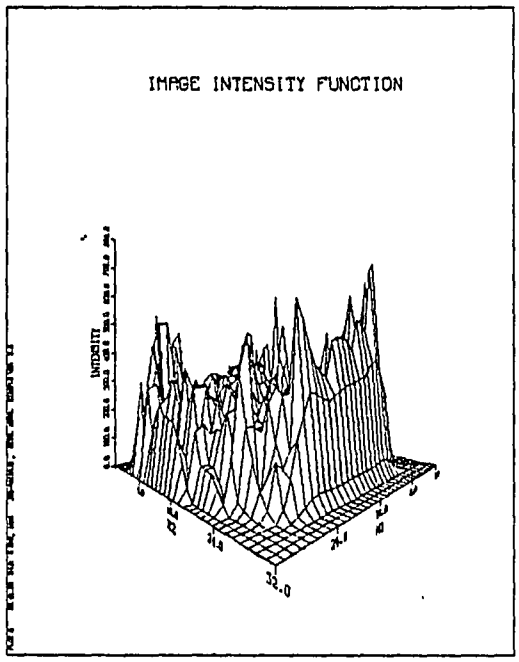


IMAGE INTENSITY FUNCTION

(f)

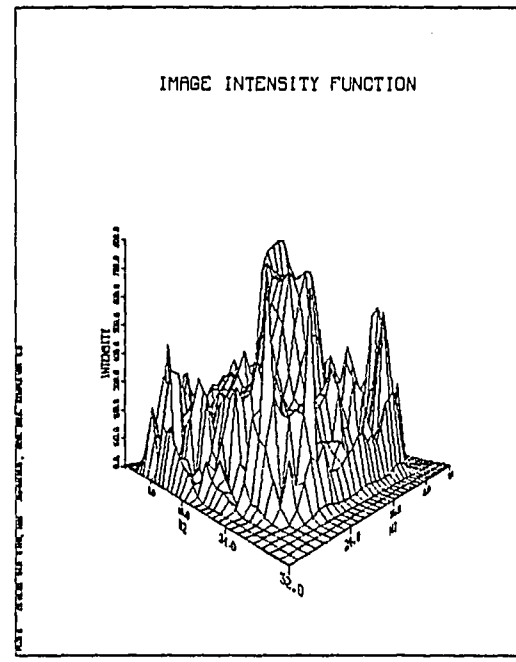


IMAGE INTENSITY FUNCTION

(h)

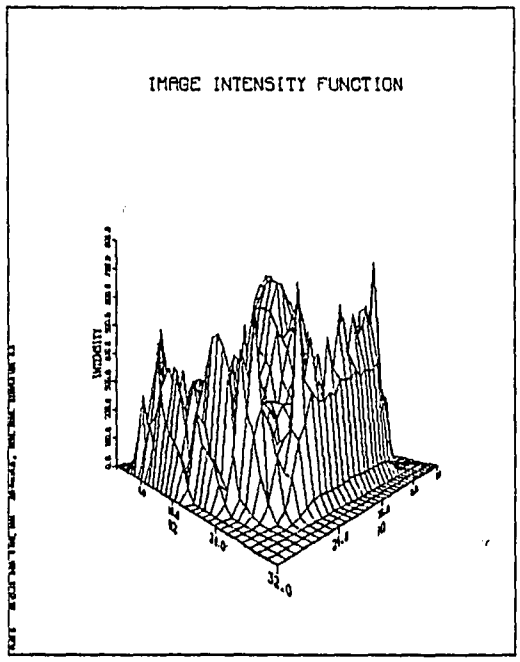


IMAGE INTENSITY FUNCTION

(e)

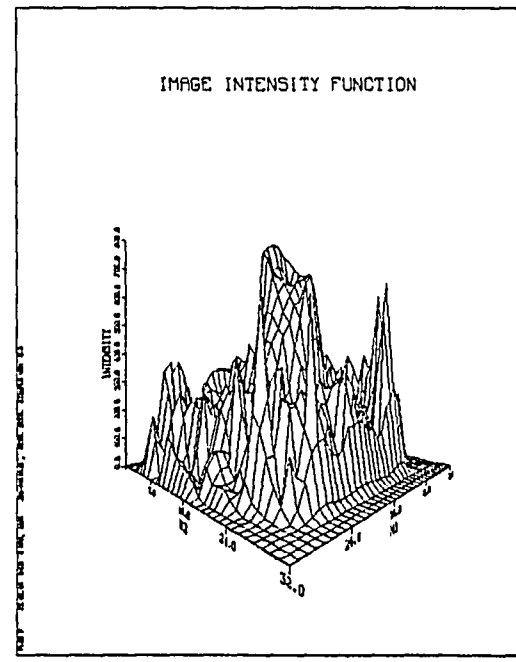


IMAGE INTENSITY FUNCTION

(g)

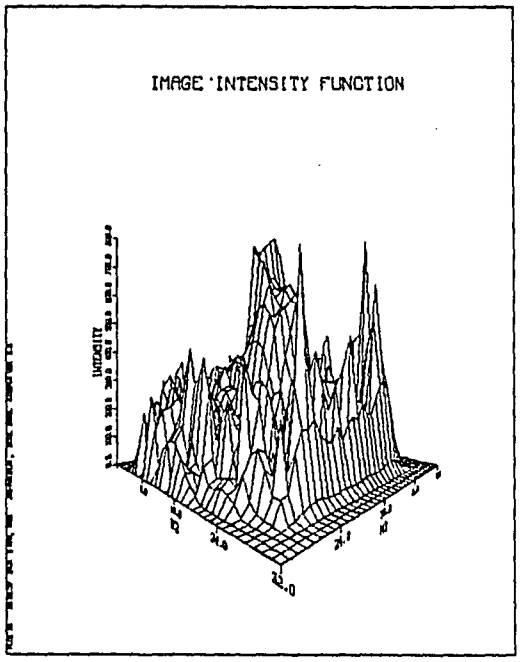


IMAGE INTENSITY FUNCTION

(j)

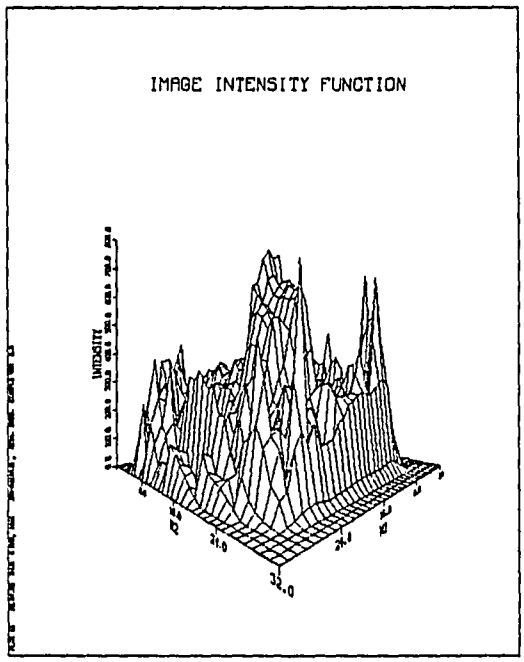


IMAGE INTENSITY FUNCTION

(l)

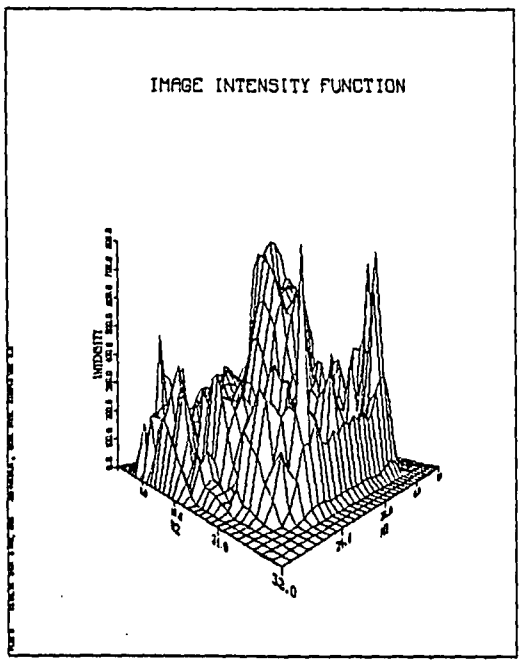


IMAGE INTENSITY FUNCTION

(i)

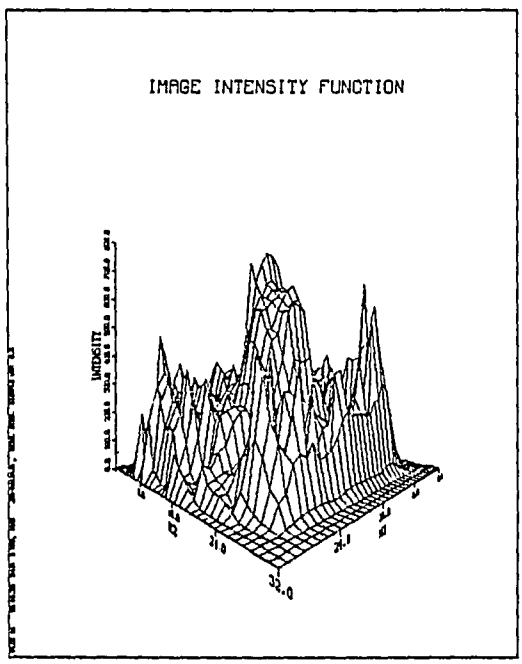


IMAGE INTENSITY FUNCTION

(k)

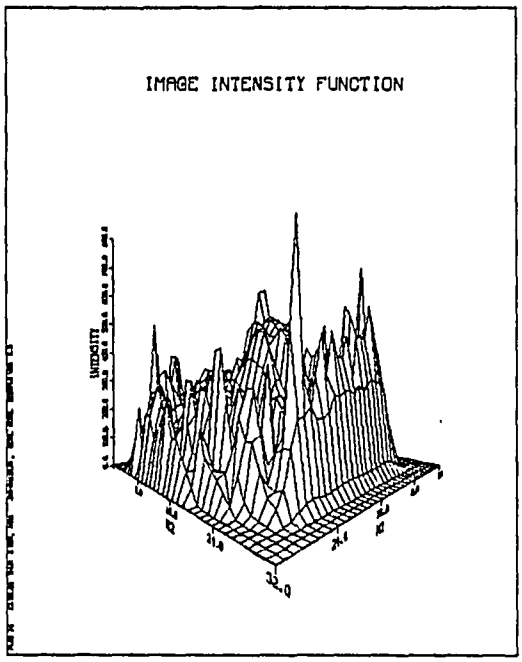


IMAGE INTENSITY FUNCTION

(n)

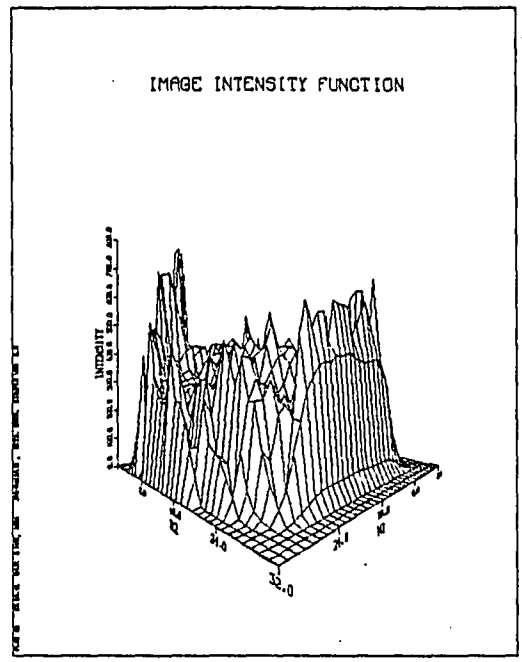


IMAGE INTENSITY FUNCTION

(p)

