ELECTRICAL ENGINEERING DEPARTMENT

UNIVERSITY OF TENNESSEE

KNOXVILLE
TN 37916

Final Report. Contract NAS8-29271

Project Title

The Application of Image Enhancement Techniques to Remote Manipulator Operation*

Principal Investigator: R.C. Gonzalez
Co-Investigator: J.D. Birdwell

Technical Report TR-EE/CS-74-15

September, 1974

* With the exception of the Summary and pages 12, 12(a)-12(d), and 40, this report is a duplication of a thesis entitled "Development of a Language-Based Digital Image Enhancement System," written by John D. Birdwell, and submitted to the Graduate School, University of Tennessee, as partial fulfillment for the degree of Master of Science with a major in Electrical Engineering.

SUMMARY

This report summarizes the principal image enhancement approaches investigated under contract NAS8-29271. These approaches can be divided into two categories: (1) image domain transformations, and (2) frequency domain transformations.

Image domain transformations operate directly on the image to be enhanced. Letting x represent the gray levels of the picture elements in the original image, these transformations are of the form y = T(x), where y represents the gray levels of the picture elements in the enhanced image. In other words, the transformation T directly alters the gray levels of the original image to produce an enhanced image. The advantages of this type of transformation is good control over enhancement functions, simplicity of implementation, and computational speed.

Frequency domain transformation are based on the Fourier transform. The enhancement process consists of (1) taking the Fourier transform of the original image (2) altering the transform by means of filtering techniques, and (3) taking the inverse Fourier transform to produce the enhanced image. This approach is ideally suited for the elimination of noise and for sharpening operations. It can also be used to perform some of the enhancement functions of image domain transformations such as contrast enhancement, but with con-

siderably more difficulty. Frequency domain transformations also offer the advantage of real-time implementation by means of laser techniques.

The two image enhancement approaches are discussed in Chapters 2 and 3. Chapter 4 presents some results in the development of an image enhancement language which requires little knowledge of enhancement techniques on the part of the operator. This language serves as the interface between the operator and the image enhancement system. Chapter 5 presents some experimental results using both enhancement approaches. The computer programs used in obtaining these results are summarized in the Appendix.

In addition to the work reported here, considerable additional hardware and software development was carried out under this project. As has been pointed out in the progress reports issued during the course of the contract, a complete image enhancement system has been assembled for this project under the sponsorship of the University of Tennessee. The documentation of this system as well as further results on image enhancement techniques will be provided during Phase II of the investigation which will cover the period September 1, 1974 through August 31, 1975.

ABSTRACT

The purpose of this project was to design and implement methods of image enhancement which can be used by an operator who is not experienced with the mechanisms of enhancement to obtain satisfactory results.

Two major areas of image enhancement were investigated. The first techniques to be investigated were the transformations which operate directly on the image domain. This investigation resulted in the new technique of contrast enhancement described in Chapter II. The second area of investigation concerned transformations on the Fourier Transform of the original image and included such techniques as homomorphic filtering, as described in Chapter III.

Finally, the methods of communication between the enhancement system and the computer operator were analyzed, and a language was developed for use in image enhancement. A working enhancement system was created and the program listing appears in the Appendix.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

This work deals with adaptive image enhancement by means of digital computer techniques. These techniques prove useful in situations where it is desired to suppress or improve image degradation effects such as noise, extremely light or dark areas, shadows, poor focus, and insufficient lighting. These degradations can often obscure details that are present in an image.

There are many systems which can enhance an image; however, these systems require a substantial amount of trial and error on the part of an operator to find an adequate enhancement technique. Furthermore, the best approach for enhancing one image may be undesirable for another. Therefore, this thesis is concerned with the development of an image enhancement system which can determine from the image much of the information that previously had to be provided by an operator. The element of trial and error still subsists, but it is hoped that the time required in experimentation can be substantially reduced.

Since all processing of a picture is done by computer, it is desirable at this time to review a few of the basics of digital signals and digital image processing. A digital image is an array of elements, called pixels (short for "picture elements"). These pixels can be either real valued or complex. The array can be thought of in the computer as a two-dimensional subscripted array of numbers. The range of pixel values is arbitrary, but in many instances is thought of as the unit interval (0,1), where zero represents black and one represents white. Since the pixels are binary numbers, they are discrete-valued. In the work reported in this thesis, each pixel can assume 256 possible values (gray levels) in a real image, and 65536 values in a complex image. These numbers are used to conform to an existing system on which the results are displayed. The range of pixel values used is either (-.5,.5) or (0,1), depending on the specific task.

In this work image processing techniques are divided into two broad categories. An operation on an image is placed in the first category if it can be performed directly on the image space. Similarily, an operation is placed in the second category if techniques involving the Fourier Transform space are required. For many images, an operation from one category is better than one from the other category. For instance, suppose that the edges of an image are fuzzy and that sharp edges are required. Then, it is very difficult to enhance the image using techniques from the first category; however, when the Fourier Transform is taken of the image, the high spatial frequencies correspond to the sharp transitions in the image, and therefore by multiplying the high frequency components by a gain greater than unity, an increase in the apparent sharpness of the inverse transform is realized. As another example, suppose that the pixels in an image are close together in value. The resulting image lacks contrast, and objects are not always easily distinguished from one another. Here, with even 256 gray levels, enough information is in the image to restore the contrast of the picture, since the eye can generally distinguish at most in the neighborhood of two dozen gray levels. This type of enhancement can often be performed with a technique known as homomorphic filtering using the logarithm of the image; however, it is usually quite difficult to find a filter which is satisfactory. Original to this thesis is a technique whereby this type of enhancement can be done easily on the original image. The technique is basically a non-linear transformation on the pixels, and the parameters needed for the operation are easily computed. Variations on this transformation and its applications are thoroughly discussed in the following chapter.

CHAPTER II

IMAGE DOMAIN TRANSFORMATIONS

In order to develop a useful transformation on an image, certain criteria must be introduced. Suppose that there are two distinct gray levels in the original image. Then if gray level A is whiter than grey level B, it is logical to require that if T is the image transformation, then T(A) be whiter than T(B). In mathematical terms, this constraint requires that T be a monotonic, non-decreasing function. It would be best if T were increasing, since there would then be no information loss in T due to the mapping of two or more gray levels onto one level; however, this is not practical, since T is a transformation between two discrete sets.

Now, something should be mentioned concerning the boundary requirements on T. Suppose that T maps the unit interval from zero to one onto the unit interval from zero to one. This mapping and monotonicity require that $T(0)=0$ and that $T(1)=1$. In other words, in order to form an image with the same range of gray levels, these conditions must hold.

At this point, heuristic arguments must be applied which will determine the final class of transformations. A common approach used requires that the distribution of the pixels in the resulting image be uniform. On examination, this constraint implies that the integral from zero to x of the distribution function be linear. This integral, however, is the cumulative distribution function, and is the required transformation T on the original image. Thus, the transformation T(x) on any pixel with gray level x is given by

$$T(x) = \int p(y)dy, \text{ over } 0 \text{ to } x, \qquad (2.1)$$

where p(y) is the probability density function of the pixels in the original image and x ranges from zero to one. This transformation has become very popular, and is usually referred to as histogram equalization (1). This transformation can work fairly well in many situations where the pixels in the image are clumped in groups; however, there is no control over what histogram equalization does to an image, and the relationship between pixels in an image can sometimes deteriorate.

A new approach developed in this thesis results from the following much weaker constraints. Suppose that it is required only that the very low valued and the very high valued pixels remain approximately stationary through the transformation; however, the middle values are allowed to change radically. In other words, if x is close to zero or if x is close to one, then T(x) is approximately equal to x. The argument behind this constraint is as follows: Suppose that a picture contains a scene which has very dark shadow areas and very light highlights. After enhancement, the picture should retain the blacks and the whites. If a picture does not retain these intensities through the enhancement process, the result will be a mixture of muddy grays where the minimum and maximum intensities were. The image will have a washed-out appearance due to an apparent lack of contrast.

Many enhancement problems involve an image that either has too high or too low a dynamic range. Thus, the image does not have the proper contrast between the light and dark areas required for good viewability. The information, however, is in many cases there, since there are many more gray levels in most images than can be distinguished by the eye. In a high-contrast scene, this information is often contained in the dark and in the light portions of the spectrum. Thus a good candidate for a transformation would be a function which spreads these dark and light areas of the spectrum out toward the center, while meeting the previously mentioned criteria. In a low-contrast scene, the information needed from the picture is often contained in the middle regions of the spectrum. Here, a good transformation to use would be one which spreads the center band of the spectrum out toward the edges. Also, it is obvious that a useful transformation should work in both cases of low and of high contrast.

In choosing the transformation function, it is noted that if a function is used which maps the finite picture interval onto the (infinite) real line, and that if this function has an inverse and is non-decreasing, then multiplication of the points generated by the function from the pixel interval by a positive constant and then applying the inverse function results in the desired spreading or contracting effect on the center band of the pixel spectrum, depending on whether the constant multiplier is greater than or less than one. At this point, a function that fits this description must be chosen. The function chosen here is a mapping of tan(x) applied to a linear function f(x) which maps (0,1) onto the domain of tan(x), $(-\pi/2,\pi/2)$, where $f(0)=-\pi/2$. The resulting transformation is

$$T(x)=\mathrm{arcf}(\arctan(a*\tan(f(x)))),\ x \neq 0 \text{ or } 1, \qquad (2.2)$$

and

$$T(0)=0,\ T(1)=1,$$

where arcf is the inverse of f, and the symbol * indicates multiplication. This function satisfies all the constraints previously set forth.

In order for this procedure to be effectively used in image enhancement, there must be a satisfactory way to estimate a usable value for the multiplication constant. This multiplication constant is directly related to the change in contrast between the original and the enhanced image. When the derivative of T with respect to a is taken, the result is

$$T_a(x,a)=\tan(x)/(1+a*a*\tan^2(x)), \qquad (2.3)$$

where the linear function f is for simplicity taken to be unity. Note that near the edges of the pixel spectrum, tan(x) becomes very large, dominating $T_a(x,a)$, and T_a becomes small. Thus, the choice of a has little effect on the ends of the spectrum. In the center area of the spectrum, also, tan(x) becomes small and T_a is small. Thus, there is little change in the values of pixels located in the center of the pixel spectrum with variations of a. Between the center and the edges of the spectrum, however, the parameter a controls the shape of the transformation curve. Also, and very important, is the fact that a*a, and not just a, appears in T_a, as this will relate to the variance, which is a measure in units of distance squared.

Now examine the derivative of T with respect to x, given by

$$T_x(x,a)=a(1+\tan^2(x))/(1+a*a*\tan^2(x)). \qquad (2.4)$$

Note that when a is equal to one, that $T_x=1$ for all x. As a increases beyond one, T_x becomes greater than one near the center of the pixel spectrum, and less than one at some point toward the edges of the spectrum. When a is less than one the opposite result occurs. Solving for x where $T_x(a)=1$, yields $x=\arctan(1/\sqrt{a})$ in the domain of tan(x), a function which can be used to relate the spreading effect of the tangent transformation to a.

At this point, it becomes necessary to develop a measure of the contrast of the picture that can be found by a computer. First, the contrast of the picture must be adequately defined, and this definition must be consistent with the viewer's idea of contrast in the image. Obviously, the more contrast that an image has, the larger the range of the values of a majority of pixels in the image must be. Thus, the variation of values of pixels must be large in a scene with a high contrast. A measure of this variation which is often used is the variance of the pixel values. This measure can be used to approximate the contrast of a scene, and relates well to a*a, since this factor controls the spreading effect of the tangent transformation, and thus controls the variance of the new image directly. Of course, the variance does not always give a true indication of the contrast in the desired sections of the scene, but what is needed is a starting point and a way for the system operator to vary the enhancement parameters according to his requirements. The selection of a will be discussed below.

Thus far, it has been assumed that the mean of the image to be enhanced is in a region near the center of the pixel spectrum. This assumption, however, is rarely valid. The above transformation is inadaquate, since, if the original image has a mean which is not centered in the spectrum, and if the scene is of low contrast, then the application of the previous operation will result in an image in which the main body of information is shifted even further toward one edge of the pixel spectrum. This process results not only in a degraded image, but also in an irretrievable loss of information due to quantization error. Thus, the transformation needs to be modified so that it can also be used to enhance pictures whose means are not centered in the spectrum.

The necessary modification can be performed using a class of functions known as Mobius transformations in complex analysis (2). These functions are continuously differentiable, and given two ordered sets of three points, a unique Mobius function exists which maps one of these sets onto the other set, and its inverse exists and is a Mobius transformation. On the real line, these functions have the same property and are real valued when the points in one set are all either increasing or decreasing. These Mobius functions are of the form

$$f(x)=(Ax+B)/(Cx+D). \quad (2.5)$$

In order to modify T, consider the two ordered sets (0,m,1) and (0,.5,1), where m is the mean value of the pixels in the original image. Then there exists a Mobius transformation M(x) such that M(0)=0, M(m)=.5, and M(1)=1. Solving for A, B, C, and D, this transformation is of the form

$$M(x)=Ax/(1+Bx), \quad (2.6)$$

where $A=(1-m)/m$,

and $B=(1-2m)/m$.

The inverse of this function is

$$I(y)=y/(A-By). \qquad (2.7)$$

Now, if x is a pixel on the interval (0,1) in the original image, and if m is the mean value of the pixel spectrum, then y=M(x) is a pixel in an image in which the mean of the original picture is mapped to .5. This new picture can be operated on by T, and when the process is completed, the inverse mapping I(y) may be used to create a result which has an average intensity approximating the mean of the original image. If this inverse mapping is not used, then in most cases the resulting image will have a mean in the neighborhood of .5. Although the mean of the intermediate picture is not necessarily the center of the pixel spectrum, the results of this process on an image can be quite dramatic.

An outgrowth of the technique described above is realized when I(y) is not the inverse of the original function M(x), but is instead a second Mobius transformation which maps the center of the pixel spectrum onto a point other than the original mean. Using a transformation such as this, the average brightness level of the image can be shifted. For example, if the original image is too dark, the brightness and contrast of the image can be increased, resulting in a nearly normal appearance in the final image. In the image enhancement system developed with this thesis, this new center point is chosen to be the center of the pixel spectrum. This choice has the advantage that there are an equal number of available gray levels on each side of the center value, thus minimizing the overall quantization error for an image whose gray levels have a Gaussian probability density function.

