NASA CR- 170440 — E83-10174
CR-170440

URBAN AREA CHANGE DETECTION PROCEDURES WITH REMOTE SENSING DATA

Final Report Under Contract No. NAS5-26127

(E83-10174) URBAN AREA CHANGE DETECTION PROCEDURES WITH REMOTE SENSING DATA Final Report (Colorado State Univ.) 79 p HC A05/MF A01 CSCL 05B

N83-17937

Unclas
G3/43 00174

Prepared for

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

By

Eugene L. Maxwell, Principal Investigator

and

Carol J. Riordan

Earth Resources Department
Colorado State University
Fort Collins, Colorado 8[illegible]3

November 26, 1980

URBAN AREA CHANGE DETECTION PROCEDURES
WITH REMOTE SENSING DATA

Final Report Under Contract No. NAS5-26127

Prepared for

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

By

Eugene L. Maxwell, Principal Investigator

and

Carol J. Riordan

Earth Resources Department
Colorado State University
Fort Collins, Colorado 80523

November 26, 1980

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

The research reported here was supported by the NASA/Goddard Space Flight Center, Greenbelt, MD, under Contract No. NAS5-26127 and is a continuation of previous work on land cover change detection supported by the Colorado Agricultural Research Experiment Station, Colorado State University, Fort Collins, and by NASA/Goddard Space Flight Center under Contract No. NAS5-25696.

We would like to thank Mr. David Toll at NASA/Goddard Space Flight Center for his support of this research.

ABSTRACT

Detection of a change in the urban boundary has been described as an important element in the U. S. Census Bureau's census-taking procedure. The use of satellite data, such as Landsat, for this task may be appropriate. This research concentrated on developing an understanding of several underlying factors affecting the detection and identification of nonurban to urban land cover change using satellite data. In particular, computer programs were developed to create a digital scene and to simulate the effect of the sensor Point Spread Function (PSF) on the transfer of modulation from the scene to an image of the scene. The theory behind the development of a digital filter representing the PSF is given as well as an example of its application. Atmospheric effects on modulation transfer are also discussed. A User's Guide and program listings are given in an appendix.

1.0 BACKGROUND

Detection of a change in the urban boundary has been described as an important element in the U. S. Census Bureau's census-taking procedure (Christenson, et al., 1977; Christenson and Lachowski, 1977; Christenson, et al., 1978; Computer Sciences Corporation, 1978; Friedman, 1980). The use of satellite data, such as Landsat, to monitor the urban boundary appears to be appropriate because of the repetitive, small scale coverage of large areas. In addition, the digital format of the data makes it amenable to automated processing. The task is thus to study methods for using digital spectral data to detect, identify, and monitor the change from non-urban to urban cover on the urban fringe.

This research concentrated on developing an understanding of the underlying factors affecting the detection and identification of non-urban to urban land cover change using satellite data. These underlying factors were identified as the amplitude and spatial frequency of scene modulation on the urban fringe, the transfer of scene modulation to the image by the sensor, and the introduction of noise from the atmosphere and the electronics systems. Thus, the system being analyzed includes modulation of solar irradiance by scene objects and the transfer of the modulation by the atmosphere, the optics and the electronics of the sensor. A physical system diagram which illustrates the underlying factors affecting land cover change detection is given in Figure 1.1. It is instructive to consider the modulation of scene radiance as a signal carrying useful information. In this instance, it is expected that the modulation will signal a change from non-urban to urban land use.

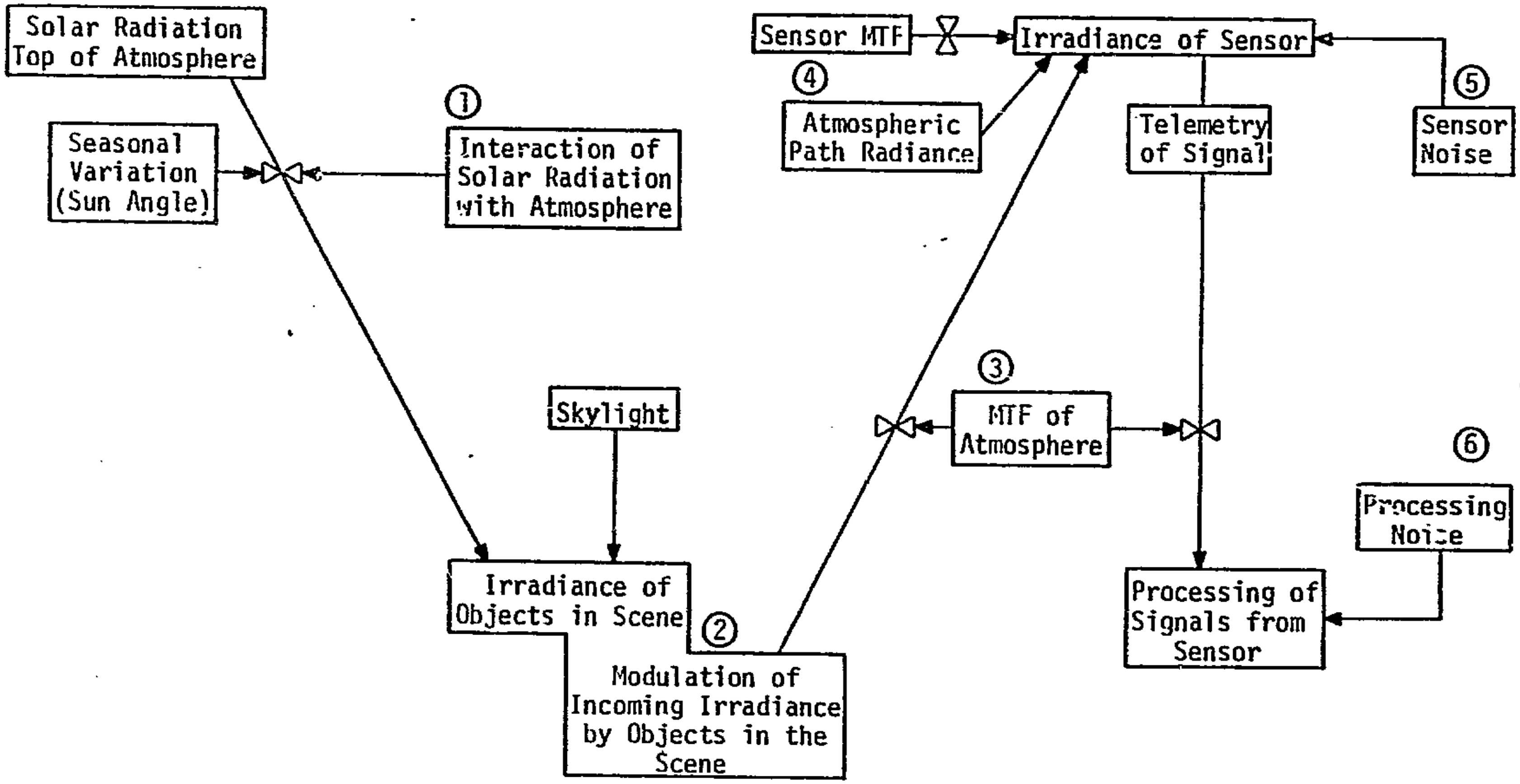


Figure 1.1 Physical system diagram.

1.1 Definition of the Change Signal

Using remote sensing methods the change from non-urban to urban land use must be signaled by a change in land cover. The change in land cover will in turn be signaled by changes in the spatial and spectral characteristics of the cover types which modulate the scene irradiance. If moderate scale photography is used, even large tract (1 to 20 acre lots) developments can readily be detected by interpreter recognition of houses and the roads connecting them (spatial features). Landsat's 76 X 76 meter instantaneous field-of-view (IFOV) cannot resolve such small features, however, and the spatial changes which have occurred cannot be identified on a Landsat image.

Fortunately, the changes in land cover will also produce spectral changes in the scene which generally will be detectable on the Landsat image. The identification of the detected spectral change as a non-urban to urban land use change, however, may or may not be possible. In general, the detection and identification of the spectral signal produced by a land use/land cover change is a function of several factors such as:

1. The amplitude of the change in radiance produced by the land cover change.
2. The total system noise level limiting the sensitivity to change, often expressed as the noise equivalent change in reflectivity (NE$\Delta\rho$).
3. The size of the scene features which are modulating the radiance and the ability of the system to transfer the modulation.
4. The uniqueness of the spectral changes relative to the identification of the land cover change.

5. The uniqueness of the land cover change relative to the identification of the land use change.

The first two factors determine the detectability of the radiance change resulting from the land use/land cover change. This signal to noise (S/N) analysis assumes that the area wherein the land cover has changed is very large compared to the resolution or IFOV of the system. The third factor considers the effect of the size of the changed area when the size falls in the range of 1 to 10 times the IFOV dimensions. Under these conditions the modulation transfer function of the entire system becomes an important factor in determining the detectability of the change signal. These first three factors are illustrated in Figure 1.2.

At A and D the radiance modulation produced by large and small fields having different land covers than the surrounding region is illustrated. At B and E the demodulated detector output signal is shown. The rounding of the boundaries of the fields are illustrative of MTF effects. The reduced amplitude of the small field signal indicates the field size is approaching the IFOV of the system. The effect of noise for S/N = 1 is illustrated at C and F. Obviously, detection of the small field is not likely.