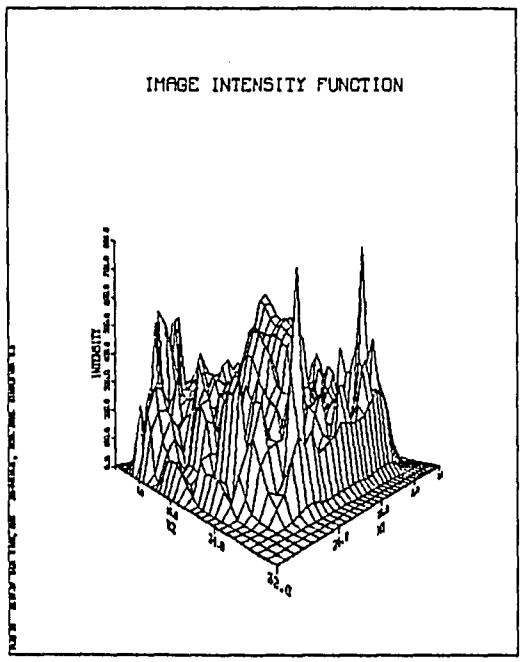


IMAGE INTENSITY FUNCTION

(m)

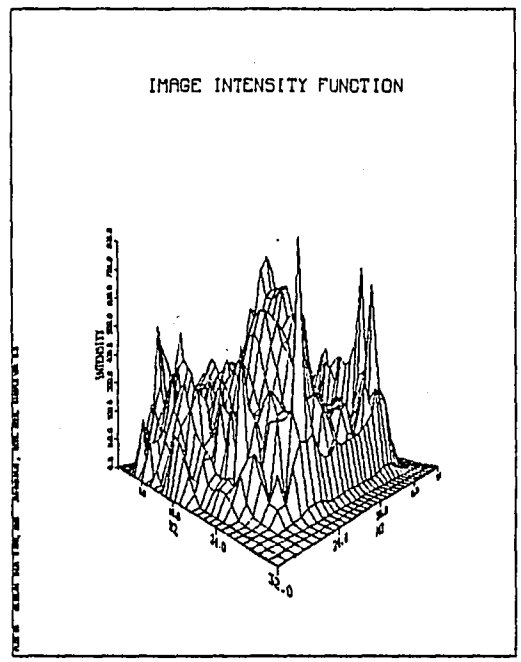


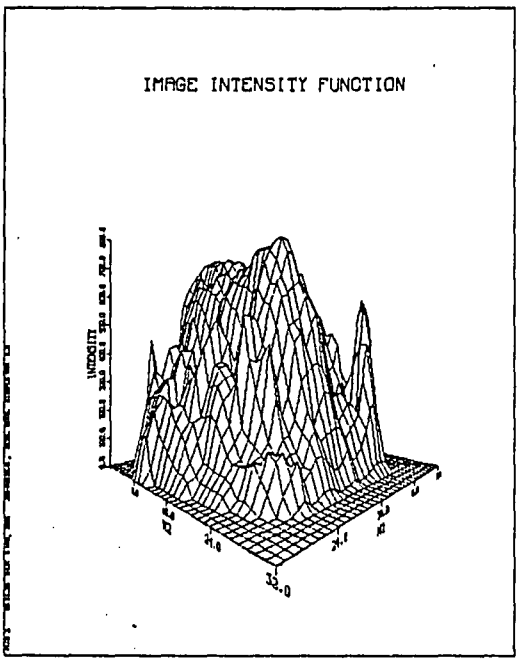
IMAGE INTENSITY FUNCTION

(o)

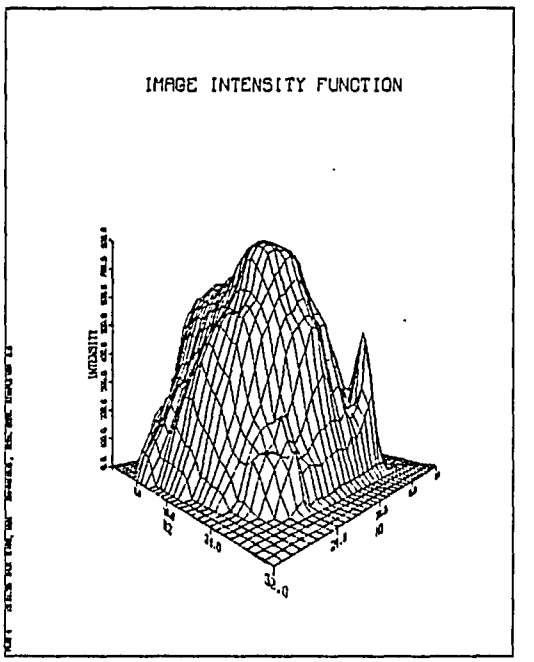
The output presentation function could include image interpolation (smoothing) and or contrast enhancement using histogram equalization. Either of these additions would improve the subjective quality of the images for viewing. Figures 3.7a through 3.7p are the original 16 infrared images after contrast enhancement. Here the full dynamic range (all gray levels) is utilized and low contrast detail is improved. These images could be displayed on a black and white monitor with a continuous gray scale.

Perhaps the most useful addition to the existing image processing system would be in the area of image coding. It presently requires 625 words of computer memory to store one, 25 by 25 thermal image. This is because the gray level of each individual pixel must be retained. This storage requirement could be reduced by transforming the image and saving only the most significant transform coefficients. Using the Walsh-Hadamard transform, the storage requirement can be reduced by 70 per cent, to approximately 200 words. Figures 3.8a through 3.8p are the inverse Walsh-Hadamard transforms of the 200 most significant WHT coefficients of each of the original 16 images. The distortion caused by the elimination of 70 per cent of the coefficients is not significant in any of the images.

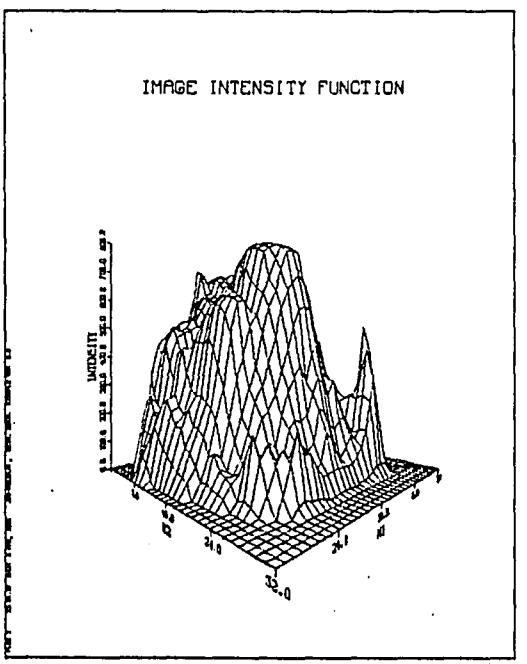
The image processing functions described above could be used individually or in any combination. Each of the functions provides a unique benefit and could be implemented in software that would require relatively little computer time or space.



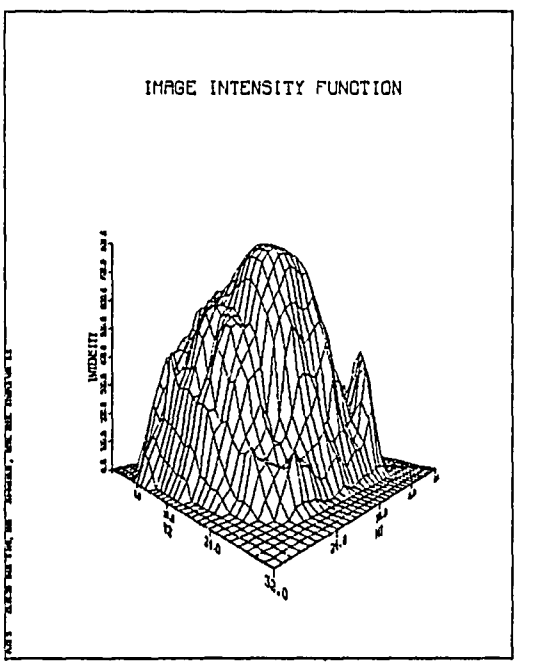
(b)



(d)



(a)



(c)

Figure 3.7 Infrared Images with Enhanced Contrast

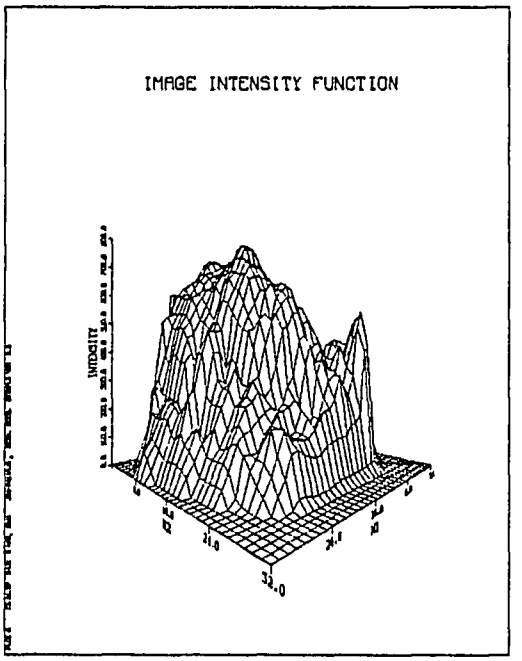


IMAGE INTENSITY FUNCTION

(f)

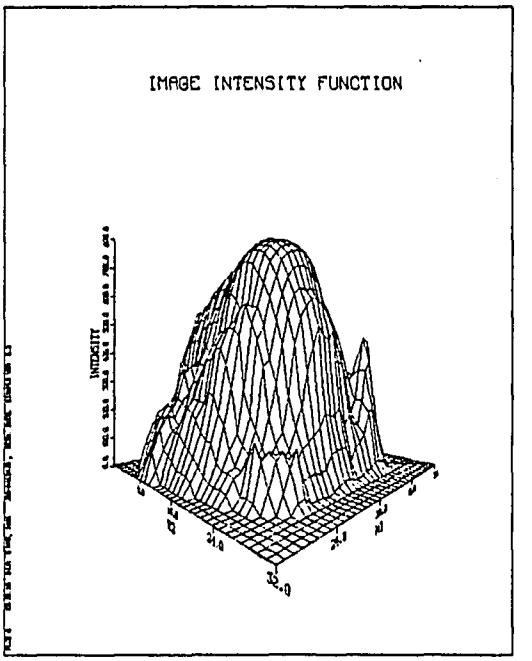


IMAGE INTENSITY FUNCTION

(h)

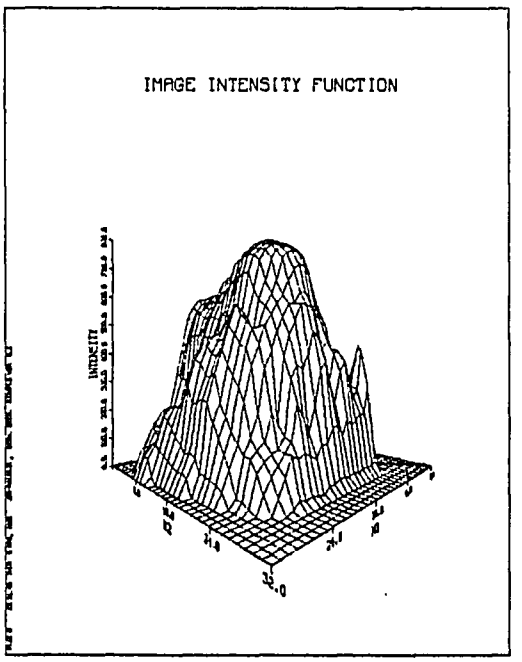


IMAGE INTENSITY FUNCTION

(e)