Let m and σ be the mean and variance of the original image. Applying the Mobius transformation of Eq. (2.6) to the points m, m-σ, and m+σ yields the three points

$$\begin{aligned} v &= M(m-\sigma) \\ w &= M(m+\sigma) \qquad (2.8) \\ 0.5 &= M(m) \end{aligned}$$

where m has been moved to the center of the pixel spectrum.

Once the Mobius transformation has been applied, the image is enhanced by means of the transformation given in Eq. (2.2). The procedure can be summarized in a step-by-step manner with the aid of Fig. 1.

Figure 1(a) shows the original pixel spectrum ranging from 0 to 1, along with the mean value m and variance σ. Figure 1(b) shows the result of applying Eq. (2.8). Note that the mean of the image has been shifted to the center of the pixel spectrum.

The first step in applying Eq. (2.2) is to map the (0,1) interval into the ($-\pi/2$, $\pi/2$) interval. This is accomplished by the function f(x) which is given by

$$f(x) = \pi(x-0.5) \qquad (2.9)$$

The result is shown in Fig. 1(c). Note that the center of

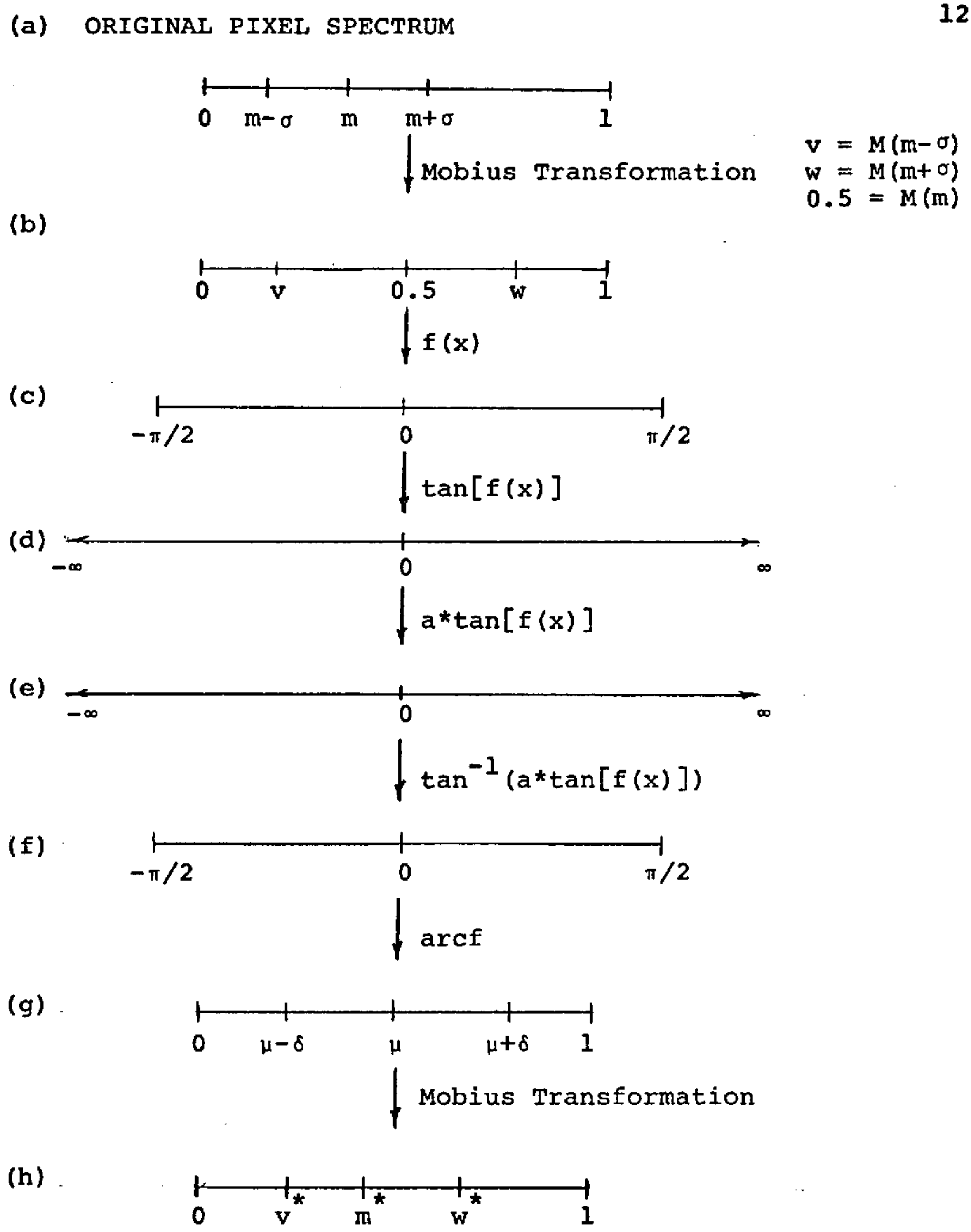

Fig. 1. Summary of Enhancement Procedure

the pixel spectrum is mapped to the origin in the $(-\pi/2, \pi/2)$ spectrum.

The next step consists of applying the tan function to the results of Fig. 1(c). This is shown in Fig. 1(d).

According to Eq. (2.2), the next step in the procedure calls for multiplication of tan $[f(x)]$ by the parameter a, as shown in Fig. 1(e). The transformation is then completed by taking the inverse tan function followed by the inverse of f, as shown in Figs. 1(f) and 1(g).

The parameter a controls the shape of the transformation curve and, therefore, it affects the spread or variance of the transformed pixels, as was previously indicated. The derivation of a relationship for a is best explained by comparing Figs. 1(b) and 1(g). In Fig. 1(b) the parameters v and w are related to original variance σ, although symmetry about the mean is not necessarily maintained. In Fig. 1(g), the parameter δ is the desired variance of the enhanced image. Suppose that δ is specified (it can be changed during an interactive enhancement procedure) and let

$$z = \begin{cases} v & \text{if } |v-0.5| > |w-0.5| \\ w & \text{otherwise} \end{cases} \tag{2.10}$$

In other words, z is the maximum of v or w about the mean in Fig. 1(b).

The parameter a must be such that T maps z to the point $\mu+\delta$, where μ is the mean of the enhanced image. Because of the nature of the transformation we have that $\mu \approx 0.5$. Then, using Eq. (2.2),

$$T(z) = \mathrm{arcf}(\arctan(a^{*}\tan(f(z)))$$

But, from Eq. (2.9),

$$f(z) = \pi(z-0.5)$$

Therefore,

$$T(z) = \mathrm{arcf}(\arctan(a^{*}\tan|\pi(z-0.5)|)$$

or

$$\tan(f[T(z)]) = a^{*}\tan|\pi(z-0.5)|$$

where the absolute value is included to prevent a from becoming negative, (See Eq. (2.11)). Since $T(z) = \mu+\delta$, we have that

$$\begin{aligned} f[T(z)] &= f(\mu+\delta) \\ &= \pi(\mu+\gamma-0.5) \\ &\approx \pi\delta \end{aligned}$$

where the last step follows from the fact that $\mu \approx 0.5$. The expression for a now becomes

$$a = \frac{\tan(\pi\delta)}{\tan|\pi(z-0.5)|} \qquad (2.11)$$

This completes the enhancement process. If desired, the mean of the enhanced image can be shifted further by using a second Mobius transformation, as shown in Fig. 1(h). This, for example, may be used to return the mean of the enhanced image to the its original value before enhancement.

The parameter δ is initially specified to be 0.35. It can be changed during interaction with the image enhancement system. The interactive enhancement procedure is discussed in Chapter IV.

CHAPTER III

FREQUENCY SPACE TRANSFORMATIONS

In this thesis, any image enhancement process which uses the Fourier Transform in the computation of a new image is categorized as a frequency space transformation. The majority of these techniques involve the computation and application of a filter to the Fourier Transform of the image. The goals of such filtering vary widely. For instance, if an image contains a periodic noise signal, then, by computation of the noise frequency spectrum and removal of this component of the Fourier Transform of the image by division, the noise in the original picture can be nearly eliminated. Thus, unwanted components of an image can be removed through frequency space transformations. With the use of the Fourier Transform, an image can also be sharpened. If the original image is unfocused, then the high frequency components of the Fourier Transform can be multiplied by a function which is greater than one in such a way as to increase the sharpness of the image. This effect is obtained because the high frequency components of the Fourier Transform correspond to sharp transitions in the image, or edges of objects.

A third use of frequency space transformations is a form of pattern, or image, recognition. If an image of an object is available, and a computerized method of recognizing this object in another image is needed, this technique can be used. First, a Fourier Transform of the object to be recognized is obtained. Then, this transform is divided into the transform of the unknown image point by point. When the inverse of the result is taken, areas where the pattern occurred in the unknown image will contain a bright dot, corresponding to an approximate delta function, and indicating recognition of that part of the image. One problem of this technique, however, is that the object in the pattern image must be aligned with the object in the unknown image for recognition to occur. Although there is as yet no good way to perform this alignment, one approach that could be used would be to pre-align the transforms of the unknown image and the pattern, using as a guide the axes where the maximum density in the transforms occur. Although this technique has not been used with any great success in computer processing of images, it has been applied in the field of optical pattern recognition (3).

Although it has been shown in the preceding paragraphs that there are many types of image enhancement using the Fourier Transform, only one group of enhancement procedures will be studied further here. This group is limited by allowing filtration on the transform with a filter which is only dependent on spatial frequency, or the norm of the complex frequency points in the transform plane. This type of filtering is often referred to as radial filtering. Of this type of filtering, two subsets will be considered; these are radial filtering on the transform of the original image, and homomorphic filtering using the transform of the logarithm of the original image.

Non-homomorphic radial filtering has some limited uses; however, as will be demonstrated, most of these uses can be approximated with homomorphic filtering. The primary use is in the enhancement of edges and of small objects. The filter curve used for this type of enhancement is, as stated before, a curve with regions in the high frequencies which have gains greater than unity. This increase in these frequencies causes edges and small objects to be enhanced in the resulting image. Very good approximations of the original scene can sometimes be produced in this way if knowledge of the blurring process which defocused the original image is available. If this process is dependent only on spatial frequency, then the inverse of the blurring process can be used as a radial filter to obtain a good resulting image. Good results using optics in this area have been obtained with an approximation to the dirac delta function being used as the blurring filter (3). These techniques work, however, only when the original information in the scene is not destroyed. If at certain areas of the Fourier Transform space, this information is missing, then obviously no filtering technique can re-create the original image.

In order to understand homomorphic filtering using the logarithm of the original image, a few basic concepts of image illumination and reflectance must be reviewed. Suppose that an object is viewed by an observer. Then what he sees is a combination of two components. These components are the illumination of the image, or the light being received by the image, and the reflectance of the image, or light being reflected into the viewer's eyes. Now the question arises as to the method of combination of these two components of indirect light. Suppose that $i(x,y)$ is the illumination on the point (x,y) of the image. Then if $r(x,y)$ is the ratio of the illumination to the amount of light received by the viewer from (x,y), then the amount received, or the pixel of the image at (x,y), denoted by $p(x,y)$, is the product of $i(x,y)$ and $r(x,y)$.

$$p(x,y) = i(x,y)r(x,y) \qquad (3.1)$$

Now suppose that filtering is to be performed in such a way as to operate separately on i and r. Since the relationship between these two components is a product, their relationship in the transform space is that of a convolution integral, which is definitely cumbersome. However, suppose that the logarithm of the original image, which is real, is used to produce an intermediate image composed of pixels $p'(x,y)$. Now, the relation between i and r is embedded in the sum of two logarithms. If

$$i'(x,y) = \ln(i(x,y)) \qquad (3.2)$$

and

$$r'(x,y) = \ln(r(x,y)) \qquad (3.3)$$

then

$$p'(x,y) = i'(x,y) + r'(x,y). \qquad (3.4)$$

Now, when the transform of p' is taken, the transform of the sum i' + r' is the sum of the transforms of i' and r'. Thus, a transform has been obtained in which the illumination and reflectance components can be much easier to work with; however, in order to filter the illumination and the reflectance components separately, these two components must occupy different regions in the transform space of the image. This requirement is fortunately the normal event, since the illumination of a scene is usually varying slowly, and large areas of the image are under approximately constant illumination, while the reflectance is usually varying rapidly, corresponding to the amount of detail in the scene.

Since the homomorphic filter operates on the logarithm of the original image, when the inverse transform is applied, an image corresponding to this logarithm is obtained. Thus, to obtain the final image, the exponentiation operation must be applied to the inverse transform. Thus, if the gain of the filter is constant and equal to one, this process will yield the original image (4).

Now that the techniques of normal and homomorphic radial filtering have been presented, a technique must be developed which can be used to generate a meaningful filter curve. In the case of non-homomorphic filtering, since the only goal is to attempt to sharpen the image, the only knowledge needed by the image enhancement system is the size of the largest object to be enhanced. This knowledge can be obtained either from the operator of the system, or an approximation can be calculated from certain properties of the image transform. This is done by examining the rate of decrease in amplitude of the components of the frequency spectrum. Since the size of an object in image space is inversely proportional to the frequency which represents this object in transform space, this rate of decay gives an indication of the minimum object size and of the object density in the image. This rate then gives an indication of the point in the frequency spectrum where a gain greater than unity should be used. Now the shape of the high frequency portion of the filter curve must be determined. From observation of filtering effects on images, this curve will degrade the image if changes occur too rapidly, and if a curve approximating an exponentially increasing function is used, good results are obtained. Here, again, a maximum allowable value for the filter at the highest frequency used must be chosen. This value depends on the amount of edge enhancement needed in the image, and to a large extent on an initial guess as to what will give good results on the part of the system designer. In the program developed with this thesis, this guess was varied until good results were obtained. When an operator specifies that objects of a certain size are to be enhanced, this program increases the filter curve by some experimentally pre-determined amount over a fairly wide band of frequencies, employing two exponential terms starting at the endpoints of this band of frequencies and multiplied together. The low frequency exponential is increasing, while the exponential starting at the high frequency endpoint is decreasing, resulting in a final increase at some point higher than the frequency band by some quantity which is almost constant. This quantity is also experimentally determined.