The fourth and fifth factors noted above have to do with the interpretation of the detected signals from a multispectral scanner such as Landsat. Simply recognizing that a spectral change has taken place is not sufficient to determine that a change from non-urban to urban cover has occurred. A spectral change from one scene to another can be due to differences in atmospheric condition, change of crops, difference in growth stage or management of non-urban land, etc. (Riordan, 1980a,b). To recognize that a change from non-urban to

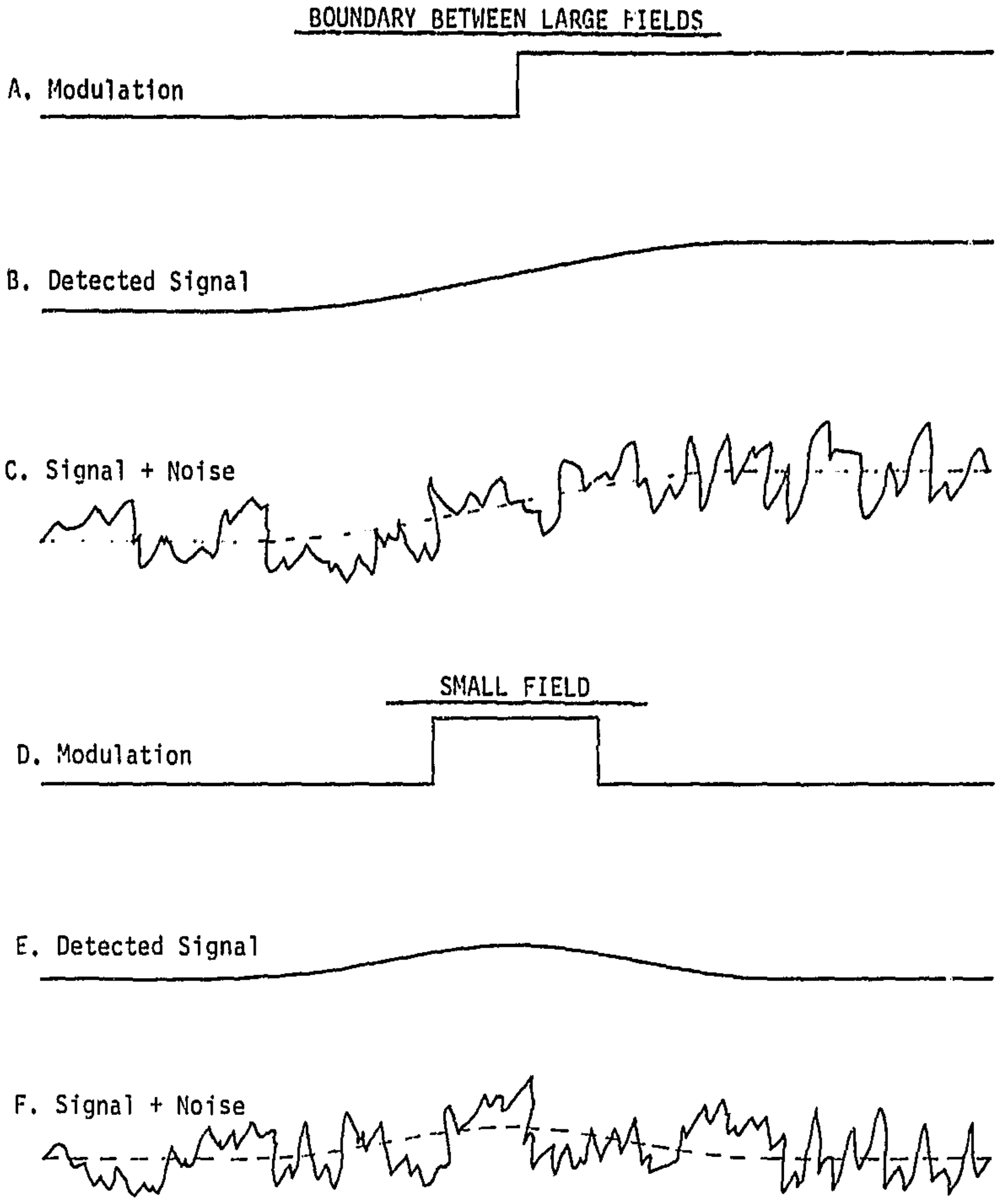


Figure 1.2 Illustrations of the effect of signal, noise and MTF on the detectability of land cover changes for large and small fields.

urban cover has occurred requires the specification of an indicator of urban cover. Attempts at identifying a unique urban fringe (sub-urban) spectral signature have not been successful since this land use is associated with a land cover of varying proportions of vegetation, structures, and urban infrastructures such as roads. Rather, a good indicator of change to urban might be a "construction" signature at the time the land is cleared and ready for building (Riordan, 1980a,b). This means that the change detection algorithm must operate in the temporal and spectral domains since changes in spectral characteristics with time will provide the clue to a land use change. Obviously, the employment of this method requires an understanding of the agricultural practices of the region of interest since most agricultural fields will be barren at one or more times during the year. Given that this knowledge exists, one can then evaluate the limitations of such a scheme by evaluating S/N and MTF system performance factors. This was the objective of the research reported herein.

2.0 TECHNICAL DISCUSSION

The primary effort undertaken on this project was the development of computer models and analytical procedures for evaluating the performance of remote sensing systems and data analysis techniques (especially change detection). A model was needed which would simulate the following:

1. The two-dimensional spatial configuration of a scene as it is observed by a remote sensing system.
2. The modulation of scene irradiance by the changing reflectance of scene components.
3. The transfer of the modulated scene radiance to the image data.
4. The addition of noise to the modulation signal.

It was also desired to have procedures capable of performing the following analyses or calculations:

a. Calculate S/N ratios for changing scene, MTF and noise conditions.
b. Analyze and characterize the modulated radiance from the scene in both the spatial distance and spatial frequency domains.
c. Display the simulated scene data for visual analysis of the effect of different MTFs and noise conditions.

The model developed on this project does not simulate the interactions between the scene and EM radiation nor was any attempt made to simulate atmospheric interactions. Rather, the scene reflectance values, atmospheric radiance and attenuation, and system noise levels are provided as data inputs to the model. These input data must be obtained from other models or from various empirical sources (image data or experimental measurements). The model uses the scene description and data inputs to generate a two-dimensional set of numbers which simulate scene reflectance and/or radiance values before and after the application of modulation transfer functions.

2.1 System Response Functions

The performance of systems which transmit or carry messages in the form of signals can be measured in part by their signal response characteristics. Telephone, radio, teletype, television and remote sensing systems are required to transfer an input signal to an output device or signal in order to achieve their design function. The system response function determines the fidelity with which the transfer is accomplished and provides a measure of system performance. The response function can be expressed in terms of the system capacity to respond to either the signal frequency content or its time/distance variations.

The rate at which a signal changes determines its waveform and its frequency content. Figure 2.1 shows five waveforms and their frequency spectrum. The continuous sinusoid at (a) presents a line spectrum whereas the impulse at (b) has an infinite spectrum of all frequencies. The step function at (c) with a very fast rise time has a broad spectrum decaying slowly in amplitude from a maximum of $f = 0$. The step function with a slow rise time (d), however, shows a more rapid decay in amplitude for the higher frequencies. Finally, the fluctuating signal at (e) has a distribution of frequencies as shown. The reader is cautioned not to confuse the spatial frequency spectrum