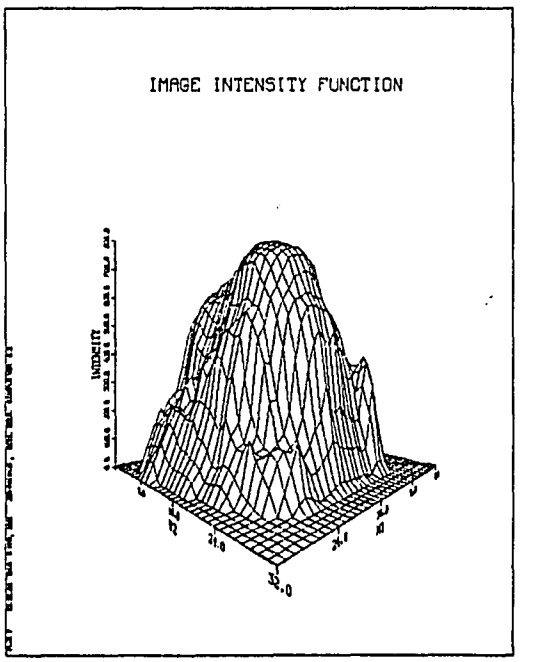


IMAGE INTENSITY FUNCTION

(g)

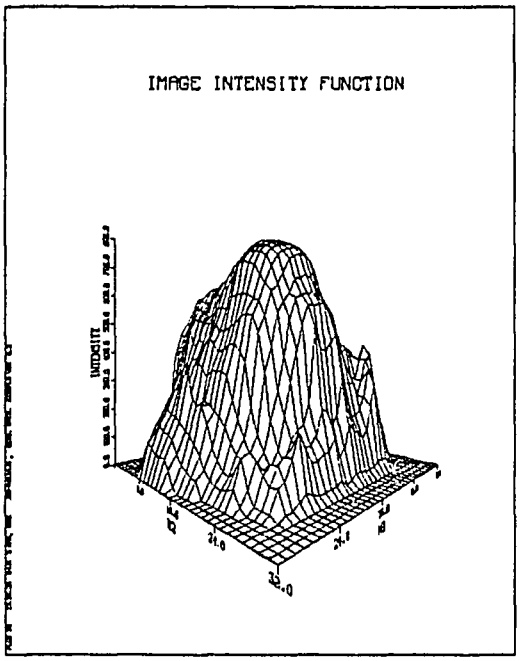


IMAGE INTENSITY FUNCTION

(j)

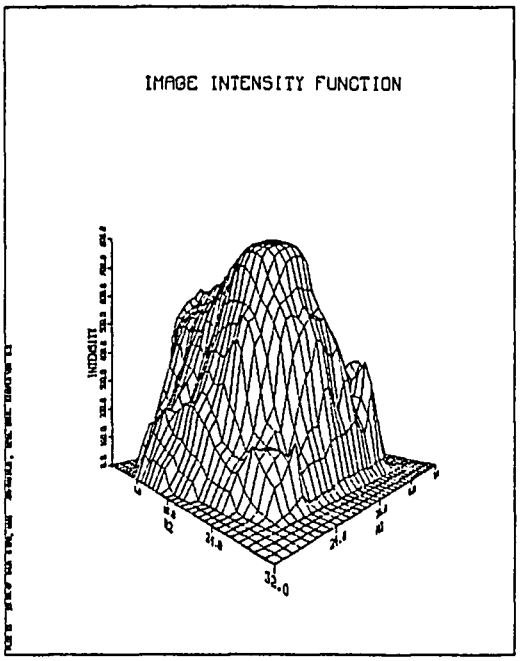


IMAGE INTENSITY FUNCTION

(l)

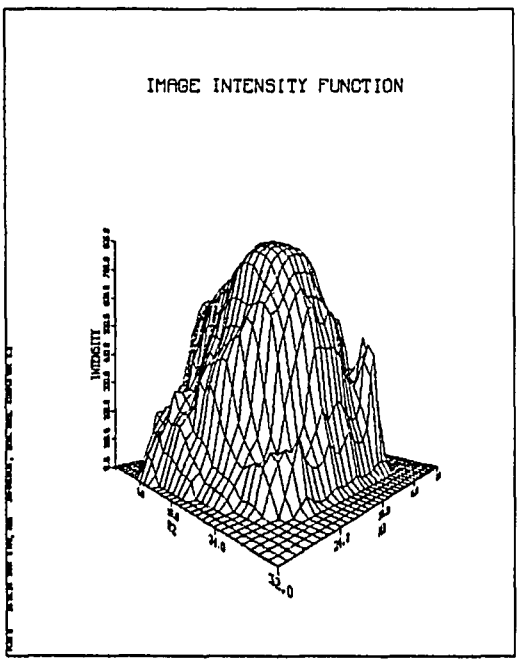


IMAGE INTENSITY FUNCTION

(i)

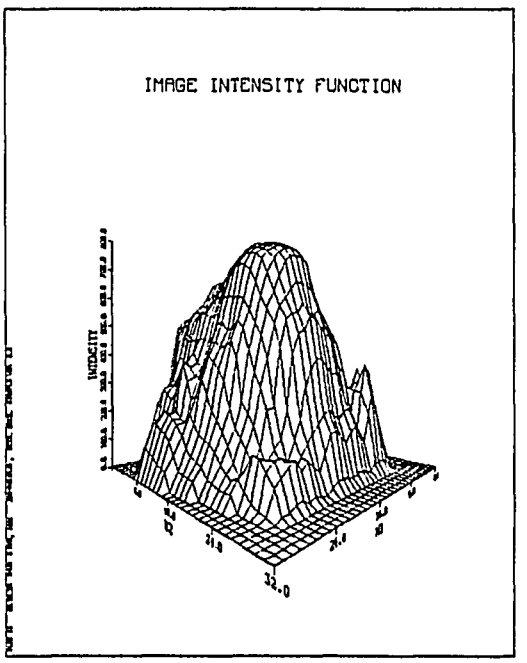


IMAGE INTENSITY FUNCTION

(k)

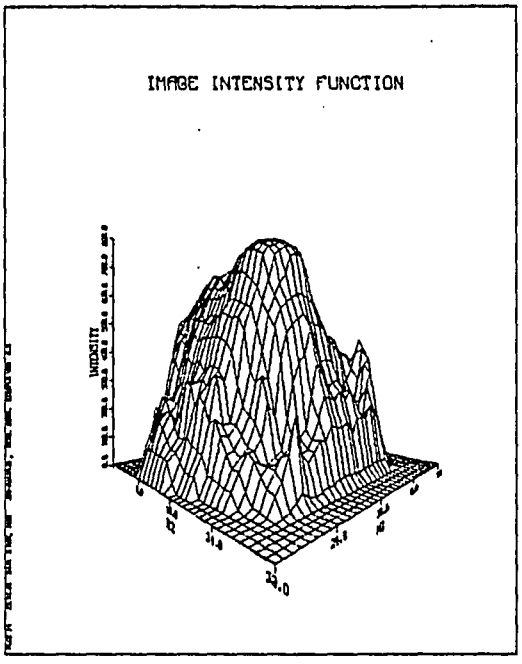


IMAGE INTENSITY FUNCTION

(n)

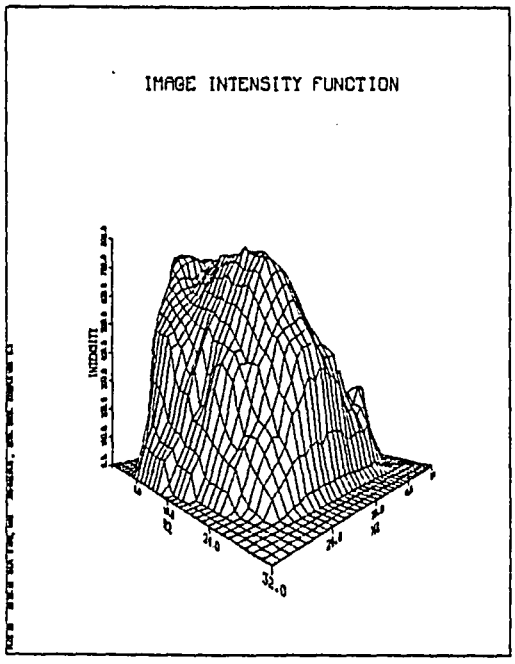


IMAGE INTENSITY FUNCTION

(p)

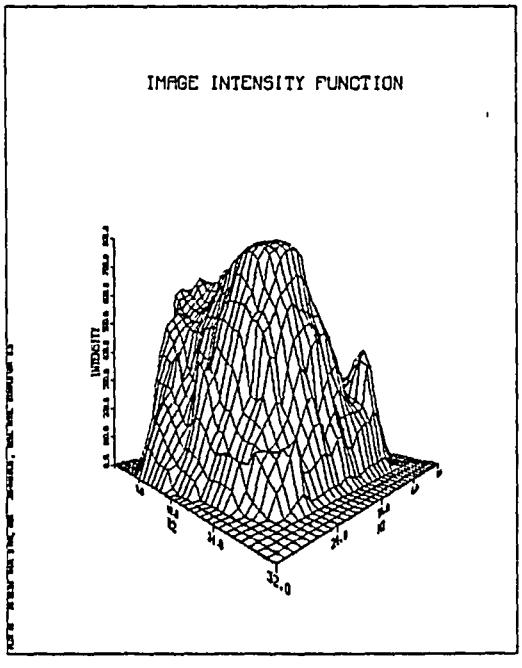


IMAGE INTENSITY FUNCTION

(m)

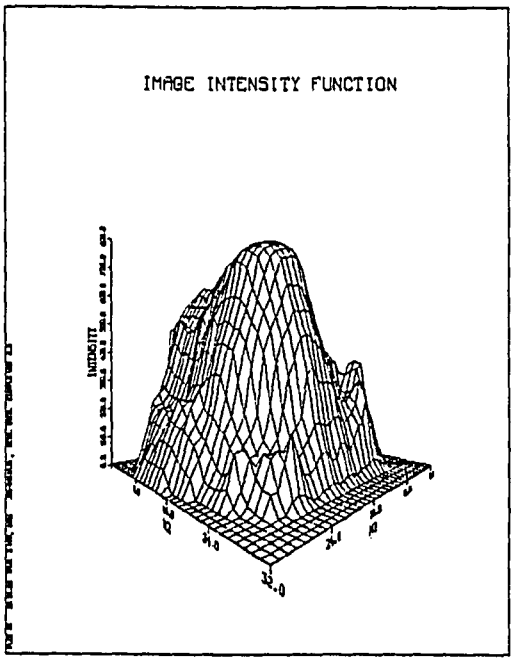
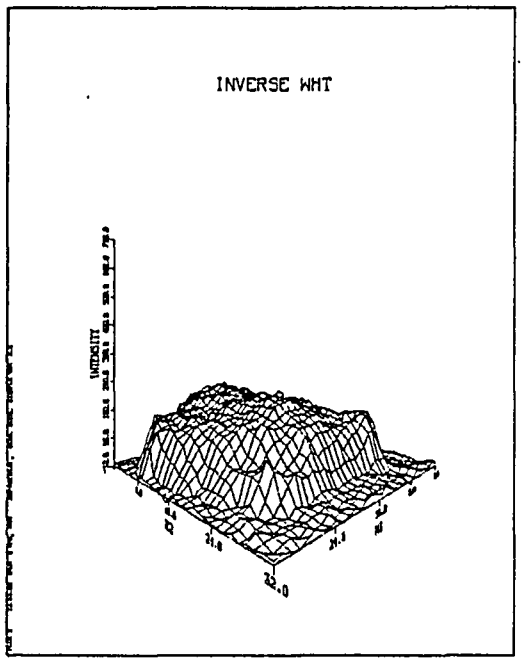
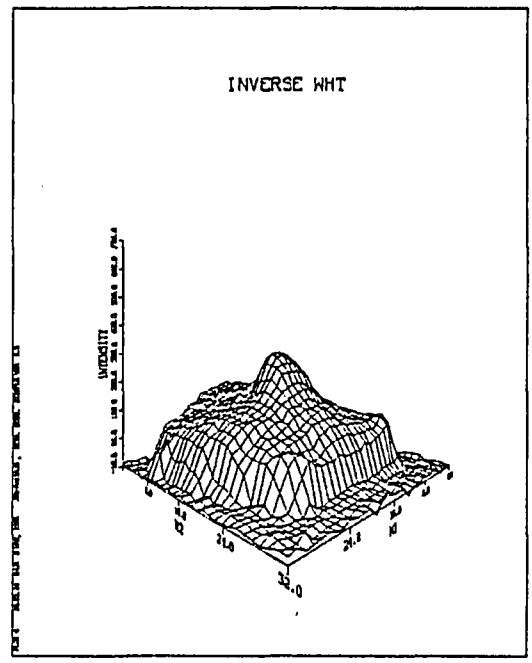