When homomorphic filtering is used, the techniques applied to the high frequencies are basically the same as those used in non-homomorphic filtering. These are the techniques used to determine the portion of the filtering curve which modifies the reflectance component of the image. The constant determined from the envelope of the frequency spectrum is used to determine the point which separates the illumination and reflectance components of the image. Although there is obviously overlap between these two components of an image, a point of division must be established in order to filter the two components separately.

The filtering curve for the illumination component of the image is fairly easily established. Since the illumination of an image controls the mean value of the image, the mean of the original image can be used as a variable in the determination of the proper filter curve, and since the amount of illumination also controls much of the contrast of the image, the variance can be used much as it was in the contrast equalization techniques discussed in Chapter II. For example, if the mean of the picture is excessively low, a gain greater than unity can be used to raise the illumination; however, this process can increase the contrast, or variance of the image to a point that can be objectionable. Thus, the variance of the original picture can be examined to determine the actual filter to be used. With these two variables in combination, a gain at the zero frequency point on the transform can be determined, and an exponential curve can be used to taper this change in gain toward a value near one at the break point between the illumination and the reflectance components of the image.

The procedure used to enhance an image using frequency space filtering can be summarized as follows:

1. The commands given to the program through the language processor, as described in Chapter IV, are examined to determine whether or not homomorphic filtering is to be used.

2. The mean and variance of the original image are computed.

3. If homomorphic filtering is to be used, the logarithm of the original image is taken. If homomorphic filtering is not requested, this step is skipped.

4. The forward transform of either the original image or of the result from step three is obtained.

5. In homomorphic filtering, the filtering curve is computed using the mean and variance of the original image and the decay constant of the envelope of the frequency spectrum. The mean is divided into the value representing the middle of the pixel spectrum and the result is raised to a real power, which is a constant available to the programmer, and is computed by trial-and-error. This constant may be required to have different values, depending on what is wanted by the operator. For instance, if a large portion of the picture area is dark, but the picture contains a very bright area which obscures the information in the dark areas, the operator will probably want a negative constant. This value will lower the illumination in the picture and thereby compress the dynamic range into a smaller region. The result of the above exponentiation is then

multiplied by a constant representing the desired variance of the resulting image modified by the position of the mean in the original picture. This value is then divided by the variance of the original picture. This process allows the variance of the original picture to influence the starting value of the filter curve; this influence is needed since the illumination of a scene directly controls the contrast of that scene. This value is then modified by the inverse of the computed decay constant to obtain the desired starting value of the filtering curve. The high frequency region of the filtering curve is determined by both the variance of the original picture and the command structure which is specified by the operator in homomorphic and non-homomorphic filtering. The high frequency region is a product of at most three significant exponential terms computed by the program. The relative value of the high frequency portion of the curve is determined by two constants which can be modified by the programmer.

6. The curve resulting from step five is used to filter the transform of the image and the Inverse Fourier Transform is applied to the result.

7. If homomorphic filtering was used, the exponential of the resulting image is computed to obtain the final result.

The result of these steps is the enhanced image.

CHAPTER IV

OPERATOR - SYSTEM COMMUNICATION

In any computerized processing system, there must be some form of communication between the operator of the system and the program being executed. This axiom is true whether the operator need only start and stop the program's execution, or a complicated decision process must be performed. In an image enhancement system, communication between the operator and the system allows the operator to modify parameters in the enhancement process and thereby control the resulting image. In order for this communication to occur, a language must be incorporated into the system so that an operator without knowledge of the internal program structure can obtain satisfactory enhancement results. In the image enhancement system produced in conjunction with this thesis, a language was developed and implemented. A description of this language is contained in the following pages.

The language implemented in the image enhancement system program in this thesis allows the operator to specify both standard options available in the system and various modifications to these options which can be useful in numerous enhancement tasks. The commands necessary for the processing of one image are defined to be one command input stream. The language is written in free form on 80 character records, such as cards or lines on a terminal, and consists of a series of commands to the language processor separated by commas and ending with an 'END' command, which must be the last command on the last record of the command stream for one image. Multiple command input streams can be written, and are processed sequentially. Each command has an inverse command, which is the command preceded by the characters 'NO'. If any error is detected in a command stream, that stream is deleted and processing is continued with the record after the next 'END' command.

Each command available to the operator is contained in one or both of two categories, depending on whether or not a command is used when the transform of an image is taken during processing. The operator specifies whether or not the transform is to be used in the enhancement process by using the command 'TRANS'. If 'TRANS' is specified, the transform is used. If 'NOTRANS' is specified, the transform is not used. If this command is omitted, the default option is 'NOTRANS'.

There are three commands which should always be specified, since these commands determine where the input and output files are located, and the file number of the output file. These commands are 'INFILE=', specifying the input file, 'OUTFILE=', specifying the output file, and 'FILENO=', specifying the output file number, where each of these commands is followed by a decimal number specifying the appropriate file or file number. The input file and output file are the equivalent of the dataset reference number in FORTRAN, and the output file number is the number which is written in the label record of the file plus one, and is specified so compatibility between this system and the display system currently available on a Digital Equipment Corporation PDP11/40 at the University of Tennessee is assured. The output file and input file specification should always be greater than nine, because the enhancement system reserves files one through nine for temporary and program control datasets. One other command which should be mentioned at this time is the 'PRINT' command, which when specified causes printing of the curves used to generate an enhanced image and of a histogram of this image. The default option is 'NOPRINT'.

When 'NOTRANS' is specified, there are two types of commands which can be used by the operator. One type has no operand, while the other type has an optional operand field. There are three commands in the first type. These commands are 'HIST', 'ENDARK', and 'ENLIGHT'. When one of these commands is omitted, the inverse command is the default value. The 'HIST' command forces the system to perform histogram equalization on the input file. If this command is specified, all other commands, with the exception of the file commands and 'END', are ignored. The 'ENDARK' command informs the system that the areas to be enhanced are the dark areas of the image; similarly, the 'ENLIGHT' command directs the system to enhance the light areas of the image. These two commands also give the operator the option of having the system form a value to be used in the tangent transformation described in Chapter II which is not necessarily the mean. When this approach is used, the system finds the point in a portion of the pixel spectrum where the largest number of pixels occurs and treats this point as the mean of the picture. The region of the pixel spectrum used depends on whether 'ENDARK' or 'ENLIGHT' is specified. If both commands are specified, the entire pixel spectrum is used in computing the maximum value.

The second type of command under 'NOTRANS' has an optional operand field which is a numeric value, and is optionally separated from the command by a semicolon. If the operand field is omitted, the value is assumed to be zero. There are two commands in this type; these are 'CONT' and 'ENBAND'. The 'CONT' command specifies that contrast enhancement is to be employed in creating the new image. The value in the operand field should be in the range from zero to 255. If the operand is out of this range, the operand value modulo 256 is used. If the operand is zero, the system determines the appropriate value to be used in contrast modification. If the operand is not zero, the value computed by the system is modified; when the operand value is less than 128, the computed value is modified so that the final image has less contrast than it would have if the computed value was used. If the operand value is greater than 128, the computed value is modified so that more contrast is obtained. When the operand value is 128, no change in the computed value occurs. The amount of change possible in the computed value is controlled by a constant which is specified during generation of the system. The 'ENBAND' command allows the operator to set the value which is treated as the mean of the picture by the system. The operand is in the same format as is the operand for the 'CONT' command. The operand value must be in the range of one to 255 for proper results. Values at extremes of this range should be used cautiously, since the value chosen is mapped onto the center of the pixel spectrum in the final image.

When 'TRANS' is specified, there are four commands available to the operator. Of these, three commands are in the first type discussed under the 'NOTRANS' command, and one is in the second type. The three commands in the first type are 'ENDARK', 'ENLIGHT', and 'SHARP'. Of these three commands, the function of 'ENDARK' and 'ENLIGHT' is similar to the function of these commands under the 'NOTRANS' command; however the techniques used for applying these commands to the image are radically different. The 'SHARP' command requests that an attempt be made to sharpen the image using high frequency enhancement techniques of radial filtering.

The one command which is in the second type of commands is 'ENOBJ'. The value of the operand must be in the range from one to 99, and represents a percentage. This command requests the system to enhance objects of sizes between one and 99 percent of the horizontal linear size of the displayed image and all objects of smaller size.

The commands specified under 'TRANS' determine whether or not homomorphic filtering is employed. If 'ENDARK' or 'ENLIGHT' is specified, or 'SHARP' is not specified, homomorphic filtering is employed. In all other cases, radial filtering on the transform of the original image is used. Thus, to filter on the transform of the original, without using the intermediate homomorphic transformations, 'SHARP' must be specified, and 'ENOBJ' is the only other command of the four available commands which can be used. Several complete enhancement examples are described in the next chapter.

CHAPTER V

EXPERIMENTAL RESULTS

In this chapter, the results of both the image space and the frequency space procedures are discussed. Pictures and details of the language commands used are also presented. All of the pictures on the following pages which are connected with image space transformations are compressed into sixteen gray levels when viewed. The compression is a mapping which simply assigns contiguous ranges of sixteen levels of the 256-level pixel spectrum to unique gray levels in the displayed image. The complex images used with the frequency space transformation are compressed from 65536 gray levels to 256 gray levels and are displayed in the resulting form. All of the pictures presented here contain 128 points in each line both horizontally and vertically, resulting in 16384 pixels in each picture. The observation should be made at this time that the image space transformation is considerably faster and easier to perform than is the frequency space transformation. The difference in speed between the image space transformation and the homomorphic filtering process is approximately a factor of four in favor of the image space process.

The first pictures to be presented are results of image space enhancement techniques.

Result # 1

The object of the enhancement of the picture in Figure 1 was to resolve detail hidden in the shadow on the face. The image space transformation was used. The results of the enhancement process appear in Figure 2. The mean value of the pixels in the original picture is 106.4. The mean value of the pixels in the enhanced picture is 104.7. The variance of the pixels in the original picture is 555.1; the variance in the enhanced picture is 4800.4. The commands which were used to enhance the picture in Figure 1 are as follows:

```
NOTRANS,INFILE=10,OUTFILE=21,FILENO=2,CONT,END     (5.1)
```

Note that the contrast of the scene is increased, and as a result, more detail is visible in the dark side of the face.

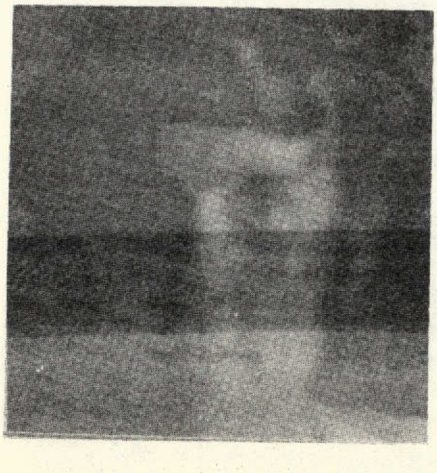

Figure 1.
Face before enhancement.

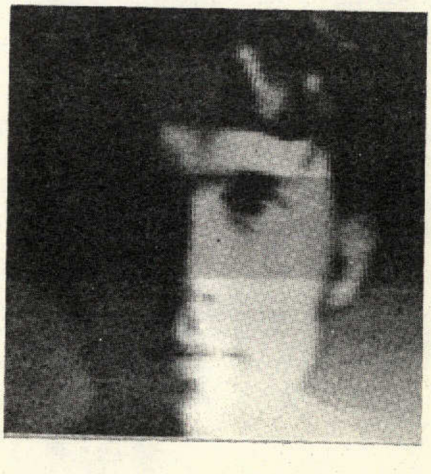

Figure 2.
Face after enhancement.

Result # 2

The picture in Figure 3 was made with the hand sidelighted so that detail in the small fingers and in the thumb was forced into a very small region of the pixel spectrum. Contrast enhancement was used to increase the detail to a visible level. The resulting image is in Figure 4. The mean of the original picture is 114.6; the mean of the enhanced picture is 112.0. The variance of the original picture is 763.2; the variance of the enhanced picture is 5437.9. The commands which were used to enhance the image in Figure 3 are as follows:

```
NOTRANS,INFILE=11,OUTFILE=22,FILENO=3,CONT,END      (5.2)
```

Note that the area around the thumb and small fingers has been improved by this enhancement.

Result # 3

When the picture in Figure 5 was taken, the watch around the arm was obscured in the dark area of the picture. The object of the enhancement procedure was to make this watch visible. The result of the enhancement process is in Figure 6. The mean of the original picture is 112.8; the mean of the resulting picture is 113.3. The variance of the original picture is 368.6; the variance of the resulting picture is 5757.6. The commands which were used to enhance the watch are as follows:

```
NOTRANS,INFILE=12,OUTFILE=23,FILENO=4,CONT,END      (5.3)
```

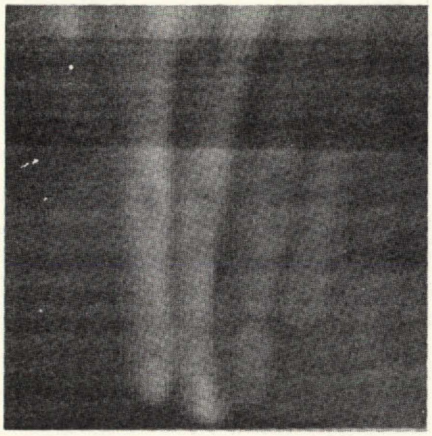

Figure 3.
Hand before enhancement.

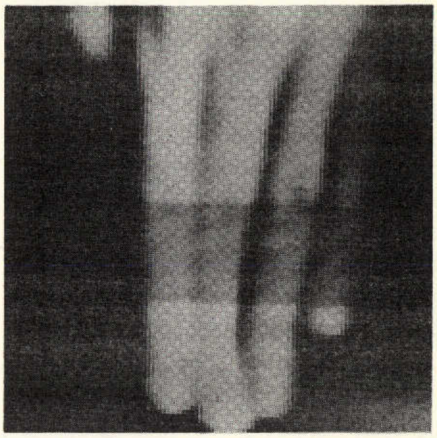

Figure 4.
Hand after enhancement.