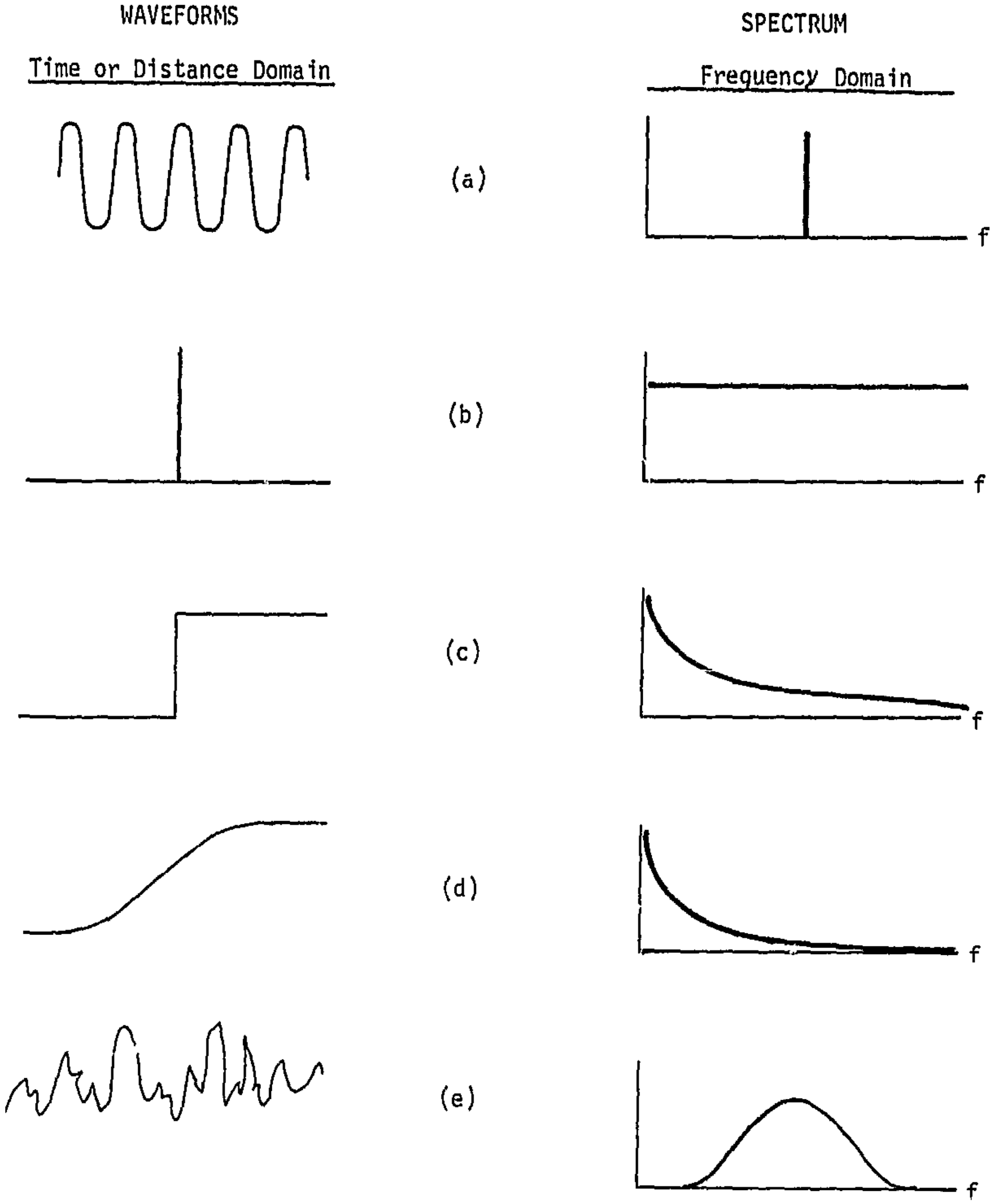


Figure 2.1 Pictorial transform pairs of waveforms and the associated spectrum.

of the modulation from a scene with the EM spectrum of electromagnetic radiation.

The time or distance response of a system to a waveform can only be expressed for idealized shapes such as the impulse or step. This is accomplished by defining the change in wave shape produced by the system. These time or distance functions are typically designated impulse and step responses. For remote sensing systems they are more often referred to as point or edge spread functions. From Figure 2.1 we note that the spreading of an edge or step (compare (c) and (d)) results in a reduction in the high frequency content of the edge. Hence, the equivalence of the point spread function (PSF), edge spread function (ESF), and frequency response curves in Figure 2.2 should be apparent. For remote sensing systems, the frequency response cruves are most often referred to as the modulation transfer function (MTF) which defines the transfer for modulated radiance as a function of the modulation (signal) frequencies.

When analyzing system responses, therefore, we have the choice of using a point spread function or a frequency response function. If either function is available, the other may be obtained via a Fourier transform. In other words, the PSF and MTF form a Fourier transform pair, either of which may be used for system analyses.

The equivalence of operating a model in either the spatial distance or spatial frequency domains allowed us the freedom to select the method easiest to implement for the application at hand. Both domains were used on this project.

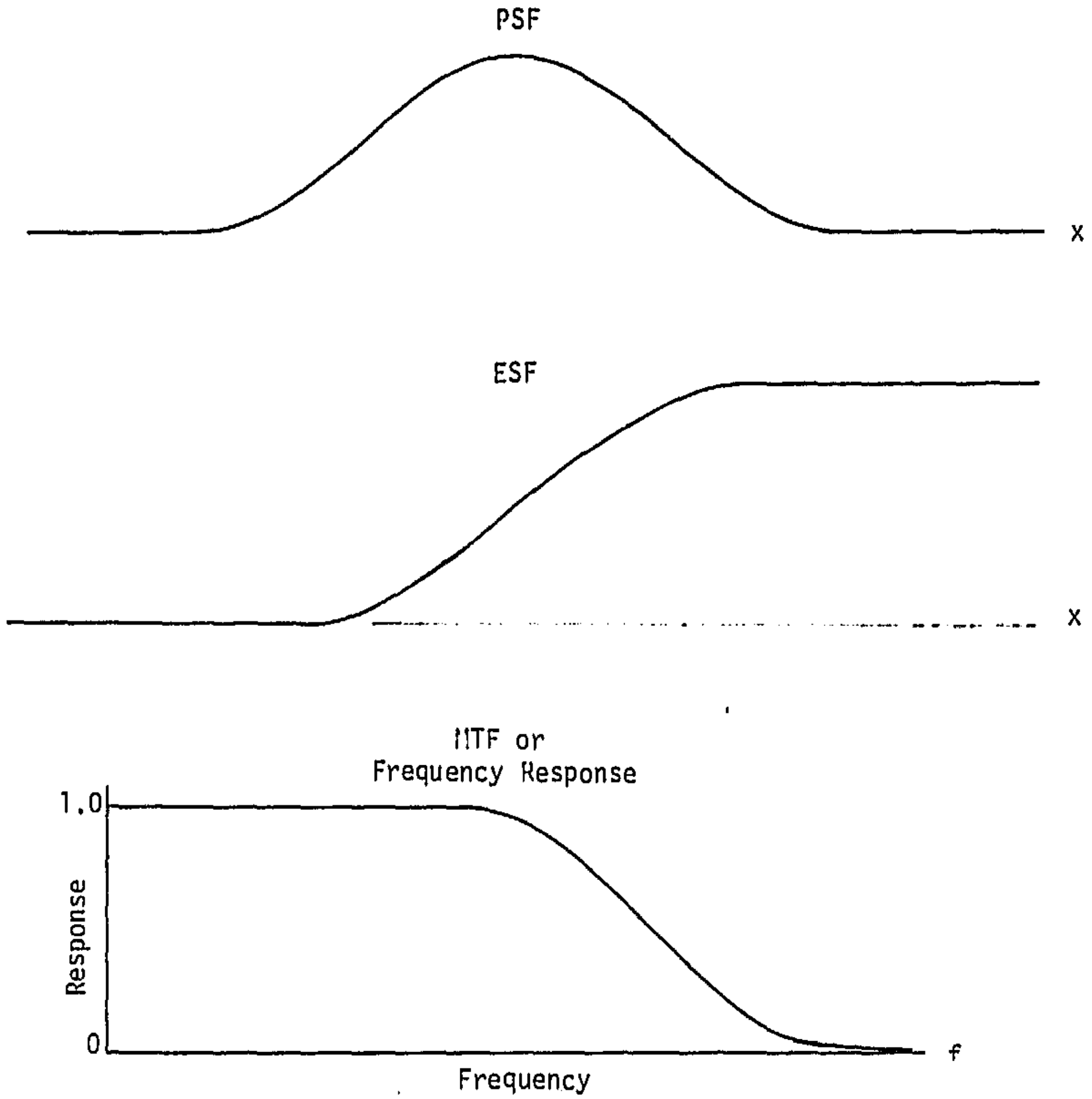


Figure 2.2 A comparison of Edge Spread Function (ESF), Point Spread Function (PSF) and Frequency Response curves for the same remote sensing system.

2.2 The Modulation Transfer Function

The proportion of the modulation that is transferred from the scene to the image at a particular frequency is defined by the modulation transfer factor. The function that describes the change in the transfer factor with a change in frequency is the modulation transfer function (MTF). Figure 2.2 provides a graphical example of an MTF.

To apply an MTF to scene variation (scene modulation), the spatial dimensions of the scene must be converted to the frequency domain using the Fourier transform. The effect of the system MTF on the scene can then be determined by multiplying the scene frequency spectrum (spatial) by the system MTF. This will result in a modified scene spectrum which accurately indicates the loss in certain modualtion components (spatial frequency). This simple procedure cannot be used, however, to regenerate the scene, as modified by the system MTF. In order to accomplish this in the frequency domain we must use the optical transfer function (OTF) which contains phase as well as amplitude terms of the frequency response function. This procedure, if used, requires the transformation of the scene into the frequency domain (retaining phase as well as amplitude information for each frequency term), multiplication of the transformed scene by the system OTF and performance of an inverse transform back to the spatial distance domain to create the modified scene. This is a very complex and expensive process which requires more knowledge about the system response function than is usually available. For these reasons it is more practical to model the effect of the system response on the scene in the spatial distance domain, and then use the frequency domain to evaluate the results.

2.3 The Point Spread Function

The effect of the system MTF on recorded scene readiance values can be easily modeled by convolving the original scene radiance with the system PSF. In effect, the performance of a Fourier transform of an entire scene, followed by the application of frequency domain filtering of the scene, followed by the inverse Fourier transform of the scene has been replaced by the Fourier transform of the MTF (to obtain the PSF) and the application of a spatial (distance) filter (convolution) to the scene.

The PSF is applied in the form of a digital filter that is passed over the scene, which results in the spreading or blurring of the edges of objects and boundaries of fields. The reduction in high frequency modulation can be measured by comparing the spatial frequency spectrum of the scene before and after the application of the PSF. The empirical transfer function obtained in this manner should be equivalent to the theoretical MTF calculated from system design considerations. The PSF, therefore, effects a direct modification of the physical appearance of the scene, equivalent to the actual effect of the remote sensing system.