IMAGE INTENSITY FUNCTION

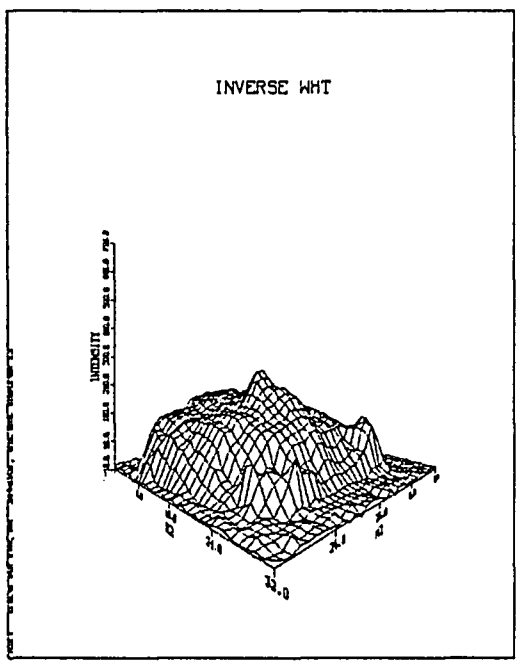
(o)



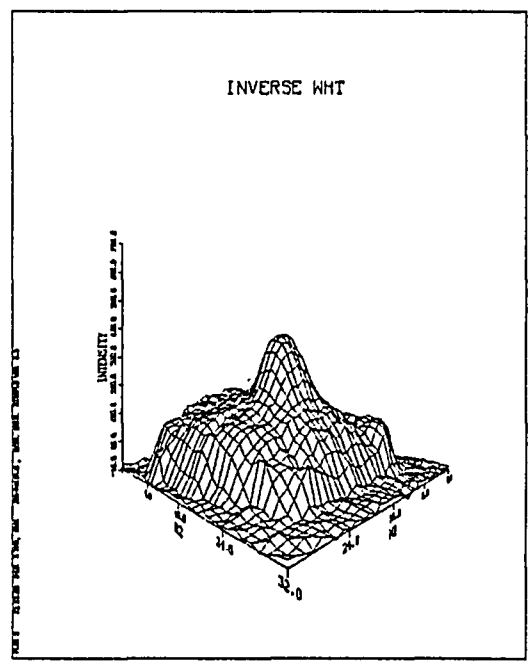
(b)



(d)

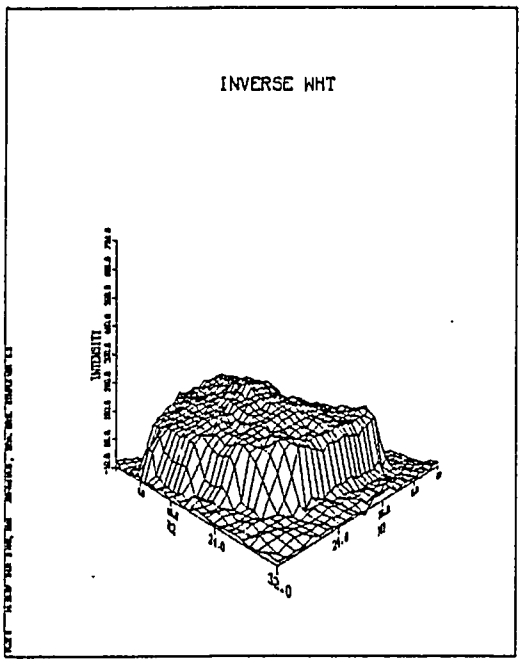


(a)

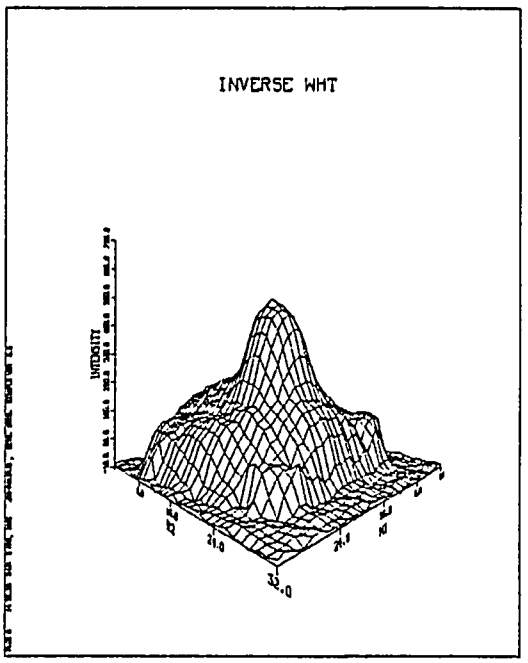


(c)

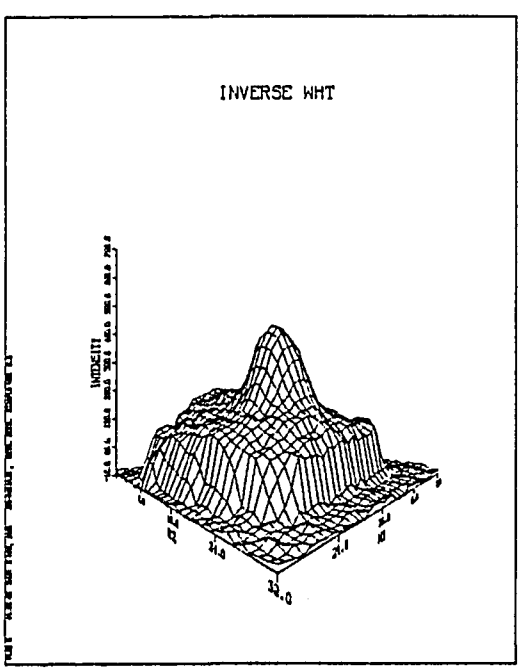
Figure 3.8 Walsh-Hadamard Transform Coded Infrared Images



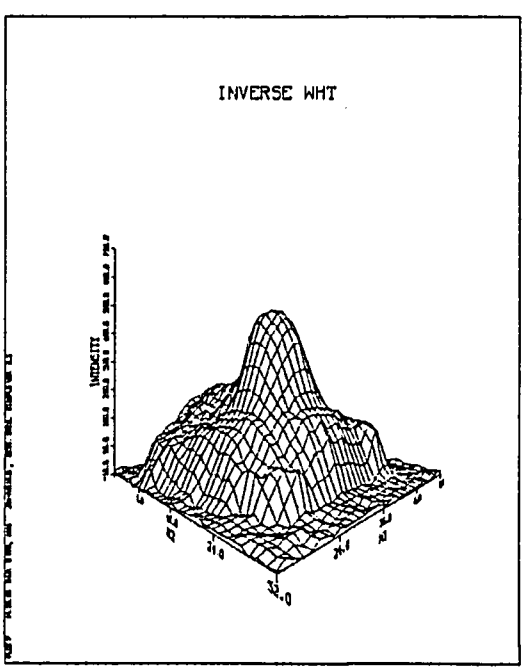
(f)



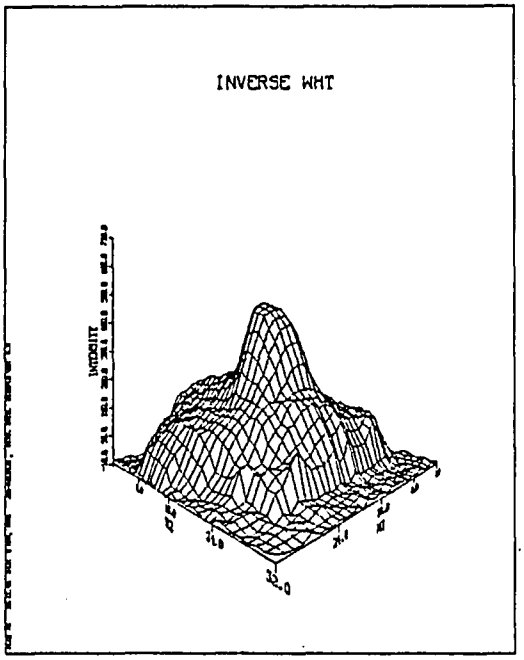
(h)



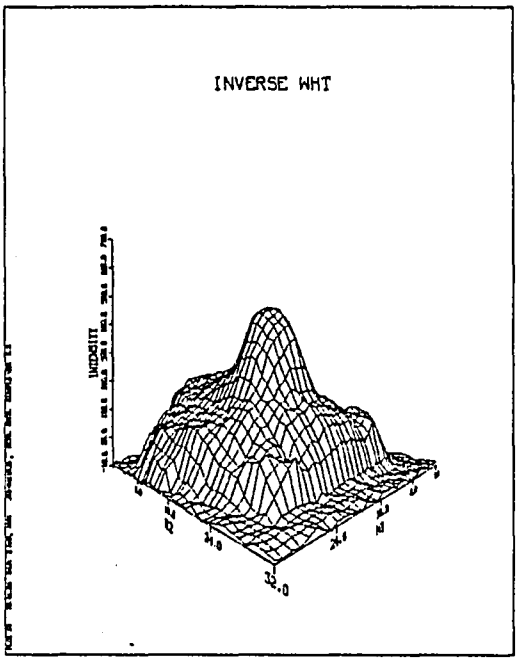
(e)



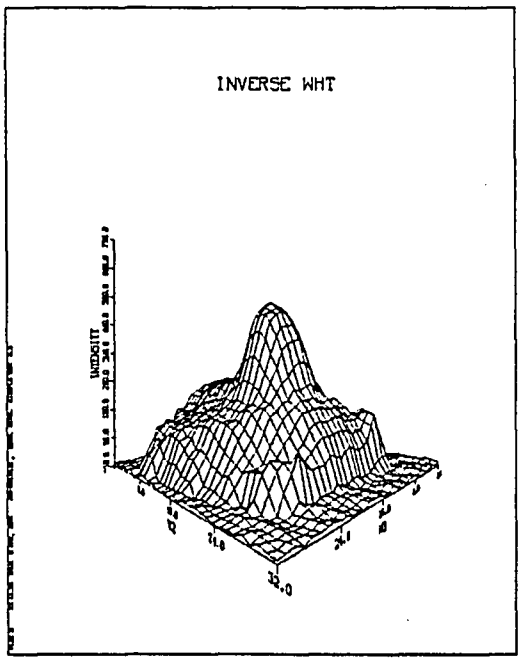
(g)



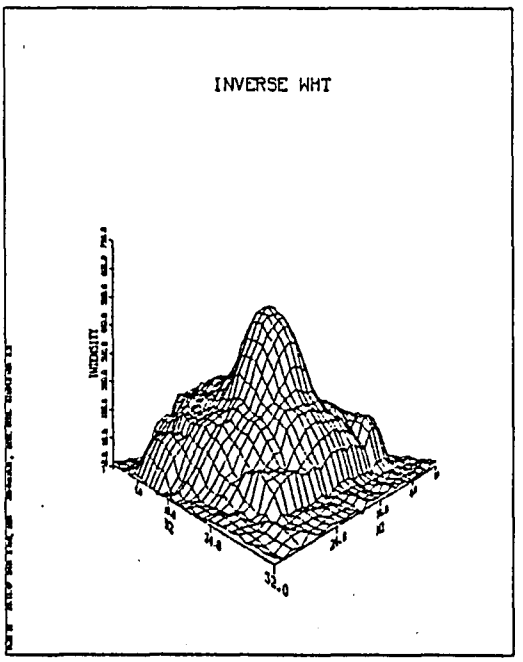
(j)



(l)