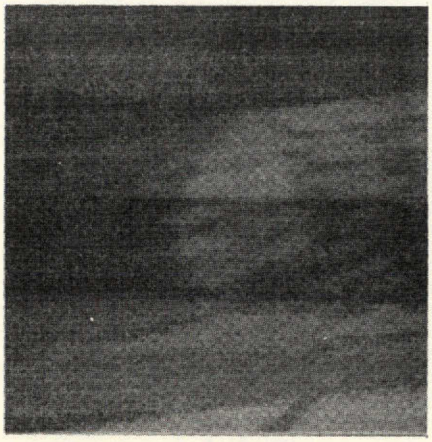

Figure 5.
Watch before enhancement.

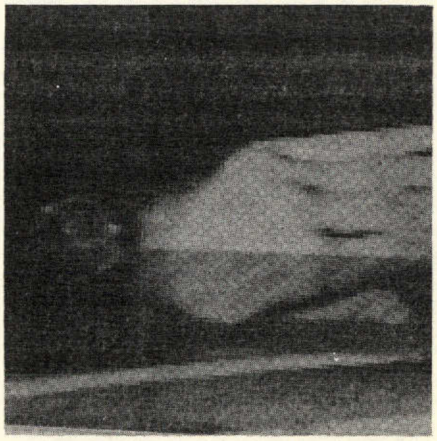

Figure 6.
Watch after enhancement.

Result # 4

Due to sidelighting, the details of the profile shown in Figure 7 are obscured. As can be seen in Figure 8, enhancement reveals these details to the observer; even the teeth are visible. The mean of the original picture is 114.8; the mean of the enhanced picture is 110.9. The variance of the original picture is 1160.7; the variance of the enhanced picture is 4462.3. The commands which were used to enhance the original image are as follows:

```
NOTRANS,INFILE=13,OUTFILE=24,FILENO=5,CONT,END     (5.4)
```

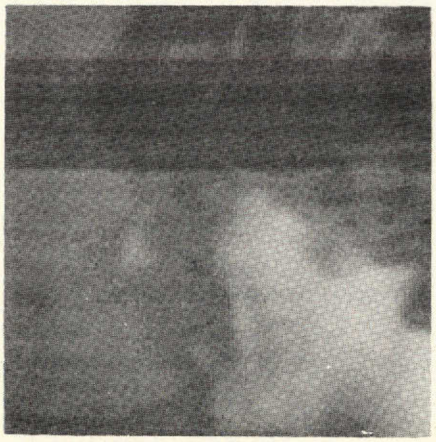

Figure 7.
Profile before enhancement.

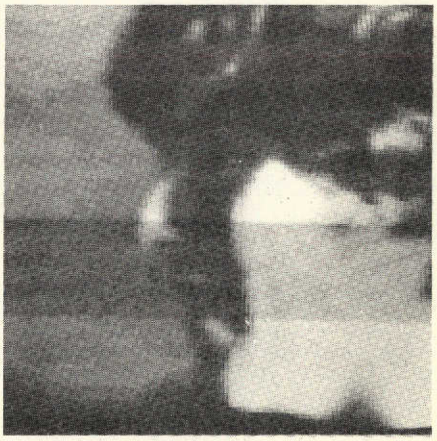

Figure 8.
Profile after enhancement.

Result # 5

In Figure 9, the high brightness levels in the top of the original picture obscure the equipment under the table.

Contrast enhancement is used to obtain the result shown in Figure 10. The next two results are variations of image space enhancement which are provided for comparison. The mean of the original picture is 89.7; the mean of the enhanced picture is 114.2. The variance of the original picture is 4150.2; the variance of the enhanced picture is 4932.1. The commands which were used for enhancement are as follows:

```
    NOTRANS,INFILE=17,OUTFILE=8,FILENO=2,CONT,END      (5.5)
```

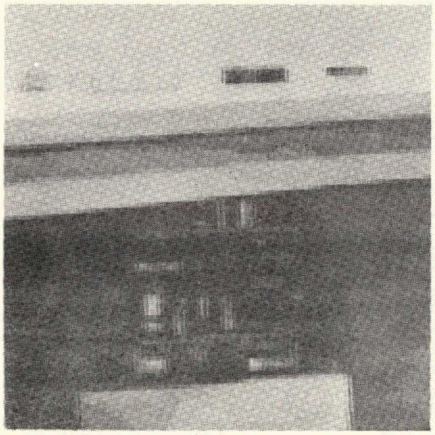

Figure 9.
Table before enhancement.

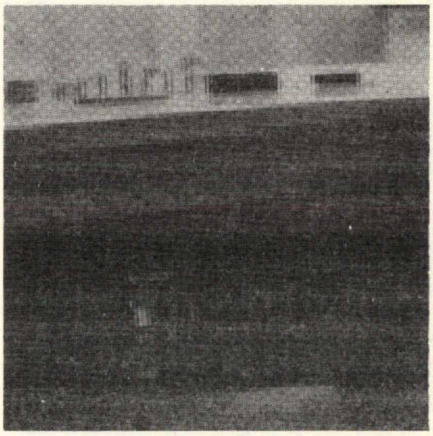

Figure 10.
Table after contrast enhancement.

<u>Result # 6</u>

Histogram equalization was used to enhance the image in Figure 9 to produce the result which appears in Figure 11. This enhancement process is demonstrated here for the purpose of comparison with the preceding and with the following results. The mean of the histogram equalized picture is 126.9; the variance is 5515.7. The commands which were used for enhancement are as follows:

NOTRANS,INFILE=17,OUTFILE=9,FILENO=11,HIST,END (5.6)

Result # 7

In the enhancement performed on the picture in Figure 9, the results of which are shown in Figure 12, a sacrifice of the high levels in the top of the original image was made in order to obtain a large amount of detail in the dark areas under the table. The maximum point in the pixel spectrum histogram was used by the system as the effective mean of the original image. The mean of the enhanced picture is 74.07; the variance is 4150.2. The commands which were used to enhance the original image are as follows:

NOTRANS,INFILE=17,OUTFILE=11,FILENO=5,ENLIGHT,
ENDARK,CONT;70,END (5.7)

Result # 8

The picture shown in Figure 13, which is the same picture as the picture in Figure 9 but is displayed using different techniques, was enhanced using homomorphic filtering to produce the result shown in Figure 14. The filtering was used to decrease the illumination, and thus the dynamic range, of the original image and allow details beneath the table to be viewed. Due to this approach in filtering, the mean and the variance of the enhanced picture are reduced considerably in magnitude. The mean of the resulting picture is 54.8; the variance is 1516.9. The commands which were used to enhance the original picture are as follows:

```
    TRANS,INFILE=17,OUTFILE=28,FILENO=9,END               (5.8)
```

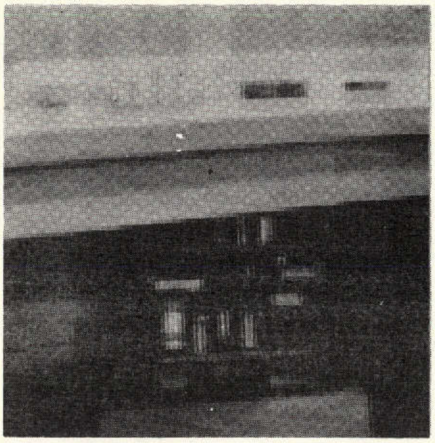

Figure 11.
Table after histogram equalization.

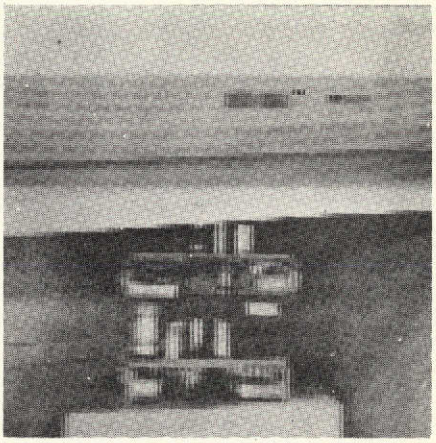

Figure 12.
Table after enhancement of dark areas.

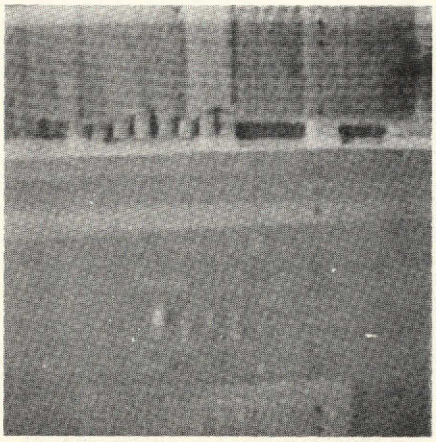

Figure 13.
Table before enhancement.

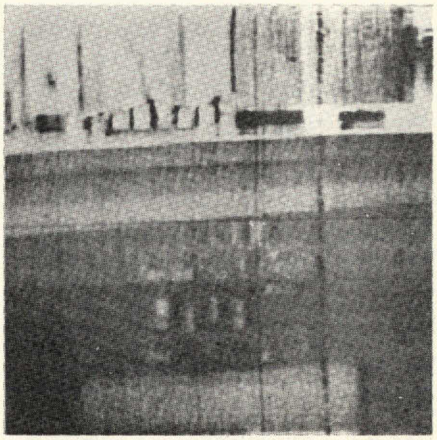

Figure 14.
Table after radial filtering.

LIST OF REFERENCES

LIST OF REFERENCES

1. Hall, E.E., Kruger, R.P., Dwyer, S.J., Hall, D. L., McLaven, R.W., and Lodwick, G.S., "A Survey of Preprocessing and Feature Extraction Techniques for Radiographic Images," IEEE Transactions on Computers, vol. C-20, no.9, pp. 1032-1044.

2. Rudin, Walter, Real and Complex Analysis, New York, New York: McGraw-Hill Book Company, 1966.

3. Stroke, George W., "Optical computing," IEEE Spectrum, Dec. 1972.

4. Oppenheim, A.V., Schafer, R.W., Stockham, T.G., "Nonlinear Filtering of Multiplied and Convolved Signals," Digital Signal Processing, Rabiner, L.R., and Rader, C.M., ed., New York, New York: IEEE Press, pp. 66-93, 1972.

5. Cutrona, L.J., et. al., "Optical Data Processing and Filtering Systems," IRE Trans. Inf. Theory, vol. IT-6, no. 3, pp. 386-400, 1960.

6. Goodman, J.W., Introduction to Fourier Optics, McGraw-Hill Book Co., New York, 1968.

7. Gonzalez, R.C., "The Application of Image Enhancement Techniques to Remote Manipulator Operation," Annual Report, Contract NAS8-29271, National Aeronautics and Space Administration, February, 1974.

8. Mees, C.E., and James, T.H., The Theory of the Photographic Process, MacMillan, New York, 1966.

APPENDIX

APPENDIX

This appendix contains the computer listing of the image enhancement system program which was developed in conjunction with the research reported within this thesis.