3.0 DEVELOPMENT OF THE MODEL

An analytical tool is needed to study the effects of sensor and atmosphere MTFs or PSFs on the transfer of ground spatial variation (modulation) to the image. An attempt was made to develop such a tool in the form of a computer model which performs the following functions:

1. Construction of a digital scene representing major ground cover types.
2. Simulation of scene reflectance or radiance values as observed at the scene.
3. Application of the system PSF to the scene, thereby simulating the image of the scene.
4. Analysis and display of the effects of the PSF on the scene digital values (spatial frequency content, etc.).
5. Simulation of the effects of the atmospheric MTF on the scene modulation.

The development of the model is described below; program listings and a user's guide are provided in Appendix A.

3.1 Construction of a Digital Scene (Program SCENE)

The digital scene representing major urban cover types can be constructed by filling a two-dimensional array with object symbols. Several methods for filling the array were considered. The two most common methods are: (1) the digitization of the scene cell by cell; and (2) the digitization of the boundaries of the polygons within which a common cover type exists. The cellular procedure is simple to implement but is labor intensive and could become very expensive for large areas. The digitization of polygons can be accomplished in much less time when object sizes are large compared to the cell size, but this method requires special digitization equipment.

An alternative method requiring no special equipment was developed for this model. This procedure places objects within the scene array by specifying position, shape and size. For example, rectangles are positioned by designating a corner position and the length and width. Circles are positioned by designating their center position and radius. A subroutine within the computer model then computes the cells which fall within the perimeter of the object. All of the cells within the boundaries of designated objects are given the object identification.

This scene construction method is rapid and quite effective when the scene is composed of objects which can be described by position and a regular shape. Irregular objects would have to be defined by a series of contiguous regular shapes. The method works very well for constructing an urban scene of buildings, streets, driveways and trees and bushes (circles of varying size). Program SCENE was written to perform these procedures.

When SCENE is used to build a scene model, the majority ground cover type, such as grass, is first used to fill the entire array. Opaque objects, such as rooftops, are superimposed (using a replacement procedure) on the background cover. Then, translucent objects, such as tree canopies, are added to the scene by combining the object codes for the translucent object and the underlying material. The result is a scene array composed of ground cover types specified by code names.

Once the scene is created, the cover code names are converted to reflectance values for any wavelength using computer text editor commands which are specific to the computer system. That is, each occurrence of a particular code name is replaced by a reflectance value, which may be the reflectance of an opaque material or the combined reflectance of an opaque material overlain with a translucent material.

3.2 Development of the PSF Model

Ideally one would derive or obtain the MTF for a remote sensing system and then perform an inverse Fourier transform to obtain the PSF. In this instance, neither the function nor sufficient information on the system design were available. Instead, a plot of the Landsat MTF was obtained from Norwood (1974). This plot was normalized relative to the cutoff frequency (that frequency at which the system response is one-half the maximum response) and replotted as shown in Figure 3.1. This normalized response curve for the Landsat MTF was compared to several filter response curves from Watt, et al. (1953). One of these curves is also plotted on Figure 3.1 and is seen to be a close match to the Landsat MTF. Watt defines this as a cosine squared through rectangular aperture filter function.

Because of the close match between the Landsat MTF and the filter response function shown in Figure 3.1, it was assumed that the impulse and step function response curves calculated by Watt for the filter can also be used to represent Landsat's PSF and ESF. This was verified, to the extent possible, by a comparison with small scale plots of Landsat's PSF taken from Dye (1975). Although the plots in Dye's paper were too small for exact comparison, it was obvious there were

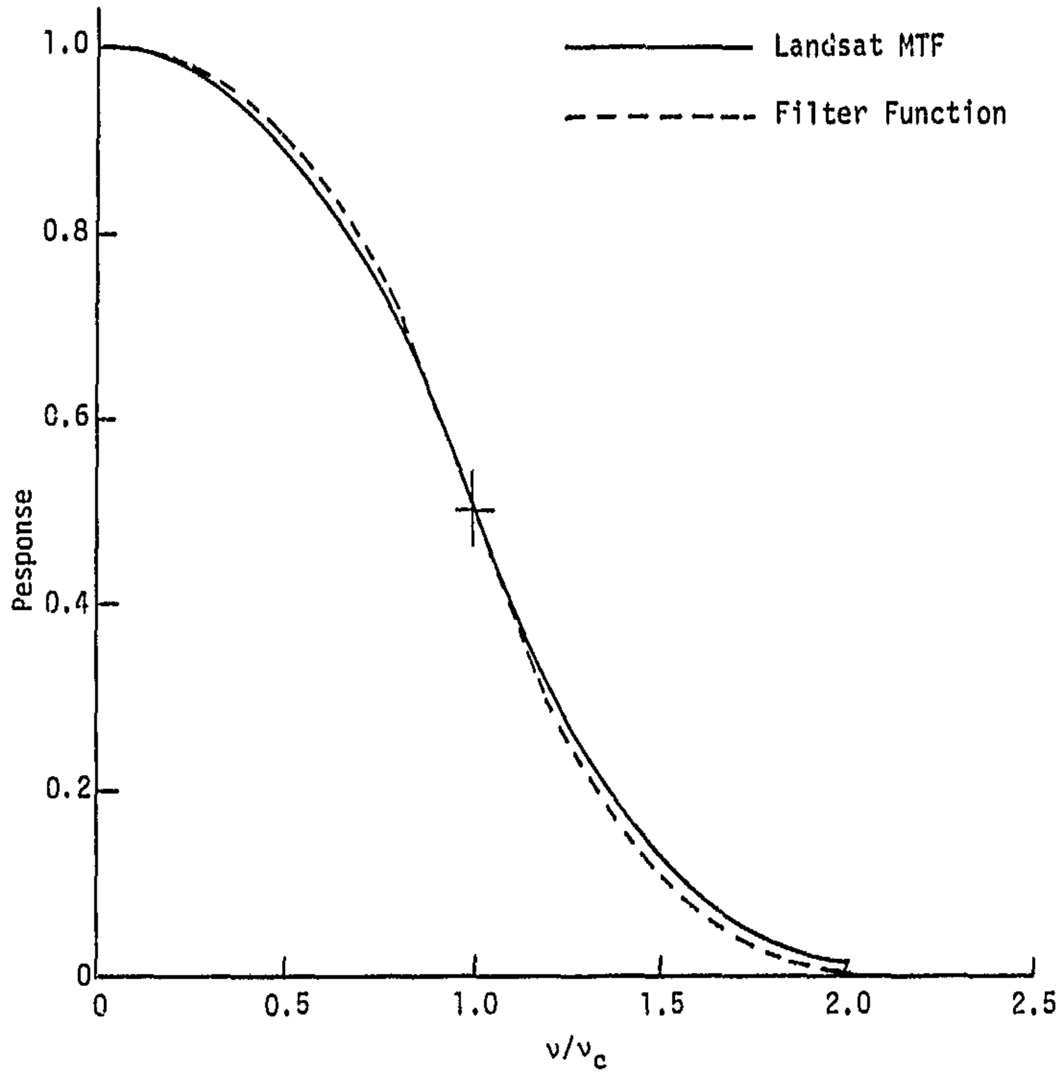


Figure 3.1 Normalized response functions for Landsat (from Norwood, 1974) and a cosine squared through a rectangular aperture filter (from Watt, et al, 1953). The cutoff frequency, ν_c, for the Landsat MTF is 0.006835 cycles/meter.

no significant differences between the results of his calculations and the curves from Watt, when both were normalized to the cutoff frequency. Hence, the PSF curve given in Figure 3.2 was used for the model of a MSS system. For Landsat, ν_c = 0.006835 cycles/meter.

3.2.1 Application of the PSF to the Scene Image

The point spread function defines the spatial distribution of a point source of light (EM radiation) within the image. This distribution, when projected back to the scene, defines the apparent source as observed on the image. If the PSF is convolved with the scene (point by point) a simulated image of the scene, as would be produced by the MSS system, is produced.

The PSF curve in Figure 3.2 is, of course, a one-dimensional representation of a two-dimensional phenomenon. If the PSF in each dimension is independent of the other, the two-dimensional PSF can be obtained by multiplying the two PSF functions to obtain off-axis values. In this manner, a two-dimensional array of values are obtained which are used as weights for a two-dimensional digital filter which is moved across the scene. Figure 3.3 shows an array of filter weights for Landsat (assuming a symmetrical PSF based on Figure 3.2) for use with a scene model made up of 7 meter cells. The dotted lines show the outline of a Landsat pixel (about 76 X 76 meters). The influences of surrounding terrain on Landsat digital numbers is apparent.