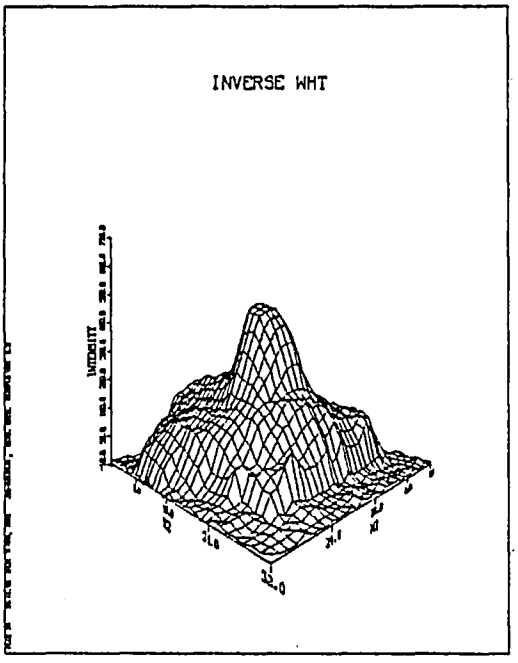


(i)

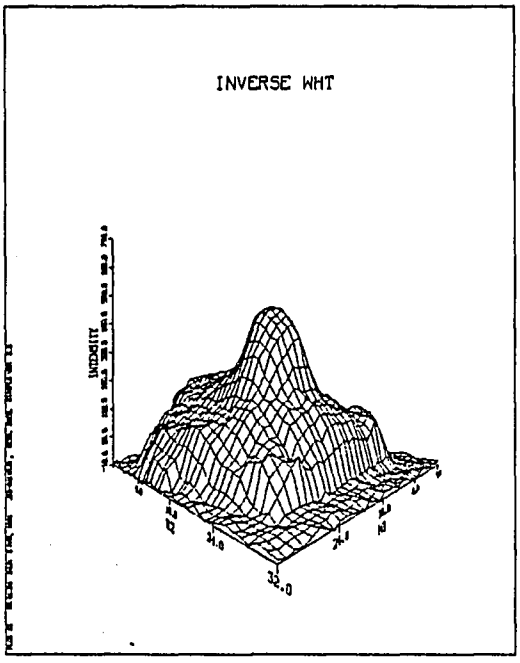


(k)

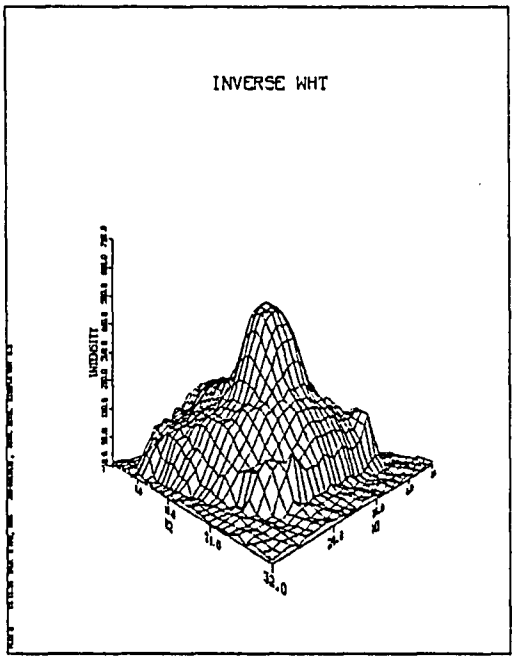
correction



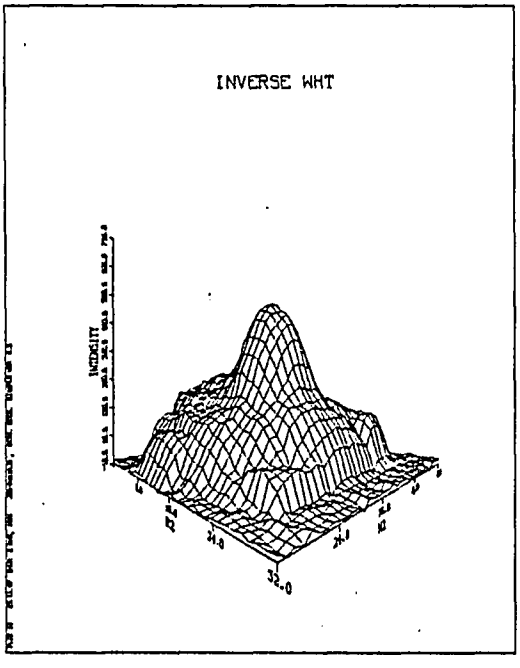
(j)



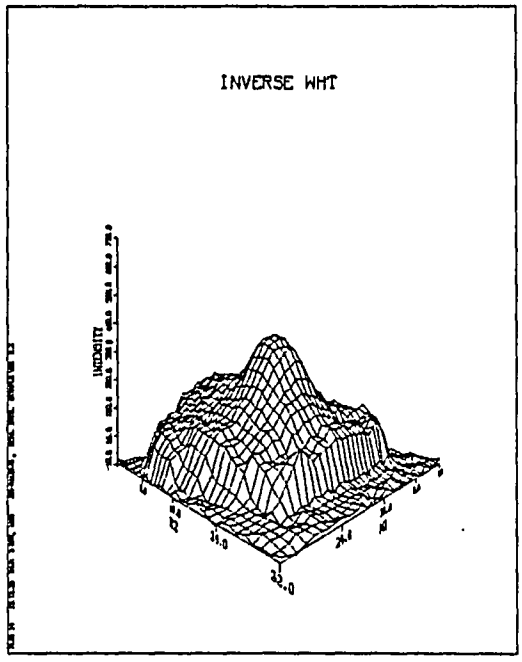
(l)



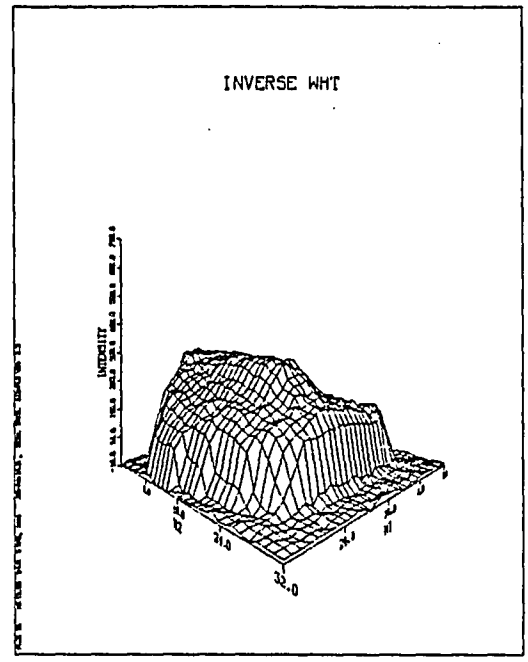
(i)



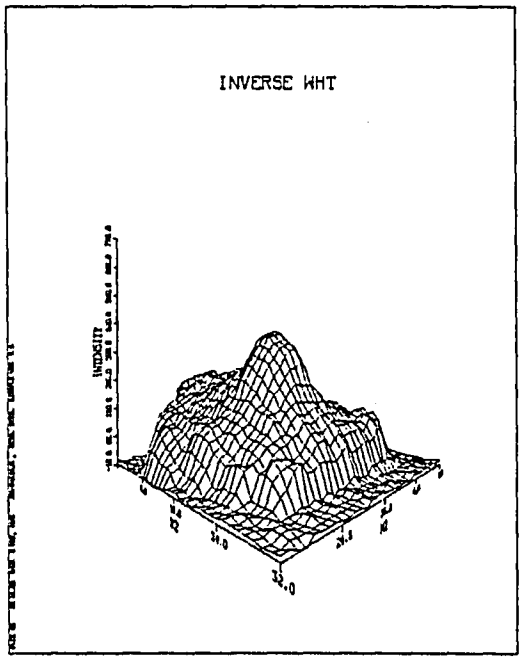
(k)



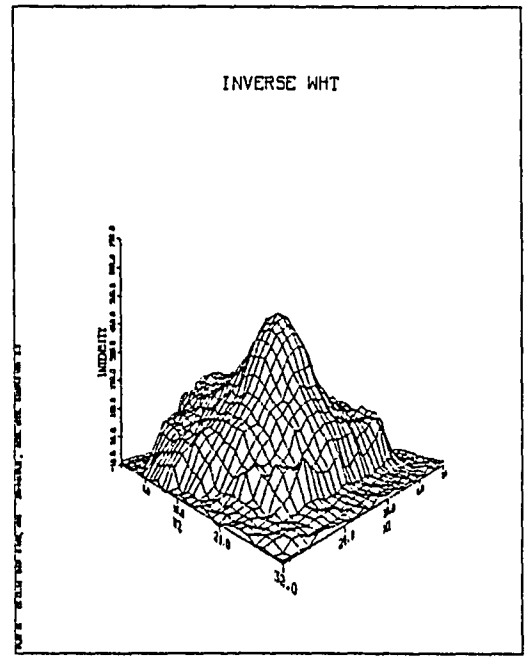
(n)



(p)



(m)



(o)

SUMMARY

This paper reviewed fundamental techniques for subjective improvement of image quality, discussed two-dimensional image transforms, and illustrated the design and implementation of two-dimensional FIR filters. It was shown that low contrast regions in images could be improved by a histogram equalization technique that expands the image dynamic range. Direct convolution filtering, with a very small mask as the filter impulse response, was demonstrated for noise removal (low pass case) and edge enhancement (high pass case). Two-dimensional Fourier and Walsh-Hadamard transforms were used to illustrate the potential for efficient image coding and direct transform domain filtering. Using the Walsh-Hadamard transform, it was shown that the required image storage space could be reduced by 70 percent. Two dimensional finite impulse response filters were designed using the windowing technique. Filters designed with a Hanning window were shown to be superior to filters designed with a rectangular window.

FORTRAN programs were written to perform each of the image processing functions described above. Thermal images from an infrared camera were processed off-line using these programs. The programs were effective in improving the subjective image quality and in removing the effects of random noise.

An image processing application was described in which an infrared camera is used to measure the temperature of materials in an iron-producing blast furnace. A scheme for adding four useful image processing functions to the existing system was described. It was shown that; (1) low pass digital filtering could be used to reduce the effect of random noise from the ironmaking process, (2) direct convolution high pass filtering could be used to locate severe temperature gradients, (3) histogram equalization could be used for contrast improvement, and (4) the Walsh-Hadamard transform could be used for efficient image coding. It was suggested that each of these functions could be implemented in software, and that they could be incorporated in cascade with the existing software at the appropriate location.

TRENDS AND FUTURE DIRECTIONS

Many systems, having some or all of the features described above, are already in existence to perform a variety of functions. Among the applications are aerial reconaissance from planes or satellites, television picture coding, computerized axial tomography, and nondestructive testing - just to name a few. In many of these applications, complex image processing functions are performed off-line. Recently, however, military applications such as target detection and industrial applications involving continuous high-speed processes are requiring that image processing functions be performed on-line, in real time.

This requirement for real time processing has led to the development of ultra-fast special purpose hardware. These processors are generally used as a "front-end" to a general purpose computer and are capable of performing fast Fourier transforms, floating point multiplications, and convolution filtering. This trend toward special purpose hardware has been accelerated by the growing popularity of distributed processing systems.

Probably the most significant factor contributing to the proliferation of image processing systems has been the development of high-quality, low-cost solid state scanners and cameras. These

solid state scanners are ideal for remote sensing applications requiring high reliability, and for industrial applications in hostile environments.

The development of better imaging devices, and the declining cost of hard-ware, should make many new applications both technically feasible and economically justified - especially in industry. The future of image processing appears to be very bright.

REFERENCES

(1) Cappillini, V., Constantinides, A., and Emiliani, P., Digital Filters and Their Applications. New York: Academic Press, 1978, p. 5.

(2) Castleman, K., Digital Image Processing, Englewood Cliffs: Prentice-Hall, 1979, p. 114.

(3) Hall, E. L., Computer Image Processing and Recognition. New York: Academic Press, 1978, p. 314.

(4) Pratt, W. K., Digital Image Processing. New York: John Wiley and Sons, 1978, p. 160.

(5) Castleman, K., Digital Image Processing. Englewood Cliffs: Prentice-Hall, 1979, p.70.

(6) Hall, E. L., Computer Image Processing and Recognition. New York: Academic Press, 1978, p. 202.