The program listing is in two parts. The first listing is an IBM 360 assembler language program which was used to process all language commands and control subsequent program flow. The second listing is a FORTRAN program which performed the requested processing to create enhanced images.

```
         TITLE 'IESLP--MACRO FOR HEXADECIMAL CONVERSION'
         MACRO
&LABEL   UNPACK &LENGTH
         LCLA  &PLACE,&LEN,&CNT
         LCLA  &T1,&T2,&T3,&T4,&T5,&T6
         AIF   (&LENGTH/2*2 NE &LENGTH).MNOTE1
         AIF   (&LENGTH GT 36).MNOTE2
&PLACE   SETA  0
&CNT     SETA  &LENGTH
&T6      SETA  0
&LABEL   SR    0,0
.LOOP    AIF   (&CNT GT 8).SETEQ8
&LEN     SETA  &CNT
         AGO   .GO
.SETEQ8  ANOP
&LEN     SETA  8
.GO      ANOP
&T1      SETA  8+&PLACE
&T2      SETA  2*&LEN-1
&T3      SETA  &T1+&T2-1
&T4      SETA  &T3+1
         UNPK  LINE+&T1.(&T2),&T6.(&LEN,8)
         IC    0,LINE+&T3
         OI    LINE+&T3,X'F0'
         SRL   0,4
         STC   0,LINE+&T4
         OI    LINE+&T4,X'F0'
&PLACE   SETA  &PLACE+2*&LEN
&CNT     SETA  &CNT-&LEN
&T6      SETA  &T6+&LEN
         AIF   (&CNT GT 0).LOOP
&T5      SETA  &LENGTH
&T5      SETA  2*&T5
         TR    LINE+8(&T5),HEX
         AGO   .END
.MNOTE1  MNOTE 12,'****LENGTH IS NOT EVEN****'
         AGO   .END
.MNOTE2  MNOTE 12,'****LENGTH IS GREATER THAN 36****'
.END     MEND
         TITLE 'IMAGE ENHANCEMENT SYSTEM LANGUAGE PROCESSOR'
LANGUAGE START 0
         SAVE  (14,12),,*
         SR    10,10
         BALR  10,0
         USING *,10
         LA    2,SAVEAREA
         ST    2,8(0,13)
         ST    13,4(0,2)
         LR    13,2
*        ENTRY POINT FOR FORTRAN
LANG     EQU   LANGUAGE
         ENTRY LANG
```

```
*        INPUT DATA SET IS FT05F001; OUTPUT DATA SET IS
*        FT06F001
         OPEN  (INPUT,INPUT,OUTPUT,OUTPUT)
*        INPUT INFORMATION IS STORED IN AN ARRAY,
*        THE ADDRESS OF WHICH IS PASSED IN REGISTER 1
*        TO EITHER NOTRAN OR TRAN.
         TITLE 'PARAMETER DEFINITION TABLE FOR IES'
NLIST    EQU   13                NUMBER OF COMMANDS IN TABLE
*    TYPE1 HAS 2 INFORMATION BYTES, BYTE 1 AS IN NSR;
*    BYTE 2 X'00'-X'FF'
TYPE1    EQU   X'10'
TYPE2    EQU   X'08'            1 INFORMATION BYTE,
*                               F'01'-F'255'
TYPE3    EQU   X'04'            1 INFORMATION BYTE, F'01'-F'99'
TYPE4    EQU   X'02'            FULLWORD OF DATA, AS IN INFILE=
TYPE5    EQU   X'01'            END
NSREQ    EQU   X'00'            NON-SPECIFIC REQUEST
         TITLE 'DATA FORMATTING DSECT'
DATA     DSECT
         DS    0F
TRANS    DS    X
OBJ      DS    X
SHARP    DS    X
HIST     EQU   SHARP
DARK     DS    X
LIGHT    DS    X
BAND     DS    X
CONT     DS    X
CONTL    DS    X
INFIL    DS    F
OUTFIL   DS    F
FILENO   DS    F
PRINT    DS    X
         DS    0F
END      DS    AL1
NADDR    EQU   END
         DS    AL3
LASTDATA EQU   *
SIZE     EQU   LASTDATA-DATA
         TITLE 'IMAGE ENHANCEMENT SYSTEM LANGUAGE PROCESSOR'
LANGUAGE CSECT
         USING DATA,11
         LA    11,FIRSTEL
         SR    0,0
         ST    0,AREA
         ST    0,PREV
         MVI   NEWFIL,X'00'               NEXT RECORD IS NEW FI
READ     GET   INPUT,BUFF
         PUT   OUTPUT,BUFF-1
         LA    4,79                       LENGTH
         LA    3,BUFF                     INPUT/OUTPUT BUFFER
         SR    1,1
```

```
DEBLK    EX    4,TRTBLK
         BC    10,OUT                      NO EMBEDDED BLANKS
         LR    0,1
         SR    0,3                         OFFSET FROM S.A.
         SR    4,0                         COMPUTE LENGTH LEFT
         BL    OUT
         LR    3,1                         NEW ADDRESS
         EX    4,TRTNBLK
         BC    8,OUT                       ALL THE REST ARE
*                                          BLANKS
         LR    0,1
         SR    0,3                         NEW OFFSET ADDRESS
         SR    4,0                         NEW LENGTH
         BL    OUT
         EX    4,MVC1
         B     DEBLK
OUT      TM    NEWFIL,X'FF'
         BNZ   LA
         MVI   DATA,X'00'                  NOW BLANK IS CARD DEL
         MVC   DATA+1(SIZE-1),DATA         CLEAR DATA TO ZEROS.
         MVI   NEWFIL,X'FF'
LA       LA    3,BUFF
BACK     CLI   0(3),C' '                   SEE IF NEXT CHAR IS
*                                          A BLANK
         BE    READ                        IF SO READ NEXT CARD
         TRT   0(80,3),COMBLSDI            TEST FOR ',',' ',';',
*                                          OR '0-9'
         BC    6,NOTLAST
         LA    1,80(0,3)
NOTLAST  LR    5,1
         SR    1,3
         BCTR  1,0                         LENGTH-1
         LR    7,1
         SR    0,0
         ST    0,NUMBER
         STC   2,CHAR
         CLI   CHAR,C','
         BE    FINE
         CLI   CHAR,C' '
         BE    FINE
         LR    6,5
         CLI   CHAR,C';'
         BNE   NUMBERS
         LA    6,1(0,6)
NUMBERS  TRT   0(80,6),COMMBLK             TEST FOR ',',' '
         BC    8,ERROR                     NONE FOUND
         LR    5,1
         SR    1,6
         BCTR  1,0                         LENGTH-1
         EX    1,PACK
         CVB   1,DWFLD
         ST    1,NUMBER
         LR    1,7                                LENGTH OF KEYWORD - 1
FINE     CLC   0(2,3),=C'NO'
         BNE   YES
         MVC   0(80,3),1(5)
         B     BACK
YES      MVI   KEY,C' '
         MVC   KEY+1(7),KEY
         EX    1,MVCKEY
         MVC   0(80,3),1(5)
         LA    6,12
         LA    7,TABLE+12*NLIST-1
         LA    4,TABLE
FIND     CLC   0(8,4),KEY
         BE    FOUND
         BXLE  4,6,FIND
         B     ERROR
FOUND    L     9,8(0,4)                           LOAD DISPLACEMENT
         AR    9,11
         TM    8(4),X'FF'
         BNZ   NOPE
         MVI   0(9),X'01'
         B     BACK
NOPE     TM    8(4),X'01'
         BO    ENDODATA
         TM    8(4),X'02'
         BNO   N1
         L     0,NUMBER
         ST    0,0(0,9)
         B     BACK
N1       TM    8(4),X'0C'
         BZ    N2
         L     0,NUMBER
         STC   0,0(0,9)
         B     BACK
N2       TM    8(4),X'10'
         BNO   ERROR
         MVI   0(9),X'01'
         L     0,NUMBER
         STC   0,1(0,9)
         B     BACK
ENDODATA L     1,AREA
         LA    1,1(0,1)
         ST    1,AREA
         C     1,=F'6'
         BL    OK
         SR    1,1
         ST    1,AREA
         GETMAIN R,LV=6*SIZE,SP=0
         ST    1,NADDR
         MVI   END,X'00'
         ST    11,PREV
         LR    11,1
         B      READTO
OK       ST     11,PREV
         LA     1,LASTDATA
         ST     1,NADDR
         MVI    END,X'00'
         LR     11,1
READTO   L      8,PREV
         UNPACK 28
         PUT    OUTPUT,LINE-1
         MVI    NEWFIL,X'00'                 NEW DATA
         B      READ
FIRSTEL  DS     15D
TRTBLK   TRT    0(0,3),BLANKCK
TRTNBLK  TRT    1(0,3),OTHERTHB
MVC1     MVC    0(0,3),0(1)
PACK     PACK   DWFLD,0(0,6)
MVCKEY   MVC    KEY(0),0(3)
         TITLE 'IESLP--TR TABLE FOR BLANKS'
BLANKCK  DC     256X'00'
         ORG    BLANKCK+C' '
         DC     C' '
         ORG
         TITLE 'IESLP--TR TABLE FOR NON-BLANKS'
OTHERTHB DC     256X'FF'
         ORG    OTHERTHB+C' '
         DC     X'00'
         ORG
         TITLE 'IESLP--TR TABLE FOR , '' '' ; 0-9'
COMBLSDI DC     256X'00'
         ORG    COMBLSDI+C','
         DC     C','
         ORG    COMBLSDI+C' '
         DC     C' '
         ORG    COMBLSDI+C';'
         DC     C';'
         ORG    COMBLSDI+C'0'
         DC     C'0123456789'
         ORG
         TITLE 'IESLP--TR TABLE FOR '','','' '''
COMMBLK  DC     256X'00'
         ORG    COMMBLK+C','
         DC     C','
         ORG    COMMBLK+C' '
         DC     C' '
         ORG
         TITLE 'IESLP--TR TABLE FOR HEXADECIMAL CONVERSION'
HEX      DC     256X'40'
         ORG    HEX+C'0'
         DC     C'0123456789ABCDEF'
         ORG
         TITLE 'IESLP--TR TABLE FOR END-SEARCH'
ENDSCRCH DC     256X'00'
```

```
         ORG    ENDSCRCH+C'E'
         DC     C'E'
         ORG
         TITLE 'IESLP--LANGUAGE DEFINITION TABLE'
TABLE    DC     0F'0',CL8'END     ',X'01',AL3(0)
         DC     0F'0',CL8'INFILE= ',X'02',AL3(INFIL-DATA)
         DC     0F'0',CL8'OUTFILE=',X'02',AL3(OUTFIL-DATA)
         DC     0F'0',CL8'FILENO= ',X'02',AL3(FILENO-DATA)
         DC     0F'0',CL8'TRANS   ',X'00',AL3(TRANS-DATA)
         DC     0F'0',CL8'ENOBJ   ',X'04',AL3(OBJ-DATA)
         DC     0F'0',CL8'SHARP   ',X'00',AL3(SHARP-DATA)
         DC     0F'0',CL8'HIST    ',X'00',AL3(HIST-DATA)
         DC     0F'0',CL8'ENDARK  ',X'00',AL3(DARK-DATA)
         DC     0F'0',CL8'ENLIGHT ',X'00',AL3(LIGHT-DATA)
         DC     0F'0',CL8'ENBAND  ',X'08',AL3(BAND-DATA)
         DC     0F'0',CL8'CONT    ',X'10',AL3(CONT-DATA)
         DC     0F'0',CL8'PRINT   ',X'00',AL3(PRINT-DATA)
         TITLE 'IMAGE ENHANCEMENT SYSTEM LANGUAGE PROCESSOR'
CHAR     DS     C
NEWFIL   DS     X
         DC     X'40'
BUFF     DS     XL80
         DC     80X'40'
         DC     X'40'
LINE     DC     80X'40'
ERLINE   DC     C' ',20C'*',C'ERROR',20C'*',35C' '
DWFLD    DS     D
PREV     DS     F
NUMBER   DS     F
KEY      DS     2F
AREA     DS     F
SAVEAREA DS     18F
ERROR    PUT    OUTPUT,ERLINE
         MVI    NEWFIL,X'00'                    CLEAR DATA NEXT PASS
         LA     8,80
TRTEND   TRT    0(80,3),ENDSCRCH
         BC     8,ANOTHER
         CLC    0(3,1),=C'END'
         BE     READ
         MVC    0(80,3),1(1)
         SR     1,3
         BCTR   8,0
         SR     8,1
         BH     TRTEND
ANOTHER  GET    INPUT,BUFF
         PUT    OUTPUT,BUFF-1
         SR     8,8
         B      TRTEND
EOD      L      11,PREV
         LTR    11,11
         BZ     RETURN                          NOTHING TO PASS ON
         MVI    END,X'80'
         LA    11,FIRSTEL