The filter array illustrated in Figure 3.3 does not include the negative values at distances beyond ½ ν_c because of the cost. Including the values out to a distance of $1/\nu_c$ quadruples the size of the filter array. This cannot be justified under most conditions. The computer

ORIGINAL PAGE IS
OF POOR QUALITY

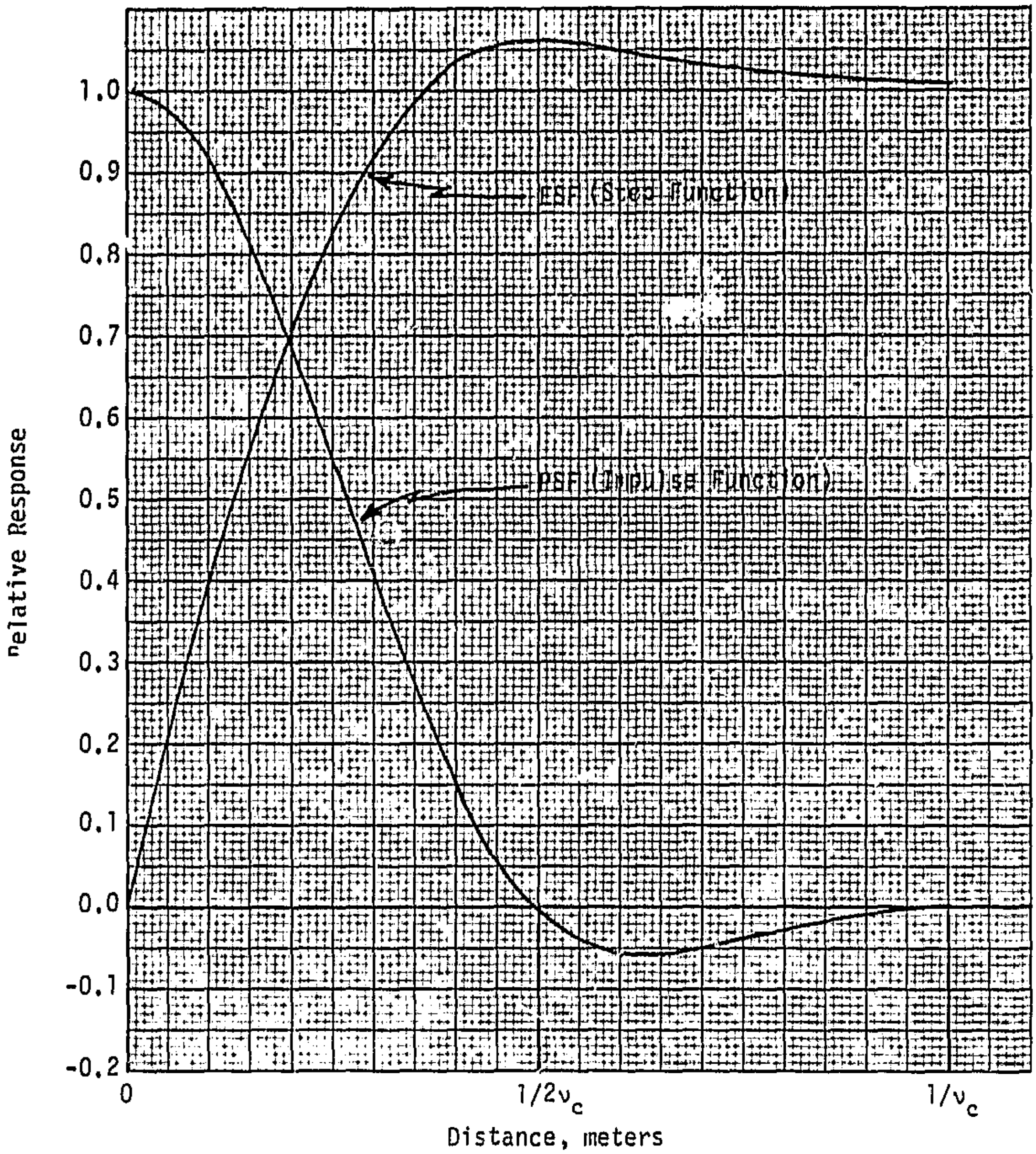


Figure 3.2 Impulse and step response functions for an idealized filter which closely resemble the Point Spread Function (PSF) and Edge Spread Function (ESF) of the Landsat MSS system. For Landsat the cutoff frequency, ν_c, is equal to 0.006835 cycles/meter. (From Watt, et al, 1953.)

ORIGINAL PAGE IS
OF POOR QUALITY

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.01	.02	.02	.03	.04	.05	.05	.06	.06	.06	.05	.05	.04	.03	.02	.02	.01	.00	.00
.00	.01	.02	.04	.06	.08	.10	.12	.14	.15	.15	.15	.14	.12	.10	.08	.06	.04	.02	.01	.00
.00	.02	.04	.07	.10	.15	.19	.22	.25	.26	.27	.26	.25	.22	.19	.15	.10	.07	.04	.02	.00
.00	.02	.06	.10	.14	.20	.26	.31	.34	.36	.37	.36	.34	.31	.26	.20	.14	.10	.06	.02	.00
.00	.03	.08	.15	.20	.29	.37	.45	.49	.52	.54	.52	.49	.45	.37	.29	.20	.15	.08	.03	.00
.00	.04	.10	.19	.26	.37	.48	.57	.63	.67	.69	.67	.63	.57	.49	.37	.26	.19	.10	.04	.00
.00	.05	.12	.22	.31	.45	.57	.69	.76	.81	.83	.81	.76	.69	.57	.45	.31	.22	.12	.05	.00
.00	.05	.14	.25	.34	.49	.63	.76	.83	.88	.91	.88	.83	.76	.63	.49	.34	.25	.14	.05	.00
.00	.06	.15	.26	.36	.52	.67	.81	.88	.94	.97	.94	.88	.81	.67	.52	.36	.26	.15	.06	.00
.00	.06	.15	.27	.37	.54	.69	.83	.91	.97	1.0	.97	.91	.83	.69	.54	.37	.27	.15	.06	.00
.00	.06	.15	.26	.36	.52	.67	.81	.88	.94	.97	.94	.88	.81	.67	.52	.36	.26	.15	.06	.00
.00	.05	.14	.25	.34	.49	.63	.76	.83	.88	.91	.88	.83	.76	.63	.49	.34	.25	.14	.05	.00
.00	.05	.12	.22	.31	.45	.57	.69	.76	.81	.83	.81	.76	.69	.57	.45	.31	.22	.12	.05	.00
.00	.04	.10	.19	.26	.37	.48	.57	.63	.67	.69	.67	.63	.57	.48	.37	.26	.19	.10	.04	.00
.00	.03	.08	.15	.20	.29	.37	.45	.49	.52	.54	.52	.49	.45	.37	.29	.20	.15	.08	.03	.00
.00	.02	.06	.10	.14	.20	.26	.31	.34	.36	.37	.36	.34	.31	.26	.20	.14	.10	.06	.02	.00
.00	.02	.04	.07	.10	.15	.19	.22	.25	.26	.27	.26	.25	.22	.19	.15	.10	.07	.04	.02	.00
.00	.01	.02	.04	.06	.08	.10	.12	.14	.15	.15	.15	.14	.12	.10	.08	.06	.04	.02	.01	.00
.00	.00	.01	.02	.02	.03	.04	.05	.05	.06	.06	.06	.05	.05	.04	.03	.02	.02	.01	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

Figure 3.3 Point spread function values for the Landsat MSS system for a scene with 7 meter cells. The dotted lines show the outline of a Landsat pixel.

program which calculates the values (weights) for the two-dimensional filter array is called MAKWTS.

The array of weights generated by MAKWTS is input to program PIXPRO which creates pixel values based on application of the filter array (PSF) to the cellular scene model. Although a symmetrical PSF was assumed, the program can be modified to use different PSFs in the two dimensions.

3.3 Modeling the Effects of the Atmosphere and Random Noise on Modulation Signal/Noise (S/N)

The atmosphere affects scene modulation and S/N in several ways. The addition of path radiance (PR) to radiance leaving the scene reduces modulation by increasing the mean level of radiance leaving the scene. To illustrate:

$$(1)\quad \text{Modulation} = \frac{I(max) - I(min)}{I(max) + I(min)}$$

If path radiance is added to I(min) and I(max), the equation becomes

$$(2)\quad \text{Modulation} = \frac{(I(max) + PR) - (I(min) + PR)}{(I(max) + PR) + (I(min) + PR)}$$

Taking a numerical example with I(max) = 6, I(min) = 3, and PR = 1:

$$M(1) = \frac{6 - 3}{6 + 3} = .33 \qquad M(2) = \frac{(6 + 1) - (3 + 1)}{(6 + 1) + (3 + 1)} = .27$$

Therefore, the modulation has been reduced by the addition of path radiance across the scene. This also results in a reduction in contrast (Imax/Imin) since the initial contrast if 6/3 = 2 and the final contrast os 7/4 = 1.75. However, the signal, or absolute difference between the radiance values for the cover types, remains the same (6 - 3 = 3; 7 - 4 = 3).

On the other hand, atmospheric attenuation of scene radiance results in a reduction of signal, but no reduction in modulation or contrast. For example, adding atmospheric transmittance to (1) yields,

$$(3)\quad \text{Modulation} = \frac{T(I(max) - I(min))}{T(I(max) + I(min))}$$

which results in no reduction in modulation. Similarly, atmospheric attenuation produces no change in image contrast. The signal, however, will be reduced from (6 - 3) = 3 to (4.2 - 2.1) = 2.1 which in turn affects S/N.

Different atmospheric conditions will also introduce variability or noise into the absolute radiance values detected by the MSS system, thereby affecting the S/N ratio and our ability to detect change. Atmospheric and other noise sources can be modeled by using the computer system random number generator to generate random numbers within specific bounds and adding the output numbers to the scene sample values.

3.4 Analysis and Display of the Effects of PSF on Scene Modulation

The output pixels from program PIXPRO can be repeated in one dimension using program REPEAT (to match the scale of the input scene) for use in calculation of a one-dimensional transfer function between the input scene and output pixels. The transfer function indicates the amount of variability (modulation) that is transferred from the input scene to output pixels at specific frequencies.

The input scene and output pixels can be displayed using various techniques chosen by the user. Choice of the display depends on the analysis tools available to the user such as classification techniques with theme displays, graymapping, texture analysis and display, etc. The scene and two-dimensional pixel data must first be converted to the user's system compatible format.