(7) Ibid p. 123

(8) Ibid p. 130

(9) Ahmed, N., Rao, K. R., Orthogonal Transforms for Digital Signal Processing. Berlin: Springer-Verlag, 1975, P. 30.

(10) Hall, E. L., Computer Image Processing and Recognition. New York: Academic Press, 1979, p. 143.

(11) Pratt, W. K. et al, "Hadamard Transform Image Coding," Proceedings IEEE, 57, 1, January 1969, pp. 58-68.

(12) Cappillini, V. et al, Digital Filters and Their Applications, New York: Academic Press, 1978, p. 20.

(13) Rabiner, L. R., and Gold, B., Theory and Application of Digital Signal Processing . Englewood Cliffs: Prentice-Hall 1975, p. 456.

(14) Cappillini, V. et al, Digital Filters and Their Applications. New York: Academic Press, 1978, p. 99.

REFERENCES

(15) Cappillini V. et al, Digital Filters and Their Applications. New York: Academic Press, 1978, p. 59.

(16) Haung, T. S., "Two Dimensional Windows," IEEE Transactions on Audio and Electroacoustics, AU-20, 1, March 1972, pp. 88-89.

(17) Castleman, K., Digital Image Processing. Englewood Cliffs: Prentice-Hall, 1979. p. 184.

(18) DISSPLA Software User's Manual. San Diego: Integrated Software Systems Corporation, 1970.

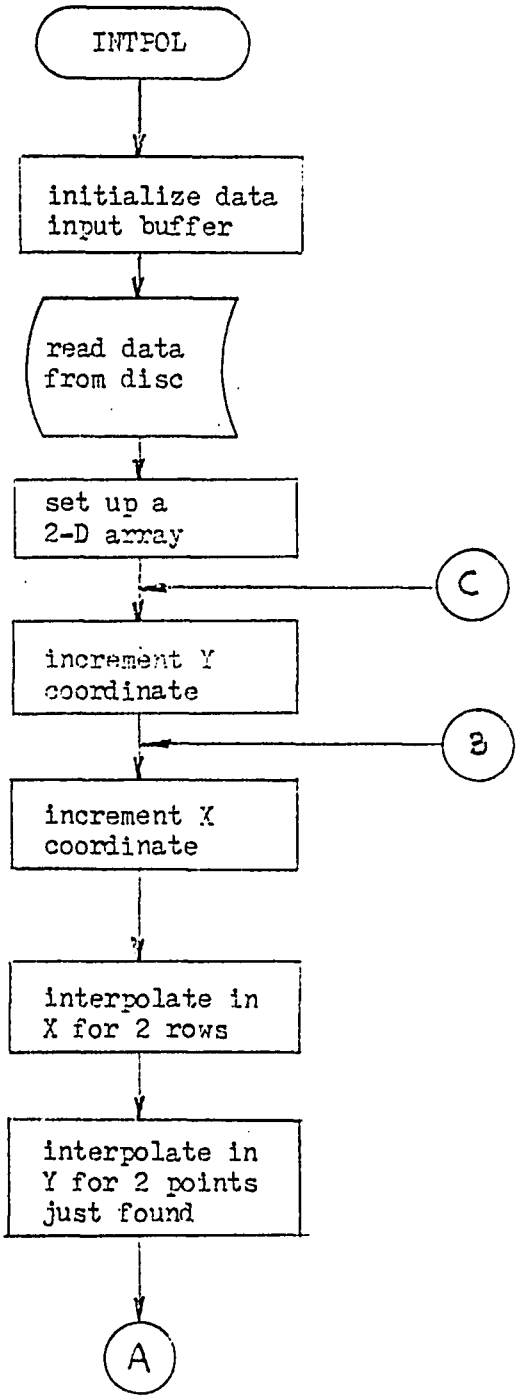
(19) IMSL Library Reference Manual . Houston: IMSL Corporation, 1980.

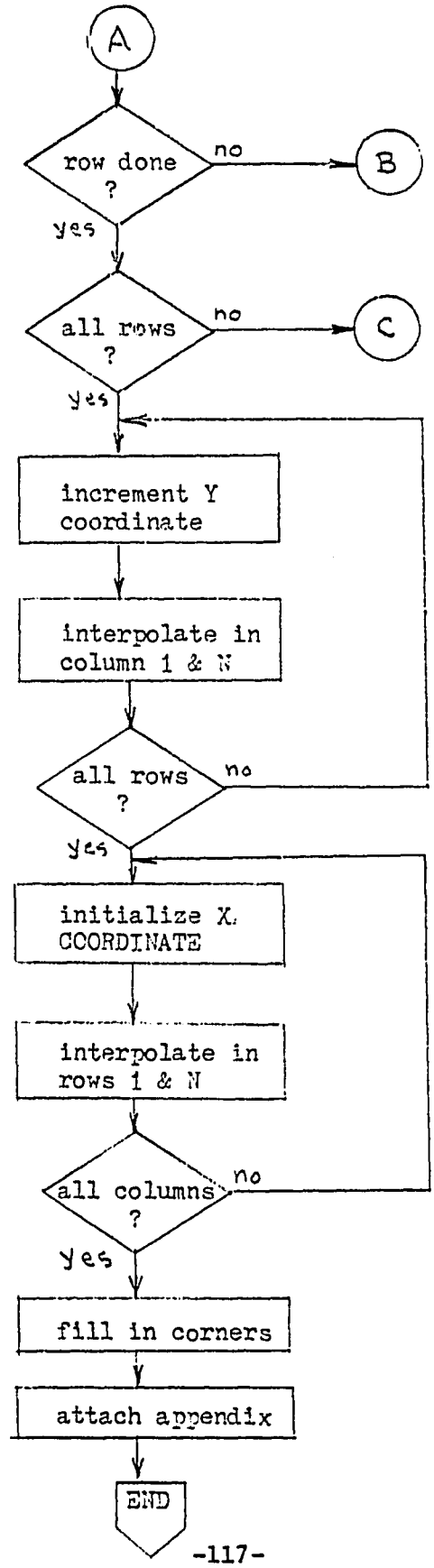
(20) Schenck, A. J. "Image Processing System for Blast Furnace Infrared Camera," Bethlehem Steel Corporation Research Report (unpublished), January 1981.

APPENDIX

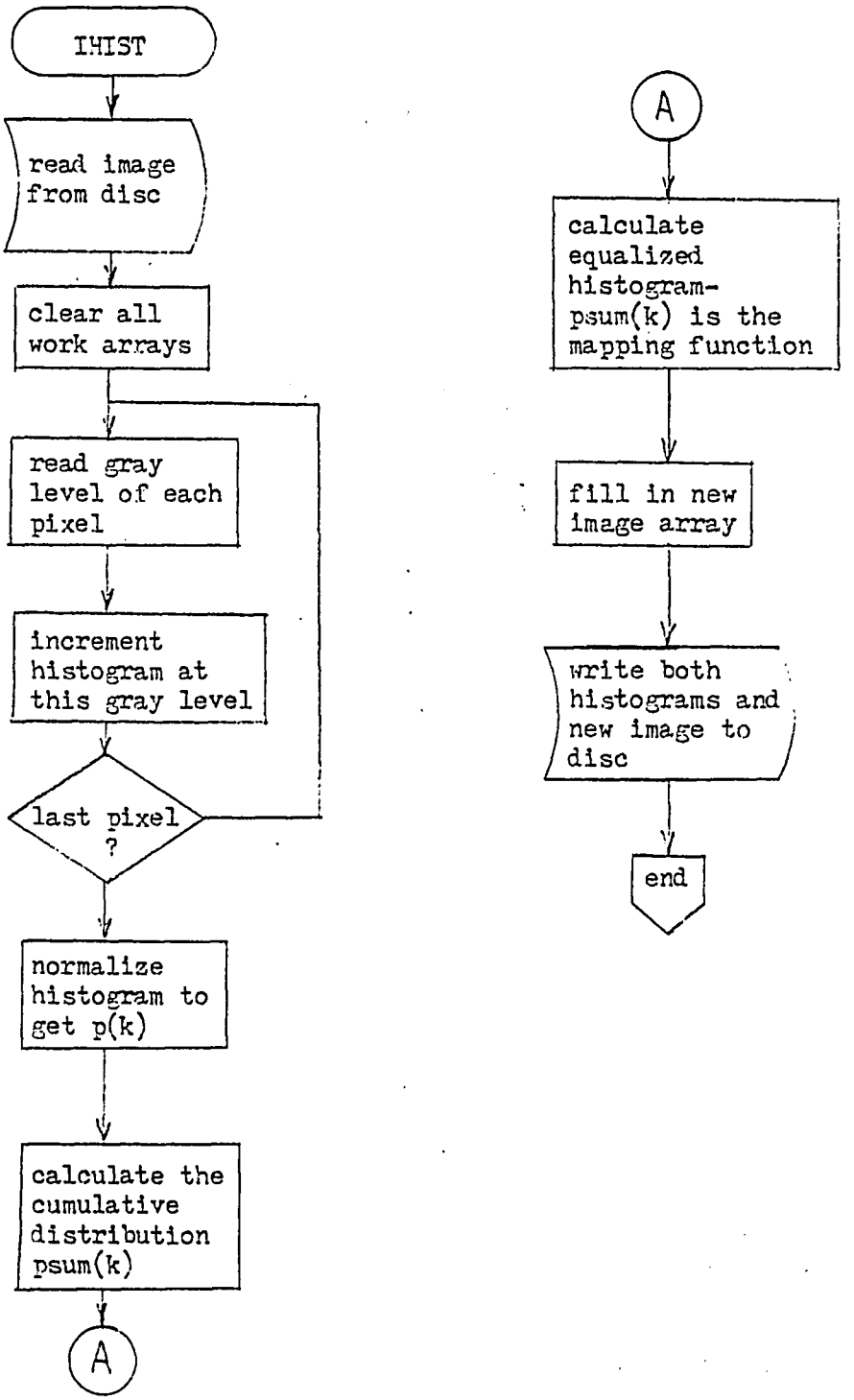
This appendix contains program flowcharts for the image processing software developed in Chapter 2.