         MVC   LINE(80),LINE-1
         MVI   LINE-1,C'1'
         PUT   OUTPUT,LINE-1
         MVI   LINE-1,C' '
         CLOSE (INPUT,,OUTPUT)
         L     15,=V(IBCOM#)
         BAL   14,64(15)        INITIALIZE FORTRAN ABEND
CALLSEQ  TM    TRANS,X'FF'
         BNZ   TRANCALL
         CALL  NOTRAN,((11))
         B     SEE
TRANCALL CALL  TRAN,((11))
SEE      TM    END,X'80'
         BO    RETURN
         L     11,NADDR
         B     CALLSEQ
RETURN   L     13,4(0,13)
         RETURN (14,12),T
INPUT    DCB   DDNAME=FT05F001,MACRF=GM,RECFM=FB,LRECL=80,
               BLKSIZE=800,DEVD=DA,EODAD=EOD,DSORG=PS
OUTPUT   DCB   DDNAME=FT07F001,MACRF=PM,RECFM=FBA,LRECL=81,
               BLKSIZE=810,DEVD=DA,DSORG=PS
         END
CLEAR    START 0
         SAVE  (14,12),,*
         BALR  10,0
         USING *,10
         LA    2,SAVEAREA
         ST    13,4(2)
         ST    2,8(13)
         LR    13,2
         L     2,CLRAREA
         LA    3,127
         SR    4,4
         STC   4,0(2)
LOOP     MVC   1(256,2),0(2)
         LA    2,256(2)
         BCT   3,LOOP
         MVC   1(255,2),0(2)
         L     13,4(13)
         RETURN (14,12),T
         ENTRY CONV
CONV     SAVE  (14,12),,*
         BALR  10,0
         USING *,10
         LA    2,SAVEAREA
         ST    2,8(13)
         ST    13,4(2)
         LR    13,2
         L     2,0(1)
         L     2,0(2)          LOAD LENGTH
```

```
          L      3,4(1)          LOAD AREA ADDRESS
          L      4,8(1)
          CLI    0(4),C'E'
          BNE    ASCII
          XLATE  (3),(2),TO=E
          B      RET
ASCII     XLATE  (3),(2),TO=A
RET       L      13,4(13)
          RETURN (14,12),T
SAVEAREA  DS     18F
CLRAREA   DC     V(WORKST)
          END
```

```
      SUBROUTINE NOTRAN(DATA)
      INTEGER INFIL,OUTFIL,CONTL/0/,BAND/0/,PL*2(256),
     1   IPIC*2(4096),TABL*2(256)
      INTEGER FILENO
      LOGICAL*1 DATA(28),D,LL,BA(4),CN,CL(4),IN(4),O(4),F(4)
      EQUIVALENCE (INFIL,IN(1)),(OUTFIL,O(1)),(CONTL,CL(1)),
     1   (BAND,BA(1)),(FILENO,F(1))
      REAL CONST(10),MEAN
      COMMON /STOR/IPIC
      READ (3) CONST
      REWIND 3
      D=DATA(4)
      LL=DATA(5)
      BA(4)=DATA(6)
      CN=DATA(7)
      CL(4)=DATA(8)
      DO 10 I=1,4
      F(I)=DATA(16+I)
      IN(I)=DATA(8+I)
   10 O(I)=DATA(12+I)
C         SEE IF HISTOGRAM EQUALIZATION IS REQUESTED.
C         IF SO, ALL
C         OTHER REQUESTS ARE IGNORED.
      IF(DATA(3))GO TO 60
      CALL HIGH(INFIL,PL,MA,&1001)
C         IF BA(4) NE 0, IGNORE D, LL
      IF(.NOT.(BA(4).OR.D.OR.LL))GO TO 40
      IF(BA(4))GO TO 30
      ILOW=1
      IHIGH=256
      MAX=0
      J=0
      MEANPL=0
      IF(.NOT.D)ILOW=129
      IF(.NOT.LL)IHIGH=128
      DO 100 I=ILOW,IHIGH
      MEANPL=MEANPL+PL(I)
      IF(MAX.GE.PL(I))GO TO 100
      MAX=PL(I)
      J=I
  100 CONTINUE
      MEAN=MEANPL/FLOAT(IHIGH-ILOW+1)
      B=MAX-(MAX-MEAN)*CONST(1)
      MEAN=FLOAT(J)
C         CONST(1) IS %/100. OF MAX - MEAN
C         BELOW MAX TO USE IN
C         COMPUTING CONTRAST
C
C         COMPUTE SPREAD
   32 K=0
      DO 200 I=ILOW,IHIGH
      L=J-I
      M=J+I
      IF(L.LE.0)GO TO 120
      IF(PL(L).GT.B)K=I
  120 IF(M.GT.256)GO TO 200
      IF(PL(M).GT.B)K=I
  200 CONTINUE
      SPREAD = FLOAT(K)/256.
      GO TO 5000
   30 ILOW=1
      IHIGH=256
      IL=BAND-2
      IH=BAND+2
  305 IF(IL.LE.0)IL=1
      IF(IH.GT.256)IH=256
      M=0
      DO 31 I=IL,IH
   31 M=M+PL(I)
      IF(M)34,33,34
   33 IL=IL-2
      IH=IH+2
      GO TO 305
   34 MEAN=FLOAT(M)/(IH-IL+1)
      B=MEAN-MEAN*CONST(1)/2.
      MEAN=BAND
      J=BAND
      GO TO 32
 5000 XA=.5
      XB=.5
      AMEAN=MEAN/256.
      ALPHA=(1.-AMEAN)/AMEAN
      BETA=(1.-2.*AMEAN)/AMEAN
      A=AMEAN-SPREAD
      B=AMEAN+SPREAD
      IF(A.GT.0.)XA=ALPHA*A/(BETA*A+1.)
      IF(B.LT.1.)XB=ALPHA*B/(BETA*B+1.)
      C=AMAX1(.5-XA,XB-.5)
C         CONST(2) IS THE UPPER VALUE ON (0.,1.) THAT
C         .5+C IS TO BE
C         MOVED TO.  INITIALLY, CONST(2)=SQRT(2)/4.+.5
      CONS=ABS(TAN((CONST(2)-.5)*3.141593)/TAN(C*3.141593))
C         CONS IS THE CONSTANT TO BE USED IN
C         DETERMINING CONTRAST
C
C         IF NOCONT SPECIFIED, SKIP OVER NEXT PART.
      IF(.NOT.CN)GO TO 50
   35 IF(CONTL.EQ.0)GO TO 50
C         THE CONTRAST VARIES ROUGHLY WITH 1./CONS**2
C         IF CONTL¼128, WANT TO MAKE CONS LARGER
C         IF CONTL¢128, WANT TO MAKE CONS SMALLER
C         LET CONST(3) BE A PROPORTIONALITY
C         CONSTANT THAT MAPS
C         (-1.,1.) ONTO SOME (1/K,K), K¼1 THROUGH EXP(X).
```

```
C         LET K INITIALLY BE 25.  THEN CONST(3)=3.2188758
C         (=LN(25.))
C         INITIALLY.
      CONS=CONS*EXP((CONTL-128.)*CONST(3)/128.)
      GO TO 50
   40 IF(.NOT.CN)GO TO 45
      READ (INFIL,1) IPIC
      CALL STAT(MEAN,VAR,INFIL,&1001)
      WRITE (1,2) MEAN,VAR
    1 FORMAT(32(128A1))
    2 FORMAT(' NOTRAN MEAN = ',1PE13.6,' VARIANCE = ',
     1   1PE13.6)
      REWIND INFIL
      AMEAN=MEAN/256.
      ALPHA=(1.-AMEAN)/AMEAN
      BETA=(1.-2.*AMEAN)/AMEAN
      FA=.5
      FB=.5
   41 X=SQRT(VAR)/256.
      A=AMEAN-X
      B=AMEAN+X
      IF(A.GT.0.)FA=ALPHA*A/(BETA*A+1.)
      IF(B.LT.1.)FB=ALPHA*B/(BETA*B+1.)
      C=AMAX1(.5-FA,FB-.5)
      IF(C.NE.0.)GO TO 42
      PROD=(CONST(2)*256.)**2/VAR
      VAR=CONST(2)
      CONST(2)=CONST(2)*PROD
      GO TO 41
   42 CONS=ABS(TAN((CONST(2)-.5)*3.141593)/TAN(C*3.141593))
      GO TO 35
   45 CONS=1.
      MEAN=128.
   50 CALL CONT(CONS,TABL,MEAN,CONST(4))
      IF(DATA(21)) CALL PLOT(TABL)
C         CONST(4) IS THE POINT TO WHICH THE
C         MEAN VALUE IS MOVED.
C         INITIALLY, CONST(4)=128.
  505 IF(FILENO.EQ.0) FILENO=OUTFIL
      CALL RWLABL(INFIL,OUTFIL,FILENO)
      DO 51 I=1,4
      CALL READ(&1001,INFIL)
      CALL TR(IPIC,TABL)
      CALL WRITE(OUTFIL)
   51 CONTINUE
      REWIND INFIL
      REWIND OUTFIL
      IF(DATA(21)) CALL HIST(OUTFIL,&1001)
      READ(OUTFIL,1) IPIC
      CALL STAT(MEAN,VAR,OUTFIL,&1001)
      WRITE(1,2) MEAN,VAR
      REWIND OUTFIL
      RETURN
 1001 WRITE(1,3)
    3 FORMAT('0NOTRAN I/O ERROR')
      RETURN
   60 CALL HISTEQ(INFIL,TABL,&1001)
      GO TO 505
      END
      SUBROUTINE HIGH(INFIL,DATA,MAX,*)
      INTEGER*2 DATA(256),IPIC(4096)
      COMMON /STOR/IPIC
      READ (INFIL,1) IPIC
    1 FORMAT(32(128A1))
      DO 100 I=1,256
  100 DATA(I)=0
      MAX=0
      DO 200 II=1,4
      CALL READ(&1001,INFIL)
      DO 200 I=1,4096
      J=DATA(IPIC(I)+1)+1
      IF(J.GT.MAX)MAX=J
  200 DATA(IPIC(I)+1)=J
      REWIND INFIL
      RETURN
 1001 RETURN 1
      END
      SUBROUTINE HIST(IUNIT,*)
      INTEGER*2 IHIST(256),STAR/'*'/,DASH(99)/99*'-'/,
     1   LINE(99)/99*' '/
      CALL HIGH(IUNIT,IHIST,MAX,&1001)
      WRITE(6,2000)IUNIT,MAX
 2000 FORMAT('1 HISTOGRAM OF DATA ON UNIT ',I2,5X,
     1   'MAX= ',I5//
     1   14X,'± ',20('----±'),' ±')
      DO 300 I=1,256
      K=100./MAX*IHIST(I)-.5
      J=99-K
      IF(K.LT.1.OR.J.LT.1)GO TO 280
      WRITE(6,3000)I,(DASH(IK),IK=1,K),STAR,
     1   (LINE(IJ),IJ=1,J)
  300 CONTINUE
      GO TO 301
  280 IF(J.LT.1)GO TO 281
      WRITE(6,3000)I,STAR,LINE
      GO TO 300
  281 WRITE(6,3000)I,DASH,STAR
      GO TO 300
  301 WRITE(6,4000)
 3000 FORMAT(6X,I3,5X,'± ',100A1,' ±')
 4000 FORMAT(14X,'± ',20('----±'),' ±')
      RETURN
 1001 RETURN 1
      END
      SUBROUTINE CONT(CONS,IARRAY,M,T)
      INTEGER*2 IARRAY(256)
      REAL MEAN,M
      MEAN=M/256.
      TO=T/256.
      A=(1.-MEAN)/MEAN
      B=(1.-TO)/TO
      C=(1.-2.*MEAN)/MEAN
      D=(1.-2.*TO)/TO
      DO 100 I=1,255
      H=FLOAT(I)/256.
      F=A*H/(C*H+1.)
      G=(F-.5)*3.1415927
      X=ATAN(CONS*TAN(G))
      GI=X/3.1415927+.5
      FI=GI/(B-D*GI)
  100 IARRAY(I+1)=256.*FI+.5
      IARRAY(1)=0
      RETURN
      END
      SUBROUTINE PLOT(ARRAY)
      INTEGER*2 ARRAY(256),TEMP,LLOW,LUP
      LOGICAL*1 MARK,BLK/' '/,STAR/'*'/,PRT(64),CHAR
      WRITE(6,10000)
10000 FORMAT('1')
      DO 100 I=1,64
  100 PRT(I)=BLK
      DO 200 I=1,64
      II=I*4
      IJ=260-II
      LLOW=256-II
      LUP=260-II
      MARK=.FALSE.
      DO 150 J=1,256,4
      JREF=257-J
      CHAR=BLK
      JREFM=JREF-3
      DO 140 K=JREFM,JREF
      TEMP=ARRAY(K)+1
      IF(TEMP.LE.LLOW)MARK=.TRUE.
  140 IF(TEMP.GT.LLOW.AND.TEMP.LE.LUP)CHAR=STAR
      PRT(JREF/4)=CHAR
      IF(MARK)GO TO 160
  150 CONTINUE
  160 WRITE(6,11111)IJ,PRT
  200 CONTINUE
      WRITE(6,22222)
      RETURN
11111 FORMAT(10X,I5,' ±',64A1,'±')
22222 FORMAT(16X,'±',16('---±'))
      END
      SUBROUTINE RWLABL(INFIL,OUTFIL,FILE)
      COMMON /STOR/LABEL
      LOGICAL*1 LABEL(4096),NAME(8),HOLD,TYPEO
      INTEGER FILE,OUTFIL
      INTEGER*2 FILENO,RECCNT
      EQUIVALENCE (LABEL(1),FILENO),(LABEL(7),RECCNT)
C        IPS FILE NAME GIVEN IS 'HSTEQU1$'
      DATA NAME(1)/Z48/,NAME(2)/Z53/,
     1   NAME(3)/Z54/,NAME(4)/Z45/,
     1   NAME(5)/Z51/,NAME(6)/Z55/,
     1   NAME(7)/Z31/,NAME(8)/Z24/,
     1   TYPEO/Z4F/
    1 FORMAT(32(128A1))
C        READ THE LABEL RECORD
      READ(INFIL,1) LABEL
C        INSERT NEW FILE NAME IN LABEL
      DO 100 I=1,8
  100 LABEL(I+8)=NAME(I)
C        SET UP NEW IPS FILE NUMBER
      FILENO=FILE-1
      HOLD=LABEL(1)
      LABEL(1)=LABEL(2)
      LABEL(2)=HOLD
C        FOLLOWING ASSUMES FILE BEING WRITTEN IS
C        AN OS FORMAT FILE.
      RECCNT=4
      HOLD=LABEL(7)
      LABEL(7)=LABEL(8)
      LABEL(8)=HOLD
      LABEL(3)=TYPEO
C        WRITE THE MODIFIED LABEL ON TAPE
      WRITE(OUTFIL,1) LABEL
      RETURN
      END
      SUBROUTINE READ (*,IU)
      LOGICAL*1 LPIC(8192),HOLD
      INTEGER*2 IPIC(4096)
      EQUIVALENCE(IPIC(1),LPIC(1))
      COMMON /STOR/IPIC
    1 FORMAT(32(128A1))
C        ZERO INTEGER ARRAY
      DO 110 I=1,4096
  110 IPIC(I)=0
C        READ A DATA RECORD
      READ(IU,1,END=1000) (LPIC(L),L=2,8192,2)
C        CONVERT DATA VALUES TO CORRECT NUMERIC VALUES
      DO 120 I=1,4096