4.0 APPLICATION OF THE MODEL TO A SCENE-AN EXAMPLE

This section illustrates the application of the model. An engineering blueprint of a suburban development was used to model a scene. Objects in the scene were positioned relative to the top left corner of the blueprint. The objects recorded were houses, driveways, streets, sidewalks, and trees. Tree locations were specified by coordinates of circle centers. Dimensions of length, width, and radius were specified for each object. Objects that were irregular in shape or at angles to the left and top of the blueprint were recorded in rectangular sections parallel to the blueprint edges.

4.1 Creation of the Scene

The filling of the scene array was accomplished using program SCENE. The scene array was first filled with a background cover of grass. Opaque objects (houses, sidewalks, streets, and driveways) were superimposed on the grass background. Translucent objects (trees) were then added to the scene by combining the code names for the translucent and underlying cover types.

Using the text editor supported by the computer center, the object codes were transformed to reflectance values. That is, all occurrences of a particular code name were changed to reflectance values for a particular wavelength and saved in a new file. The file containing the original code names was retained for conversion to different wavelength reflectance values at another time. Reflectance values for the cover types were taken from multispectral aircraft data obtained by Root and Miller (1971). Characteristics of the resulting scene are given in Table 4.1. A portion of the scene is shown in Figure 4.1.

Table 4.1 Characteristics of the original scene

Cover Type	Cover Code Name	Reflectance* Value (Band 5)
Grass	GRASS	10.0
Rooftop (shingles)	HOUSE	22.5
Sidewalk	SWALK	43.0
Driveway	DRWAY	39.0
Street (asphalt)	STRET	15.0
Grass/Tree	GRATR	6.5

*Band 5 reflectance values taken from Root and Miller (1971).

4.2 Creation of Sample Weights Representing the PSF

Sensor MTF cutoff frequencies were input to program MAKWTS to create a matrix of weights for input to program PIXPRO. Four cutoff frequencies were used: 0.006835, 0.01667, 0.05, and 0.1 cycles/meter. These frequencies correspond to pixel sizes of about 76 (Landsat), 30, 10, and 5 meters and assume an increase in optics diameter proportionate to the reduction in pixel size. Due to the cell size chosen (NCELSZ), each sample in created scene represented 1.905 meters (6.25 feet), therefore, a different number of weights was output depending on the cutoff frequency and number of digitized PSF values. The output weights for the four pixel sizes are given in Table 4.2.

4.3 Application of the PSF to the Scene

The weights output from program MAKWTS were applied to the digital scene using program PIXPRO. A two-dimensional, symmetrical PSF

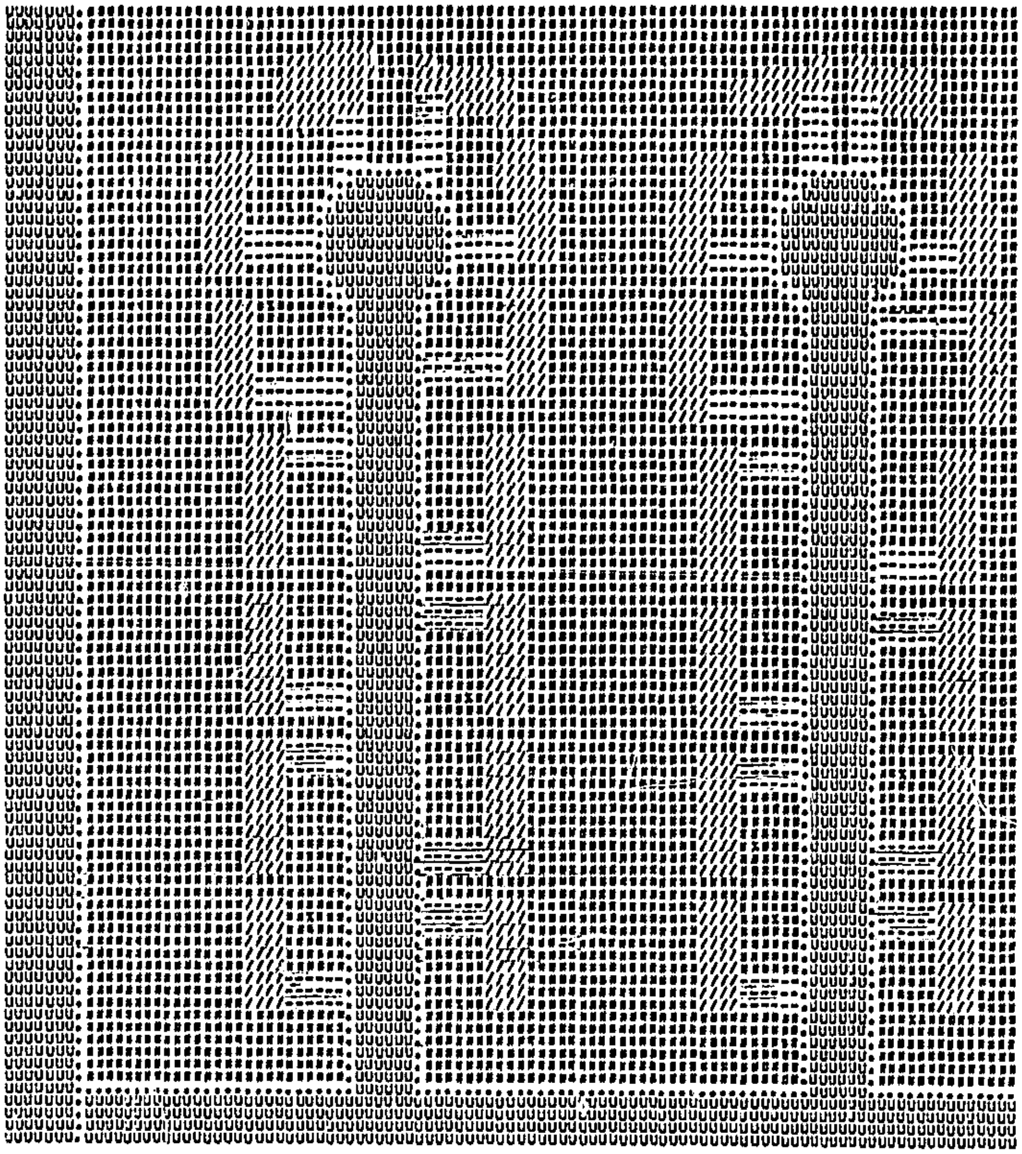


Figure 4.1 Portion of the original scene.

KEY: U = street
. = sidewalk
/ = house
= grass
% = tree/grass
- = driveway

Table 4.2 Table of output weights.

0	1.000
1	.9900
2	.9800
3	.9700
4	.9600
5	.9500
6	.9300
7	.9100
8	.9000
9	.8800
10	.8500
11	.8200
12	.7900
13	.7500
14	.7100
15	.6900
16	.6400
17	.6000
18	.5700
19	.5150
20	.4800
21	.4500
22	.4100
23	.3700
24	.3500
25	.3200
26	.2800
27	.2400
28	.2200
29	.1800
30	.1500
31	.1200
32	.1000
33	.0800
34	.0600
35	.0500
36	.0250
37	.0100
38	0.0000

****

Resolution 1

Pixel Size = 76 X 76 meters

Table 4.2 Table of output weights
(Continued)

OUTWTS2

0	.9950	
1	.0767	
2	.9500	
3	.9050	
4	.8467	
5	.7767	
6	.6833	Resolution 2
7	.6000	
8	.5033	Pixel Size = 30 X 30 meters
9	.4100	
10	.3200	
11	.2433	
12	.1700	
13	.1033	
14	.0600	
15	.0150	

****

OUTWTS3

0	.9840	
1	.9010	Resolution 3
2	.6867	
3	.4110	Pixel Size = 10 X 10 meters
4	.1722	
5	.0252	

****

OUTWTS4

0	.9600	Resolution 4
1	.6784	
2	.1847	Pixel Size = 5 X 5 meters

was assumed, therefore, the weights on the vertical and horizontal axes were the same and were multiplied together to obtain the off-axes weights. Figure 4.2 shows the weighting scheme for 10 m pixels. Application of the weighting function is equivalent to convolving the PSF with the scene.

Quadrant 1 (same as 3)	.0227	.0252	.0227				.0006
		.1722					
		.4110		Quadrant 2 (same as 3)			
		.6867					
		.9010	.8112				.0227
	.9010	.9840	.9010	.6867	.4110	.1722	.0252
		.9010	.8112	.6187	.3703	.1552	.0227
Quadrant 4 (same as 3)		.6867	.6187	.4716	.2822	.1182	.0173
		.4110	.3703	.2822	.1689	.0708	.0104
		.1722	.1552	.1182	.0708	.0297	.0043
	.0227	.0252	.0227	.0173	.0104	.0043	[illegible]

Quadrant 3

Figure 4.2 PSF weights for an MSS system having 10m X 10m pixels.

4.4 Analysis and Display of the Effects of the PSF on Scene Modulation

The pixels output from program PIXPRO were repeated in one dimension using program REPEAT to match the scale of the original scene used to create the pixels. One-dimensional transects corresponding to the pixels were extracted from the original scene using program TRN. The repeated pixels and transects of samples from the original scene were combined using program BMDDAT. Finally, the combined data set from program BMDDAT was input to program BMDO2T to calculate the transfer function between the input scene and output pixels. Values of the transfer function for the four resolutions are shown in Table 4.3 for Band 5 of Landsat.