Flowchart for Program INTPOL

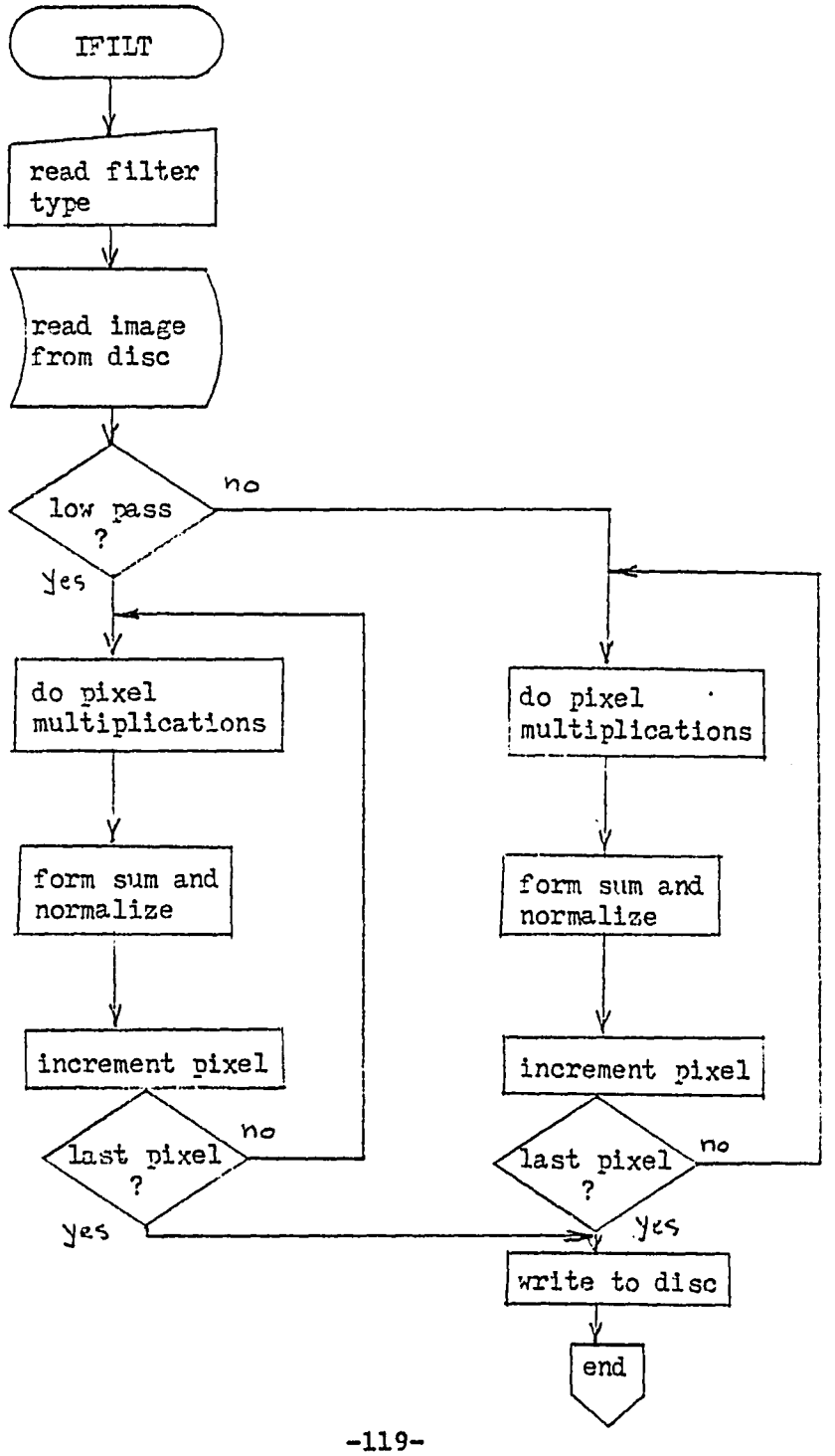



A
row done ?
no
B
yes
all rows ?
no
C
yes
increment Y coordinate
interpolate in column 1 & N
all rows ?
no
yes
initialize X. COORDINATE
interpolate in rows 1 & N
all columns ?
no
yes
fill in corners
attach appendix
END