      IF(IPIC(I).GE.128)IPIC(I)=IPIC(I)-256
  120 IPIC(I)=IPIC(I)+128
      RETURN
 1000 RETURN 1
      END
      SUBROUTINE WRITE(IUNIT)
      LOGICAL*1 LPIC(8192),HOLD
      INTEGER*2 IPIC(4096)
      EQUIVALENCE (IPIC(1),LPIC(1))
      COMMON /STOR/IPIC
    1 FORMAT(32(128A1))
C       CONVERT DATA TO CORRECT FORM
      DO 100 I=1,4096
      IPIC(I)=IPIC(I)-128
  100 IF(IPIC(I).LT.0)IPIC(I)=IPIC(I)+256
      WRITE(IUNIT,1) (LPIC(L),L=2,8192,2)
      RETURN
      END
      SUBROUTINE TR(ARRAY,TABLE)
      INTEGER*2 ARRAY(4096),TABLE(256)
      DO 100 I=1,4096
  100 ARRAY(I)=TABLE(ARRAY(I))
      RETURN
      END
      SUBROUTINE HISTEQ(IDS,TRTAB,*)
      COMMON /STOR/IPIC
      INTEGER*2 IPIC(4096),TRTAB(256)
      REAL TABUL(256)
      READ(IDS,1) IPIC
    1 FORMAT(32(128A1))
      DO 50 I=1,256
   50 TABUL(I)=0.
      DO 100 I=1,4
      CALL READ(&1001,IDS)
      DO 100 J=1,4096
  100 TABUL(IPIC(J)+1)=TABUL(IPIC(J)+1)+1.
      TABUL(1)=.0155640*TABUL(1)
      TRTAB(1)=TABUL(1)+.5
      DO 200 I=2,255
      TABUL(I)=.0155640*TABUL(I)+TABUL(I-1)
  200 TRTAB(I)=TABUL(I)+.5
      TRTAB(256)=255
      REWIND IDS
      RETURN
 1001 RETURN 1
      END
      SUBROUTINE STAT(MEAN,VAR,IDS,*)
      COMMON /STOR/IPIC
      REAL*8 MEANSQ,T,Q
      INTEGER*2 IPIC(4096)
      REAL MEAN
      IMEAN=0
      MEANSQ=0.D0
      DO 100 I=1,4
      CALL READ(&1001,IDS)
      DO 100 J=1,4096
      ITEMP=IPIC(J)
      IMEAN = IMEAN+ITEMP
 100  MEANSQ=MEANSQ+DFLOAT(ITEMP*ITEMP)
      Q=DFLOAT(IMEAN)
      T=Q/16384.D0
      MEAN=T+.5D0
      VAR=MEANSQ/16384.D0-T*T
      RETURN
1001  RETURN 1
      END
      SUBROUTINE LNPIC(IIN,IOUT)
      COMMON/TITLE/LEN,FILNO/WORKST/WORK(8192)
      LOGICAL*1 TRUE/.TRUE./,FALSE/.FALSE./
      INTEGER*2 FILNO
      LEN=0
      FILNO=IIN
      DO 100 I=1,4
      CALL GETDAT(IIN,I,TRUE)
      DO 50 J=1,8192,2
  50  WORK(J)=ALOG(WORK(J))/11.09035+1.000001
      WRITE (IOUT) WORK
 100  CONTINUE
      REWIND IIN
      REWIND IOUT
      RETURN
      ENTRY EXPPIC(IIN,IOUT,IFILE)
      FILNO=IFILE-1
      LEN=0
      DO 200 I=1,4
      READ (IIN) WORK
      DO 150 J=1,8192,2
      WORK(J)=EXP(11.09035*(WORK(J)-1.000001))
 150  WORK(J+1)=0.
      CALL PUTDAT(IOUT,TRUE,I,TRUE)
 200  CONTINUE
      REWIND IIN
      REWIND IOUT
      RETURN
      END
      SUBROUTINE RADFIL(IIN,IOUT,FIL)
      COMMON/TITLE/LEN,FILNO/WORKST/WORK
      LOGICAL*1 TRUE/.TRUE./,FALSE/.FALSE./
      COMPLEX WORK(4096)
      INTEGER*2 FILNO
      REAL FIL(91),LITTLE
      LEN=0
      FILNO=IIN
      DO 100 I=1,4
      K=(I-3)*32
      READ (IIN) WORK
      DO 50 J=1,4096
      L=K+(J-1)/128
      M=64-MOD(J-1,128)
      IREF=SQRT(FLOAT(L*L+M*M))+.5
      IF(IREF.EQ.0)IREF=1
   50 WORK(J)=WORK(J)*FIL(IREF)
      WRITE (IOUT) WORK
  100 CONTINUE
      REWIND IIN
      REWIND IOUT
      RETURN
      END
      SUBROUTINE SPECTR(IFILE,FIL)
      COMMON/WORKST/WORK
      REAL FIL(91)
      COMPLEX WORK(4096)
      DO 100 I=1,91
  100 FIL(I)=0.
      SIZE=0.
      DO 500 I=1,4
      K=(I-3)*32
      READ (IFILE) WORK
      DO 500 J=1,4096
      L=K+(J-1)/128
      M=64-MOD(J-1,128)
      IREF=SQRT(FLOAT(L*L+M*M))+.5
      A=CABS(WORK(J))+FIL(IREF)
      IF(A.GT.SIZE)SIZE=A
  500 FIL(IREF)=A
      DO 1000 I=1,91
 1000 FIL(I)=FIL(I)/A
      REWIND IFILE
      RETURN
      END
      FUNCTION ENVLOP(F)
      REAL F(91),E(13),END(13),CONS/3.023681/
      J=0
      DO 10 I=1,91,7
      J=J+1
   10 END(J)=(F(I)+F(I+1)+F(I+2)+F(I+3)+F(I+4)+F(I+5)+
     1   F(I+6))/7.
      E(1)=END(1)
      DO 20 I=2,12
      A=(END(I-1)+END(I+1))/2.
      B=END(I)
      IF(A.GT.B)B=A
   20 E(I)=B
      E(13)=END(13)
      A=E(1)
      B=.1*A
      DO 30 I=2,13
      J=I
      IF(E(I).LE.B)GO TO 40
   30 CONTINUE
   40 C=E(J)
      ENVLOP=(FLOAT(J)-(C-B)/(C-E(J-1)))*CONS
```

```
C         CONS=-64*SQRT(2)/13/LN(.1)
      RETURN
      END
      SUBROUTINE TRANS(IUNIT,IPUT,FI,HOMOM)
      LOGICAL*1 FALSE/.FALSE./,TRUE/.TRUE./,TF,
     1   FI,NOTFI,HOMOM
      LOGICAL*1 NOTH1,NOTH2
      COMMON /WORKST/WORK
      COMPLEX WORK(128,32),CWORK(4096),C
      EQUIVALENCE (CWORK(1),WORK(1))
      DEFINE FILE 2(512,64,U,IFILE)
      NOTFI=.NOT.FI
      NOTH1=.NOT.(HOMOM.OR.FI)
      NOTH2=.NOT.HOMOM.AND.FI
      SCALE=0.
      DO 30 L=1,4
      IF(NOTH1) GO TO 5
      READ (IUNIT) WORK
      GO TO 7
    5 CALL GETDAT(IUNIT,L,NOTFI)
    7 DO 10 I=1,32
      CALL FFT(WORK(1,I),128,7,FI,FALSE)
   10 CONTINUE
      DO 20 I=1,512,4
      J=(I+3)/4
      WRITE(2'I+L-1)(WORK(J,K),K=1,32)
   20 CONTINUE
   30 CONTINUE
      DO 50 L=1,4
      K=128*(L-1)+1
      READ (2'K) WORK
      DO 40 I=1,32
      CALL FFT(WORK(1,I),128,7,FI,TRUE,SCALE)
   40 CONTINUE
      WRITE (2'K) WORK
   50 CONTINUE
      IF(NOTFI)SCALE=SCALE+SCALE
      WRITE(1,11111)SCALE
11111 FORMAT(' SCALE = ',1PE13.6)
   55 DO 90 L=1,4
      DO 60 I=1,512,4
      J=(I+3)/4
      READ (2'I+L-1)(WORK(J,K),K=1,32)
   60 CONTINUE
      FIND (2'L+1)
      DO 70 I=1,4096
      C=CWORK(I)/SCALE
      IF(NOTFI) GO TO 70
      A=REAL(C)
      IF(A.GT..9999847)A=.9999847
      B=AIMAG(C)
      IF(B.GT..9999847)B=.9999847
      C=CMPLX(A,B)
   70 CWORK(I)=C
      IF(NOTH2)GO TO 75
      WRITE (IPUT) WORK
      GO TO 90
   75 CALL PUTDAT(IPUT,FI,L,FI)
   90 CONTINUE
      RETURN
      END
      SUBROUTINE IOMESS(WHERE,IUNIT,ITYP,DSN,NAME,
     1                  NREC,IPLACE)
      LOGICAL*1 WHERE(14),ITYP(2),NAME(8),CCNV(10),XTYP(2),
     1   XNAME(8)
      EQUIVALENCE (CCNV(1),XTYP(1)),(CCNV(3),XNAME(1))
      INTEGER*2 DSN,NREC
C        DS UNIT 1 IS OUTPUT DEVICE FOR DATA SET
C        UTILIZATION MESSAGES
      CCNV(1)=ITYP(1)
      CCNV(2)=ITYP(2)
      DO 10 I=1,8
   10 XNAME(I)=NAME(I)
      CALL CONV(10,CCNV,'E')
      WRITE(1,11111)WHERE,IUNIT,XTYP,DSN,XNAME,NREC,IPLACE
11111 FORMAT(' ',14A1,' UNIT NO. = ',I2,'; TYPE=',2A1,
     1   '; FILE NUMBER ',
     1   I4,'; NAME:  ''',8A1,
     1   '''; NUMBER OF DATA RECORDS = ',I4,
     2   ' RECNO = ',I4)
      RETURN
      END
      SUBROUTINE GETDAT(IUNIT,IPLACE,FORM)
      COMMON /WORKST/WORK
      INTEGER*2 DSN,NREC,IT/0/,O/Z4F/,OT(3)/Z46,Z43,Z54/,
     1   IWORK*4(1)
      COMPLEX WORK(1)
      EQUIVALENCE (DSN,NUM(1)),(NREC,RECCNT(1)),(IT,T(1)),
     1   (IWORK(1),WORK(1),LWORK(1))
      LOGICAL*1 ITYP(2),NUM(2),NAME(8),RECCNT(2),LWORK(1),
     1   T(2),FORM
      IF(IPLACE.NE.1)GO TO 4
      REWIND IUNIT
      CALL RLABEL(IUNIT,ITYP,NUM,NAME,RECCNT)
    4 CALL IOMESS('GETDAT  INPUT ',IUNIT,ITYP,DSN,NAME,NREC,
     1   IPLACE)
      CALL CLEAR
      T(2)=ITYP(1)
C        SEE IF IT IS AN ORIGINAL PICTURE.
      IF(O.EQ.IT)GO TO 10
      DO 5 I=1,3
      IF(IT.EQ.OT(I))GO TO20
C        IT IS A COMPLEX IMAGE
    5 CONTINUE
```

```
C         ERROR--WRONG TYPE
      CALL CONV(2,ITYP,'E')
      WRITE(1,1)ITYP
      CALL EXIT
    1 FORMAT('0GETDAT  --IMPROPER FILE TYPE : TYPE = ',2A1)
C         IT IS AN ORIGINAL PICTURE
   10 READ(IUNIT,2)(LWORK(K),K=4,32771,8)
    2 FORMAT(32(128A1))
   11 CONTINUE
      DO 12 I=1,4096
      J=IWORK(I+I-1)
      IF(J.GE.128)J=J-256
C         IF FORM=.TRUE., ALL DATA PTS 0¢=X¢=1.
      IF(FORM)J=J+128
C         OTHERWISE,ALL DATA PTS -.5¢=X¢=.5
   12 WORK(I)=CMPLX(FLOAT(J)/256.,0.)
      RETURN
C         IT IS A COMPLEX PICTURE
   20 DO 21 I=3,32768,8192
      J=I+8191
      II=I+4
      JJ=II+8191
      READ(IUNIT,2)(LWORK(K+1),LWORK(K),K=I,J,8),
     1   (LWORK(K+1),LWORK(K),K=II,JJ,8)
   21 CONTINUE
      DO 22 I=1,4096
      II=I+I
      J=IWORK(II-1)
      K=IWORK(II)
      IF(J.GE.32768)J=J-65536
      IF(K.GE.32768)K=K-65536
C         IF FORM=.TRUE., ALL DATA PTS 0¢=X¢=1.
      IF(.NOT.FORM)GO TO 22
      J=J+32768
      K=K+32768
C         OTHERWISE, ALL DATA PTS -.5¢=X¢=.5
   22 WORK(I)=CMPLX(FLOAT(J)/65536.,FLOAT(K)/65536.)
      RETURN
      END
      SUBROUTINE PUTDAT(IUNIT,FI,IPLACE,FORM)
      COMMON/TITLE/LEN,FILNO,TITLE/LABEL/LABEL/WORKST/WORK
      COMPLEX WORK(1)
      REAL RWORK(1)
      LOGICAL*1 LABEL(16),NAME(8),ITYP(2)/Z00,Z53/,
     1   F/Z46/,C/Z43/,
     1   U/Z3E/,D/Z3C/,T/ZFF/,LWORK(1),FI,TITLE(1),FORM
      INTEGER*2 FILNO,NREC/16/,IWORK*4(1),TEMP1
      EQUIVALENCE (LABEL(9),NAME(1)),
     1   (WORK(1),IWORK(1),LWORK(1),
     1   RWORK(1)),(ITEMP,TEMP1)
      IF(IPLACE.NE.1)GO TO 20
      ITYP(1)=C
      NAME(8)=D
      IF(FI)GO TO 10
      ITYP(1)=F
      NAME(8)=U
   10 CALL WLABEL(IUNIT,ITYP,FILNO,NAME,LEN,TITLE)
   20 CALL IOMESS('PUTDAT  OUTPUT',IUNIT,ITYP,FILNO,NAME,
     1   NREC,IPLACE)
      DO 11 I=1,8192
      ITEMP=RWORK(I)*65536.+.5
C        IF FORM=.TRUE., ALL DATA PTS 0¢=X¢=1.
      IF(FORM)ITEMP=ITEMP-32768
C        OTHERWISE, ALL DATA PTS -.5¢=X¢=.5
      IF(ITEMP.LT.0)ITEMP=ITEMP+65536
      IF(ITEMP.GT.65535)ITEMP=65535
   11 IWORK(I)=ITEMP
      DO 12 I=3,32768,8192
      J=I+8191
      II=I+4
      JJ=II+8191
      WRITE(IUNIT,1)(LWORK(K+1),LWORK(K),K=I,J,8),
     1   (LWORK(K+1),LWORK(K),K=II,JJ,8)
   12 CONTINUE
    1 FORMAT(32(128A1))
      RETURN
      END
      SUBROUTINE FFT(A,SIZ,M,INV,NYS,/SCALE/)
      INTEGER SIZ
      COMPLEX A(SIZ),U,W,T
      LOGICAL*1 INV,NYS,NOT
      NOT=.NOT.NYS
      PI=3.141593
      N=2**M
      NV2=N/2
      NM1=N-1
      J=1
      DO 7 I=1,NM1
      IF(I.GE.J)GO TO 5
      T=A(J)
      A(J)=A(I)
      A(I)=T
    5 K=NV2
    6 IF(K.GE.J)GO TO 7
      J=J-K
      K=K/2
      GO TO 6
    7 J=J+K
      IF(INV)GO TO 9
      KNV2=NV2+1
      DO 8 L=KNV2,N
    8 A(L)=-A(L)
    9 DO 20 L=1,M
      LE=2**L
      LE1=LE/2
      U=(1.0,0.)
      W=CEXP(CMPLX(0.,PI/LE1))
      IF(INV)W=CONJG(W)
      DO 20 J=1,LE1
      DO 10 I=J,N,LE
      IP=I+LE1
      T=A(IP)*U
      A(IP)=A(I)-T
   10 A(I)=A(I)+T
   20 U=U*W
      IF(.NOT.INV.AND..NOT.NYS)RETURN
      AK=FLOAT(N)
      IF(INV)AK=1.
      DO 30 I=1,N
      T=A(I)/AK
      IF(NOT)GO TO 29
      X=ABS(REAL(T))
      Y=ABS(AIMAG(T))