As expected, Table 4.3 indicates that a greater amount of variability (modulation) in the original scene is transferred to the output pixels as the size of the pixels decreases. With a pixel size of 76 X 76 meters, very little modulation from the original scene is transferred. As the pixel size is reduced to 5 meters, a greater proportion of the modulation is transferred at most frequencies as indicated by the transfer functions.

The minimum and maximum pixel values were determined for each resolution and the ranges for graymapping were arbitrarily set. Program GRARHO was used to calculate the graylevels for each pixel size. The scenes were then graymapped using program GRAPIX to examine the relationship between input scene and output pixels. These graymaps are displayed in Figures 4.3 through 4.6.

Although not demonstrated on this project, both the input scene and output pixels could be classified using different spectral bands and various classification techniques. These techniques could include

Table 4.3 Modulation transfer functions for 4 resolutions.

Resolution 1
76m X 76m
156 points
39 lags

Frequency (cycles/pixel)	Amplitude of the Transfer Function
0.000	.0330?43
.013	.0105166
.026	.0045020
.038	.0017697
.051	.0022?39
.064	.0021350
.077	.0037132
.090	.0012212
.103	.0032877
.115	.0011141
.128	.0027145
.141	.0012626
.154	.0034703
.167	.0005907
.179	.0031931
.192	.0012144
.205	.0034843
.218	.0039307
.231	.0015584
.244	.0002264
.256	.0013098
.269	.0003598
.282	.0013346
.295	.0004673
.308	.0015539
.321	.0009160
.333	.0038155
.346	.0030933
.359	.0007090
.372	.0018038
.385	.0004?47
.397	.0001315
.410	.0001651
.423	.0001170
.436	.0005366
.449	.0002460
.462	.0011242
.474	.0005840
.487	.0032676
.500	.0051613

Resolutions 2, 3 & 4
1000 points
100 lags

Frequency (cycles/pixel)	Amplitude of the Transfer Function		
	Resol. 2 30m X 30m	Resol. 3 10m X 10m	Resol. 4 5m X 5m
0.0000	.2444997	.0258799	.6001205
.0050	.1301439	.1248038	.6788301
.0100	.0753116	.5659204	.6336029
.0150	.1909857	.7248179	.8221831
.0200	.4013660	.7458319	.8078368
.0250	.4188265	.7760909	.7963374
.0300	.2729806	.7563678	.7771642
.0350	.1388116	.5966475	.7140969
.0400	.0919812	.4572441	.7123438
.0450	.0409582	.4182064	.6869173
.0500	.0402746	.3671457	.6704827
.0550	.0143148	.2935824	.6988187
.0600	.0126627	.2896581	.7480232
.0650	.0055750	.2456244	.7289461
.0700	.0115779	.1660098	.7069424
.0750	.0178336	.1054419	.6830869
.0800	.0751746	.1883449	.6112927
.0850	.1370998	.3822733	.5585724
.0900	.0879517	.3535166	.5252366
.0950	.0754691	.2877341	.5256939
.1000	.1583321	.1739534	.5307847
.1050	.0303297	.2351722	.5930438
.1100	.0213444	.2165054	.5981739
.1150	.0066860	.2049474	.5825626
.1200	.0089768	.1790724	.5264897
.1250	.0085285	.0078266	.5011310
.1300	.0070957	.0681967	.4305233
.1350	.0078297	.0709659	.4130954
.1400	.0[illegible]0767	.0530040	.3905687
.1450	.0398989	.0163427	.4002331
.1500	.0688430	.0675263	.4061560
.1550	.0202177	.1957072	.4066351
.1600	.0431757	.1838842	.3884731
.1650	.1228051	.1736466	.3403358
.1700	.0623361	.3192247	.3955765
.1750	.0088729	.3125815	.4348051
.1800	.0073686	.2712772	.3864189
.1850	.0060269	.2354102	.3101109
.1900	.0088422	.0932060	.1529174
.1950	.0081741	.0208241	.1739589
.2000	.0152022	.0103982	.2014375
.2050	.0048958	.0043505	.2063086
.2100	.0928304	.1812194	.2187629
.2150	.0500045	.1478825	.2163006
.2200	.0277342	.1735144	.1853561
.2250	.1044047	.3454596	.2282688
.2300	.1176887	.3420888	.3120373
.2350	.0242443	.2439715	.1897628
.2400	.0010665	.0534401	.1691717
.2450	.0013780	.0755796	.1818238
.2500	.0061761	.0530675	.1700879
.2550	.0034441	.0523169	.2134215
.2600	.0098600	.0644162	.1351318
.2650	.0154731	.0734089	.1555455
.2700	.0111995	.0614539	.1568385
.2750	.1165276	.0255941	.1805923
.2800	.3791855	.0575549	.2366319
.2850	.0907329	.1193681	.1602109
.2900	.0962921	.1024241	.2420289
.2950	.0793553	.0538372	.2334235
.3000	.0196210	.0225239	.0913479
.3050	.0095473	.0406649	.1547468
.3100	.0045485	.1009288	.1869260
.3150	.0063414	.1406624	.1609737
.3200	.0136567	.0254222	.1077638
.3250	.0545414	.0173146	.0125082
.3300	.0455769	.0471984	.0039878
.3350	.1562211	.0287368	.0020150
.3400	.1453995	.1038207	.0260079
.3450	.0891815	.0855986	.1426229
.3500	.1936176	.2103380	.1074578
.3550	.1393456	.3644858	.0597407
.3600	.0134916	.3763153	.1695256
.3650	.0303739	.2223122	.1748200
.3700	.0064908	.2910376	.2132293
.3750	.0026158	.2038967	.2591442
.3800	.0013145	.1892134	.2275890
.3850	.0140219	.1267664	.1189998
.3900	.0666926	.1178995	.0504077
.3950	.1203978	.0026064	.1340238
.4000	.1034563	.0007810	.1173861
.4050	.0266344	.0378287	.0727278
.4100	.1259280	.0459293	.1719699
.4150	.0955534	.0765770	.2118968
.4200	.0184799	.1034269	.2216473
.4250	.0099881	.1612386	.2080189
.4300	.0077049	.1977631	.1793529
.4350	.0059481	.1757221	.[illegible]79814
.4400	.0044422	.1373572	.3548907
.4450	.0017980	.1313837	.33[illegible]471
.4500	.0034361	.1283999	.2067196
.4550	.0639002	.0507067	.0769241
.4600	.1482316	.0737368	.[illegible]513[illegible]62
.4650	.0311975	.0569166	.1069320
.4700	.0281334	.0464363	.1029126
.4750	.1087243	.0111716	.0638413
.4800	.1240311	.0459374	.1767788
.4850	.0107751	.1980443	.2509043
.4900	.0088622	.2247773	.1953132
.4950	.0096968	.1058064	.1677316
.5000	.0033109	.1142626	.2026702