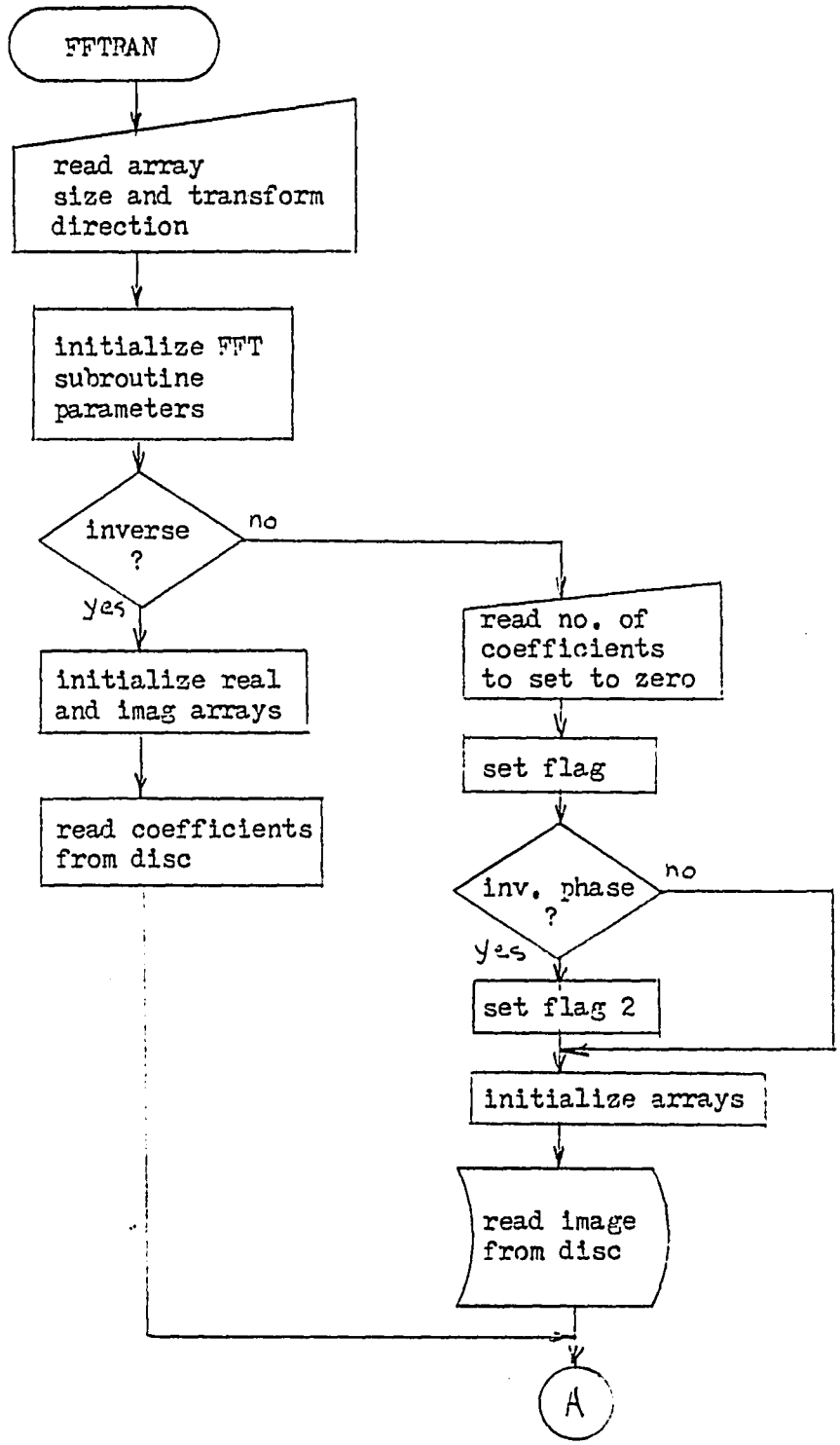
Flowchart for Program IHIST

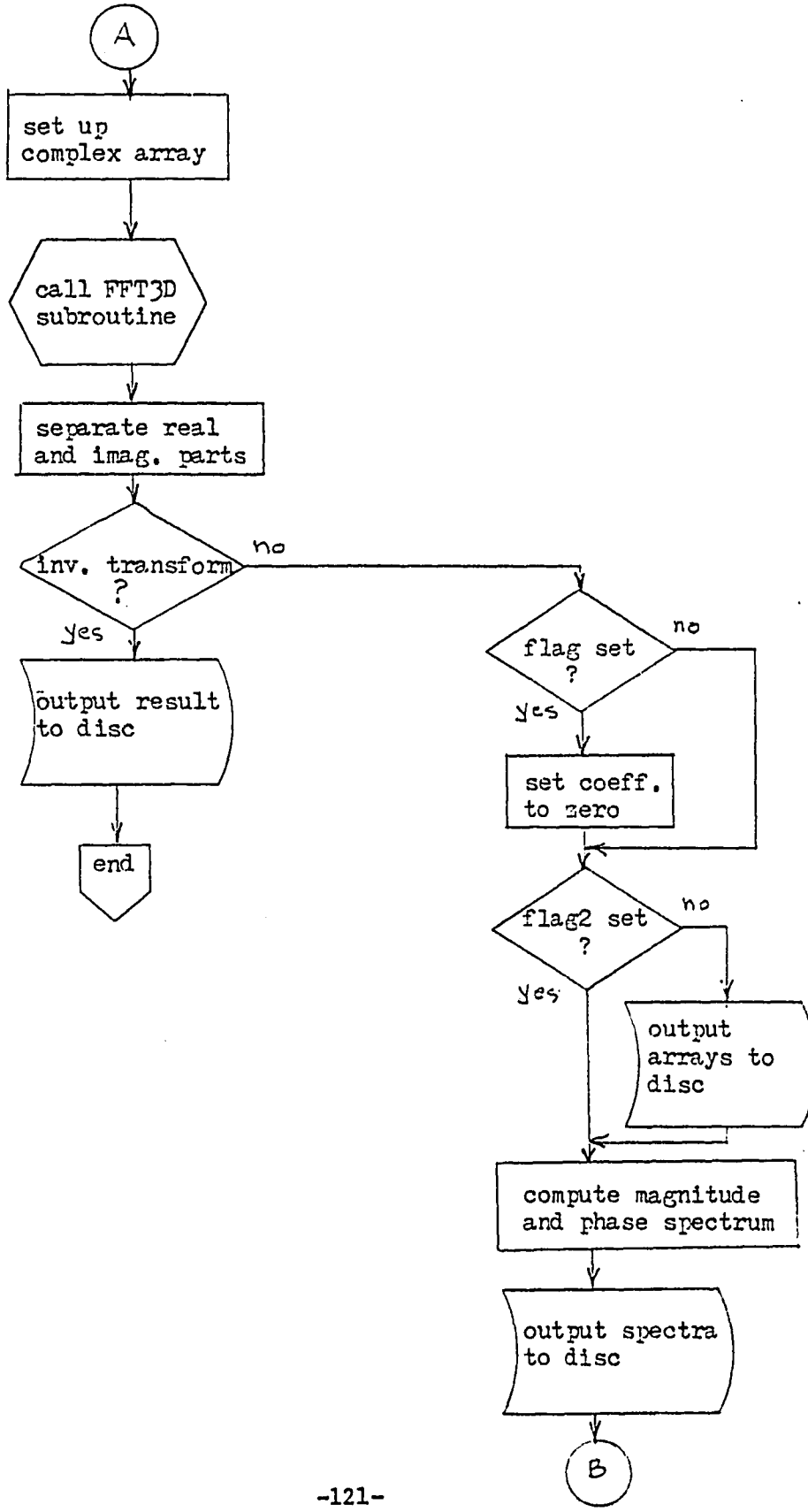


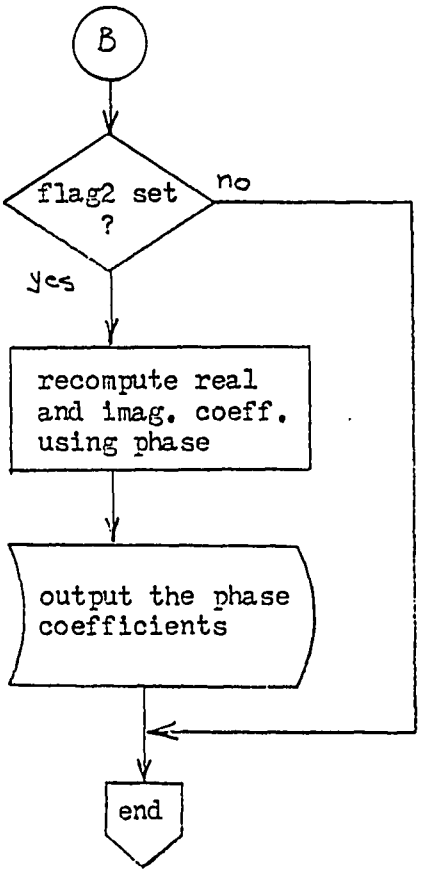
Flowchart for Program IFILT



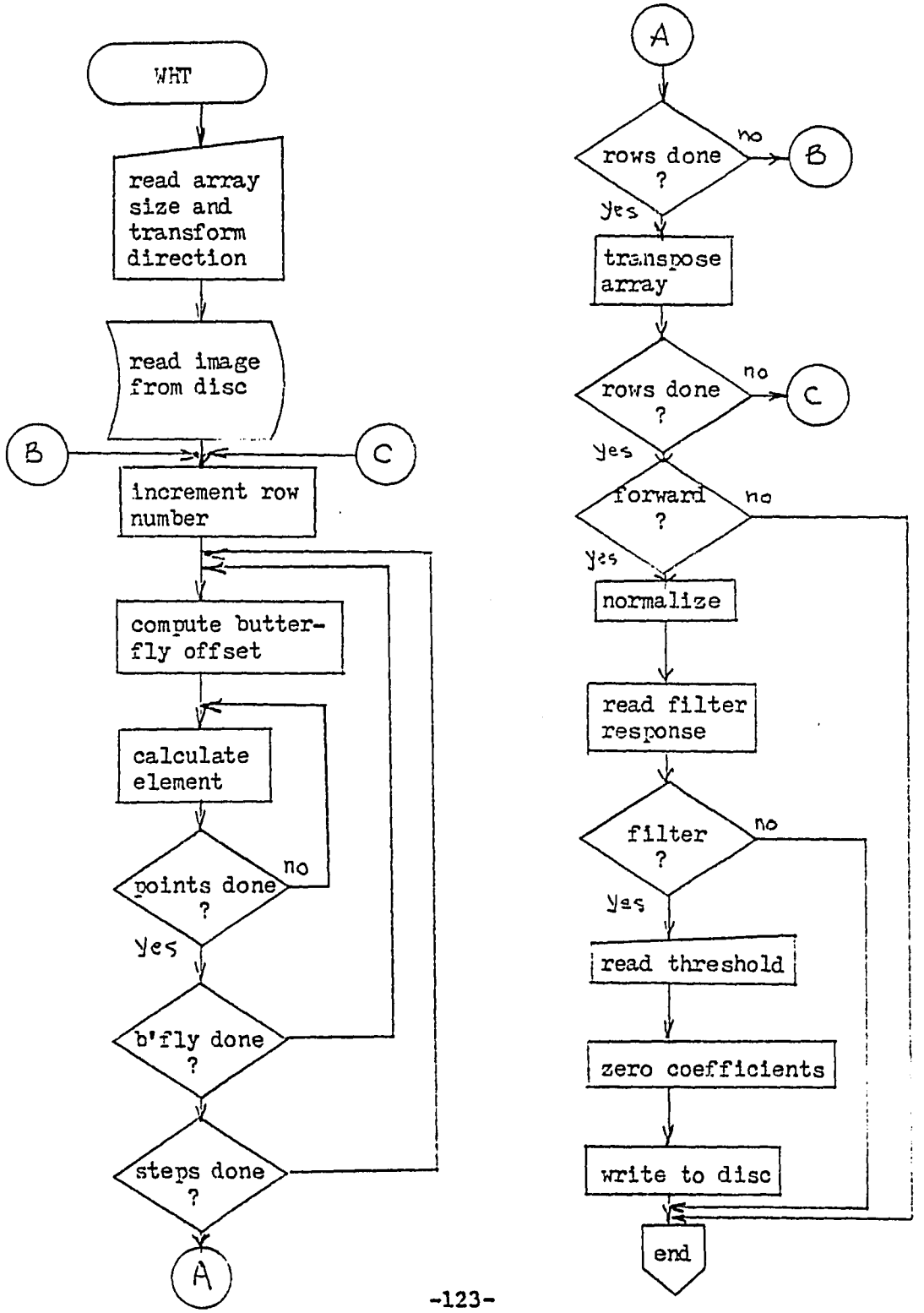
Flowchart for Program FFTRAN



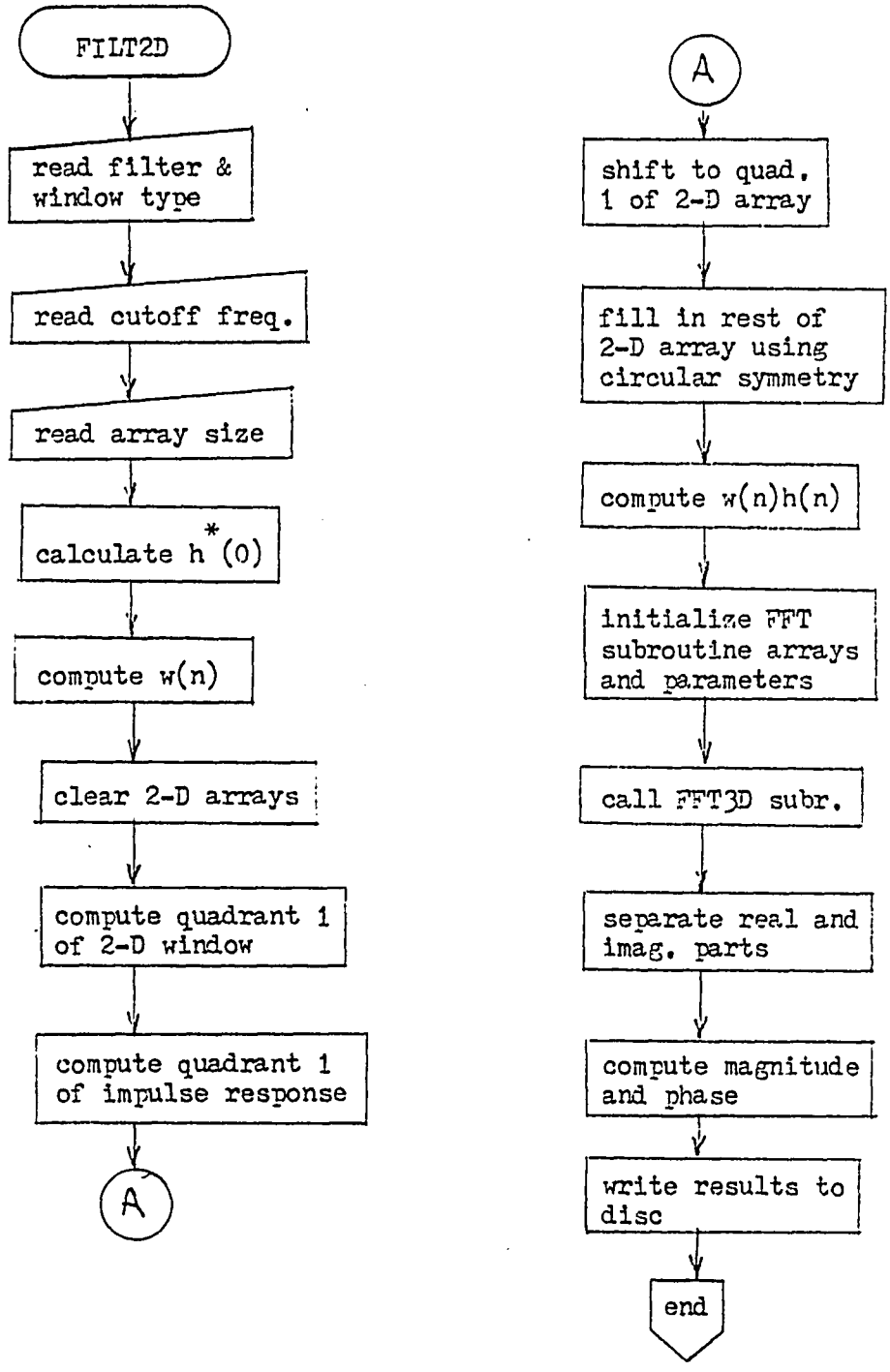

A
set up complex array
call FFT3D subroutine
separate real and imag. parts
inv. transform ?
no
yes
output result to disc
end
flag set ?
no
yes
set coeff. to zero
flag2 set ?
no
yes
output arrays to disc
compute magnitude and phase spectrum
output spectra to disc
B


B
flag2 set ?
no
yes
recompute real and imag. coeff. using phase
output the phase coefficients
end

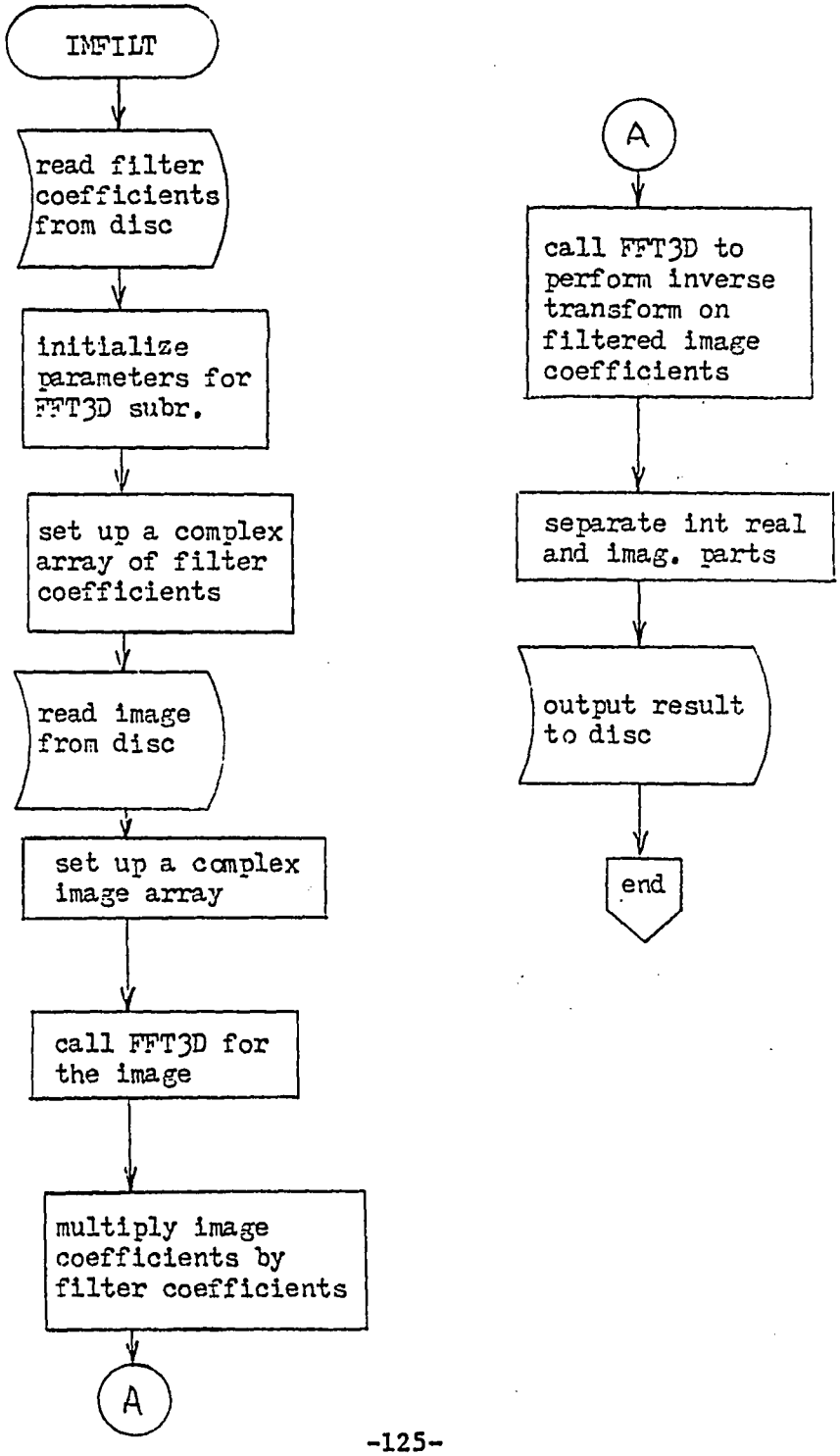
Flowchart for Program WHT



Flowchart for Program FILT2D



Flowchart for Program IMFILT



AUTHOR'S BIOGRAPHY

Arthur J. Schenck was born in Easton, Pennsylvania on July 7, 1951. He is married and currently lives in Phillipsburg, New Jersey. In 1973 he graduated magna cum laude from Lafayette College, Easton, Pennsylvania with a Bachelor of Science degree in Electrical Engineering. He is a member of Phi Beta Kappa, Eta Kappa Nu, and Tau Beta Pi, and is currently working toward a Masters degree in Electrical Engineering at Lehigh University, Bethlehem, Pennsylvania.

Since 1973 he has been with the Homer Research Laboratories of Bethlehem Steel Corporation where he has been involved with the development of nondestructive testing instrumentation and image processing systems. He has been awarded a patent for a device to detect repetitive defects in steel bars caused by cracks in rolling mill rolls. He has recently applied for a patent for a system to acquire and process images from an infrared camera which is used to monitor the temperature of materials in an operating blast furnace. He is currently involved in the development of a computerized tomography system to be used for on-line measurement of structural steel cross sectional dimensions.