      SCALE=AMAX1(SCALE,X,Y)
   29 IF(INV.AND.MOD(I,2).EQ.1)T=-T
   30 A(I)=T
      RETURN
      END
      SUBROUTINE RLABEL(IUNIT,ITYP,NUM,NAME,RECCNT)
      COMMON /LABEL/LABEL
      LOGICAL*1 LABEL(4096),NUM(2),NAME(8),ITYP(2),RECCNT(2)
    1 FORMAT(32(128A1))
C        READ THE LABEL RECORD
      READ(IUNIT,1)LABEL
C        TRANSFER FILE NAME
      DO 100 I=1,8
  100 NAME(I)=LABEL(I+8)
C        TRANSFER FILE NUMBER
      NUM(1)=LABEL(2)
      NUM(2)=LABEL(1)
C        TRANSFER FILE TYPE
      ITYP(1)=LABEL(3)
      ITYP(2)=LABEL(4)
C        TRANSFER NUMBER OF RECORDS
      RECCNT(1)=LABEL(8)
      RECCNT(2)=LABEL(7)
      RETURN
      END
      SUBROUTINE WLABEL(IUNIT,ITYP,NUM,NAME,TITLEN,TITLE)
      COMMON /LABEL/LABEL
      INTEGER TITLEN,SIZE,RECCNT*2
      LOGICAL*1 TITLE(TITLEN),BELL/Z07/,ITYP(2),
     1   NAME(8),NUM(2),LABEL(4096)
      INTEGER TYP(3)/Z53,Z4D,Z4C/,FORM(4)/Z4F,Z46,Z43,Z54/,T
      LOGICAL*1 TE(4)/4*Z00/,XQ
      EQUIVALENCE (TE(1),T),(TE(4),XQ)
      EQUIVALENCE (RECCNT,LABEL(7))
C        CHANGE TITLE
      IF(TITLEN.LT.2)GO TO 100
      IF(TITLEN.GT.1025)TITLEN=1025
      J=TITLEN-1
      DO 50 I=1,J
   50 LABEL(I+64)=TITLE(I)
      IF(TITLEN.LT.1025)LABEL(64+TITLEN)=BELL
C        TRANSFER FILE NAME
  100 DO 150 I=1,8
  150 LABEL(I+8)=NAME(I)
C        TRANSFER FILE NUMBER
      LABEL(1)=NUM(2)
      LABEL(2)=NUM(1)
C        TRANSFER FILE TYPE
      LABEL(3)=ITYP(1)
      XQ=ITYP(2)
      LABEL(4)=XQ
C        FIND SIZE OF PICTURE
      DO 200 I=1,3
      IF(T.NE.TYP(I))GO TO 200
      SIZE=128*2**(I-1)
      GO TO 250
  200 CONTINUE
      WRITE(1,2)
    2 FORMAT('0DS TYPE ERROR')
      CALL EXIT
C        DETERMINE DS TYPE
  250 XQ=ITYP(1)
      IF(T.NE.FORM(1))GO TO 300
      SIZE=SIZE*SIZE
      GO TO 400
  300 DO 350 I=2,4
      IF(T.NE.FORM(I))GO TO 350
      SIZE=SIZE*SIZE*4
      GO TO 400
  350 CONTINUE
      WRITE(6,2)
      CALL EXIT
C        DETERMINE NUMBER OF RECORDS
  400 RECCNT=SIZE/4096
      XQ=LABEL(7)
      LABEL(7)=LABEL(8)
      LABEL(8)=XQ
      WRITE(IUNIT,1)LABEL
    1 FORMAT(32(128A1))
      RETURN
      END
      SUBROUTINE TRAN(DATA)
      LOGICAL*1 DATA(28)
      INTEGER INFIL,OUTFIL,FILENO,OBJ/0/,IPIC*2(4096),
     1   FILNO*2
```

```
      LOGICAL*1 SHARP,DARK,LIGHT,OBJLOG(4),IN(4),OUT(4),
     1   FIL(4),MARK,TRUE/.TRUE./,FALSE/.FALSE./
      EQUIVALENCE (INFIL,IN(1)),(OUTFIL,OUT(1)),
     1   (FILENO,FIL(1)),
     1   (OBJ,OBJLOG(1))
      REAL CONST(10),MEAN,FILCUV(91),FREQ(91)
      COMMON /TITLE/LEN,FILNO
      EXPMOD(X)=EXP(AMAX1(0.,X))
      READ (3) CONST
      REWIND 3
      SHARP=DATA(3)
      DARK=DATA(4)
      LIGHT=DATA(5)
      OBJLOG(4)=DATA(2)
      DO 10 I=1,4
      FIL(I)=DATA(16+I)
      IN(I)=DATA(8+I)
   10 OUT(I)=DATA(12+I)
      CALL CSTAT(MEAN,VAR,INFIL)
      WRITE(1,2) MEAN,VAR
      MARK=DARK.OR.LIGHT.OR..NOT.SHARP
C        DATA SETS 8 AND 9 ARE RESERVED FOR
C        INTERMEDIATE RESULTS
C        RECFM=F, BLKSIZE=4096, AND SPACE FOR 17
C        RECORDS EACH.
      IF(.NOT.MARK) GO TO 20
C        TAKE LN OF IMAGE
      CALL LNPIC(INFIL,8)
      INFIL=8
   20 CALL TRANS(INFIL,9,FALSE,MARK)
      REWIND INFIL
      REWIND 9
      INFIL=9
      CALL SPECTR(INFIL,FREQ)
      T=ENVLOP(FREQ)
      SIZE=OBJ/100.
      IF(.NOT.MARK) GO TO 50
      IF(DARK.AND.LIGHT) GO TO 30
      IF(MEAN.LE.128..AND.LIGHT)
     1   MEAN=MEAN+CONST(6)*(256.-MEAN)
      IF(MEAN.GE.128..AND.DARK)MEAN=MEAN*(1.-CONST(6))
C        CONST(6) IS A PROPORTIONALITY CONSTANT
C        WHICH OFFSETS THE
C        MEAN WHEN THE OPERATOR SPECIFIES
C        A DIFFERENT ENHANCEMENT
   30 START = (128./MEAN)**CONST(9)*(1.-ABS((128.-MEAN)
     1   /128.))**2/VAR/CONST(10)*T-1.
      PERSIZ=T
      IF(SIZE.NE.0.)PERSIZ=1./SIZE*CONST(7)
C        CONST(7) IS THE CONSTANT THAT RELATES T TO SIZE,
C        1./SIZE*CONST(7)=T IDEALLY.
      K=PERSIZ+.5
      DELAY=FREQ(1)/FREQ(K)*CONST(8)
C        CONST(8) IS THE CONSTANT CONTROLLING THE FACTOR BY
C        WHICH THE FREQUENCIES AFTER PERSIZ ARE MULTIPLIED
C        FOR SHARPENING.  INITIALLY, CONST(8)=5.
      IF(PERSIZ.LE.T)GO TO 31
      C=ALOG(DELAY)/(PERSIZ-T)
      GO TO 32
   31 C=0.
   32 SQVAR=54.62742/SQRT(VAR)*CONST(5)
      DO 40 I=1,91
      FI=FLOAT(I)
      F1=1.+START*EXP(-FI/T)
      F2=EXPMOD(SQVAR*(FI-T))
C        CONST(5) IS THE CONSTANT CONTROLLING
C        THE COMPENSATION
C        RISING EXPONENTIAL.  INITIALLY, CONST(5)=.04913448.
      F3=EXPMOD((FI-T)*C)/EXPMOD((FI-PERSIZ)*C)
   40 FILCUV(I)=F1*F2*F3
  401 CALL RADFIL(INFIL,8,FILCUV)
      INFIL=8
      IOUT=9
      IF(.NOT.MARK)IOUT=OUTFIL
      FILNO=FILENO-1
      LEN=0
      CALL TRANS(INFIL,IOUT,TRUE,MARK)
      REWIND INFIL
      REWIND IOUT
      IF(MARK)GO TO 41
      IF(DATA(21))CALL CURVE(FILCUV)
      IF(DATA(21))CALL CHIST(IOUT,2,FALSE)
      IF(.NOT.DATA(21)) WRITE(1,3) FILCUV
    3 FORMAT(' FILCUV = ',5(1PE13.6,2X)/(8X,5(2X,1PE13.6)))
      CALL CSTAT(MEAN,VAR,OUTFIL)
      WRITE(1,2) MEAN,VAR
      RETURN
   41 CALL EXPPIC(IOUT,OUTFIL,FILENO)
      REWIND IOUT
      REWIND OUTFIL
      IF(DATA(21)) CALL CURVE(FILCUV)
      IF(DATA(21)) CALL CHIST(OUTFIL,2,FALSE)
      IF(.NOT.DATA(21))WRITE(1,3) FILCUV
      CALL CSTAT(MEAN,VAR,OUTFIL)
      WRITE(1,2) MEAN,VAR
      RETURN
   50 PERSIZ=T
      IF(SIZE.NE.0.)PERSIZ=1./SIZE*CONST(7)
      K=PERSIZ+.5
      DELAY=FREQ(1)/FREQ(K)*CONST(8)
      IF(PERSIZ.LE.T)GO TO 51
      C=ALOG(DELAY)/(PERSIZ-T)
      GO TO 52
   51 C=0.
```

```
 52 SQVAR=54.62742/SQRT(VAR)*CONST(5)
    DO 60 I=1,91
    FI=FLOAT(I)
    F1=EXPMOD(SQVAR*(FI-T))
 60 FILCUV(I)=EXPMOD((FI-T)*C)/EXPMOD((FI-PERSIZ)*C)*F1
    GO TO 401
  2 FORMAT(' TRAN  MEAN = ',1PE13.6,' VARIANCE = ',
   1   1PE13.6)
    END
    SUBROUTINE CHIST(IUNIT,OP,BYP)
    COMPLEX WORK(1),CTEMP
    COMMON /WORKST/WORK
    LOGICAL*1 BYP
    INTEGER*2 IHIST(256),STAR/'*'/,DASH(99)/99*'-'/,
   1   LINE(99)/99*' '/,OP*4
    INTEGER DISP(4,4)/'LOG-','MAGN','ITUD','E   ','REAL',
   1   3*'    ',
   1   'IMAG','INAR','Y   ','    ','PHAS','E   ',2*'    '/
    LOGICAL*1 TRUE/.TRUE./
C       OP DETERMINES THE DISPLAY MODE TO BE USED.
C       OP=1 -¼ LOG MAGNITUDE OF VALUE IS USED
C       OP=2 -¼ REAL PART OF VALUE IS USED
C       OP=3 -¼ IMAGINARY PART OF VALUE IS USED
C       OP=4 -¼ PHASE OF VALUE IS USED
C       ANY OTHER NUMBER -¼ OP=1
C       IN THE GENERATION OF THE HISTOGRAM,
C       IF BYP=.TRUE., CHIST
C       ASSUMES THE DATA IS ALREADY IN A 360-COMPLEX FORMAT
    DO 100 I=1,256
100 IHIST(I)=0
    MAX=0
    DO 200 IPT=1,4
    IF(BYP)GO TO 20
    CALL GETDAT(IUNIT,IPT,TRUE)
    GO TO 30
 20 IF(IPT.EQ.1)REWIND IUNIT
    READ(IUNIT)(WORK(I),I=1,4096)
 30 DO 200 I=1,4096
    GO TO (110,120,130,140),OP
110 TEMP=(REAL(WORK(I))*REAL(WORK(I))+
   1   AIMAG(WORK(I))*AIMAG(WORK(I)))
   1   /2.
    IF(TEMP)115,111,115
111 J=255
    GO TO 150
115 J=-11.12624*ALOG(TEMP)+.5
    GO TO 150
120 J=255.*REAL(WORK(I))+.5
    GO TO 150
130 J=255.*AIMAG(WORK(I))+.5
    GO TO 150
140 CTEMP=WORK(I)
```

```
      IF(REAL(CTEMP))145,141,145
  141 TEMP=AIMAG(CTEMP)
      IF(TEMP)144,142,144
  142 J=128
      GO TO 150
  144 J=SIGN(128.,TEMP)+128.
      GO TO 150
  145 J=81.169*(ATAN2(AIMAG(CTEMP),REAL(CTEMP))+1.570796)+.5
  150 K=IHIST(J+1)+1
      IF(K.GT.MAX)MAX=K
  200 IHIST(J+1)=K
      REWIND IUNIT
      WRITE(6,2000)IUNIT,MAX,(DISP(ID,OP),ID=1,4)
 2000 FORMAT('1 HISTOGRAM OF DATA ON UNIT ',I2,5X,
     1   'MAX= ',I5,
     1   5X,4A4,'DISPLAY'///
     2   14X,'± ',20('----±'),' ±')
      DO 300 I=1,256
      K=100./MAX*IHIST(I)-.5
      J=99-K
      IF(K.LT.1.OR.J.LT.1)GO TO 280
      WRITE(6,3000)I,(DASH(IK),IK=1,K),STAR,
     1   (LINE(IJ),IJ=1,J)
  300 CONTINUE
      GO TO 301
  280 IF(J.LT.1)GO TO 281
      WRITE(6,3000)I,STAR,LINE
      GO TO 300
  281 WRITE(6,3000)I,DASH,STAR
      GO TO 300
  301 WRITE(6,4000)
 3000 FORMAT(6X,I3,5X,'± ',100A1,' ±')
 4000 FORMAT(14X,'± ',20('----±'),' ±')
      RETURN
      END
      SUBROUTINE CSTAT(MEAN,VAR,IDS)
      LOGICAL*1 TRUE/.TRUE./
      COMMON/WORKST/WORK
      COMPLEX WORK(4096)
      REAL*8 MEANSQ,T,Q,MEAN*4,DMEAN,TEMP
      DMEAN=0.D0
      MEANSQ=0.D0
      DO 100 I=1,4
      CALL GETDAT(IDS,I,TRUE)
      DO 100 J=1,4096
      TEMP=DBLE(CABS(WORK(J)))
      DMEAN=DMEAN+TEMP
  100 MEANSQ=MEANSQ+TEMP*TEMP
      REWIND IDS
      Q=MEANSQ*4.D0
      T=DMEAN/64.D0
      MEAN=SNGL(T)
      VAR=SNGL(Q-T*T)
      RETURN
      END
      SUBROUTINE CURVE(FIL)
      REAL FIL(91)
      INTEGER*2 LINE(91)/91*' '/,STAR/'*'/,
     1   POINT*4(91),IFIL(91),
     1   BLK/' '/
      BIG=0.
      DO 100 I=1,91
  100 IF(FIL(I).GT.BIG)BIG=FIL(I)
      INDEX=101
      WRITE(6,10000) BIG
10000 FORMAT('1',10X,'MAXIMUM IS ',1PE13.6)
      DO 200 I=1,91
  200 IFIL(I)=FIL(I)/BIG*100.+.5
      IP=1
      DO 600 I=1,100
      INDEX=INDEX-1
      DO 300 J=1,91
      IF(IFIL(J).NE.INDEX)GO TO 300
      LINE(J)=STAR
      POINT(IP)=J
      IP=IP+1
  300 CONTINUE
      IF(IP.EQ.1)GO TO 500
      WRITE(6,11111)INDEX,LINE
11111 FORMAT(' ',5X,I3,2X,91A1)
      IP=IP-1
      DO 400 K=1,IP
  400 LINE(POINT(K))=BLK
      IP=1
      GO TO 600
  500 WRITE(6,11111)INDEX
  600 CONTINUE
      RETURN
      END
```

VITA

John D. Birdwell [redacted] [redacted] He attended school through the sixth grade in Knox County, and afterward in Oak Ridge. He left Oak Ridge High School at the end of his junior year and started working on his Bachelor of Science degree in Electrical Engineering at the University of Tennessee in the fall of 1971. He graduated first in the College of Engineering at the end of the winter quarter, 1974, during which he had begun his work toward a Master of Science degree in Electrical Engineering at the University of Tennessee. He fulfilled the requirements for this degree during the summer of 1974, and received the degree in December, 1974.