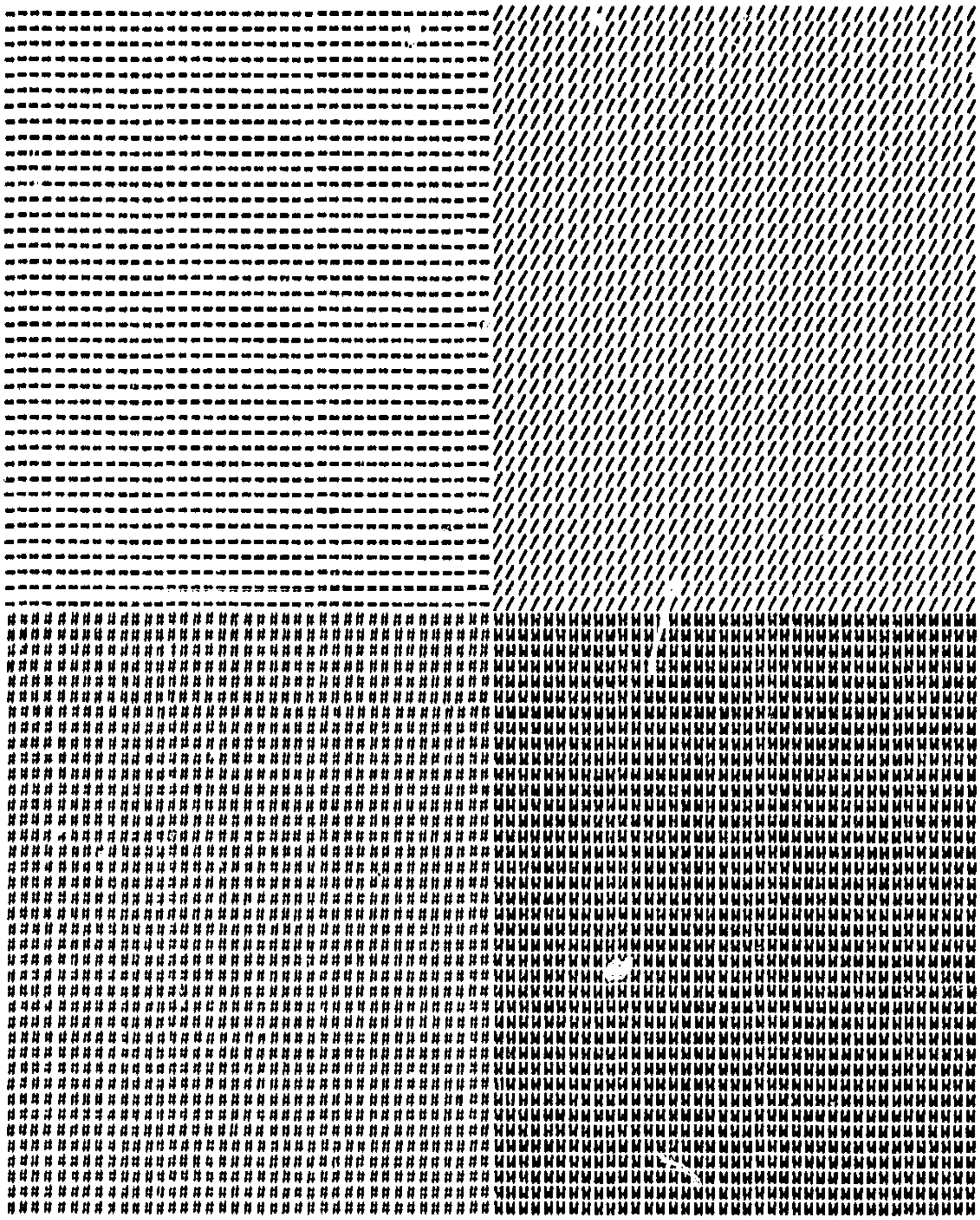


Figure 4.3 Portion of Graymap for Resolution 1, pixel size approximately 76m X 76m.

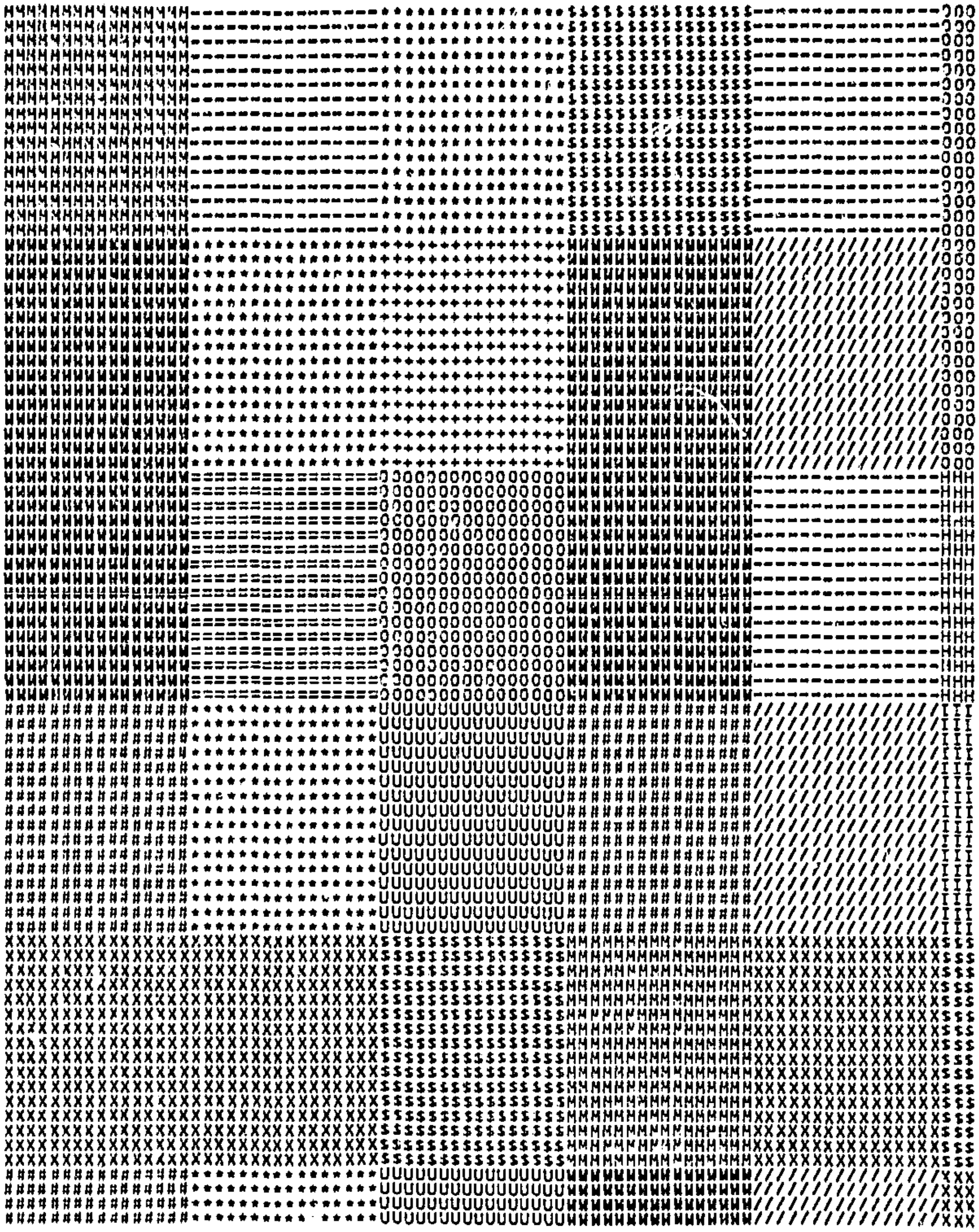


Figure 4.4 Portion of Graymap for Resolution 2, pixel size approximately 30m X 30m.

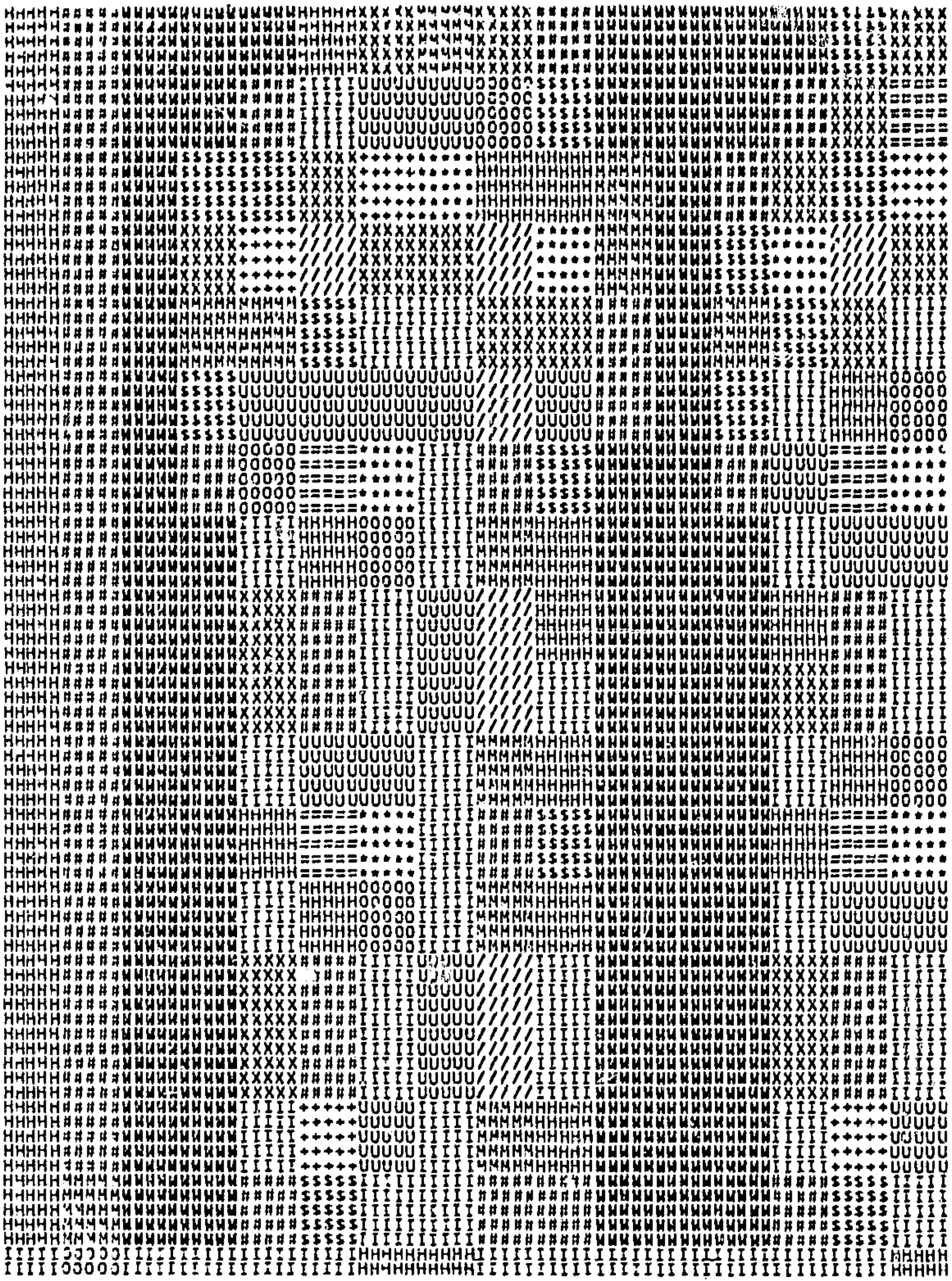


Figure 4.5 Portion of Graymap for Resolution 3, pixel size approximately 10m X 10m.

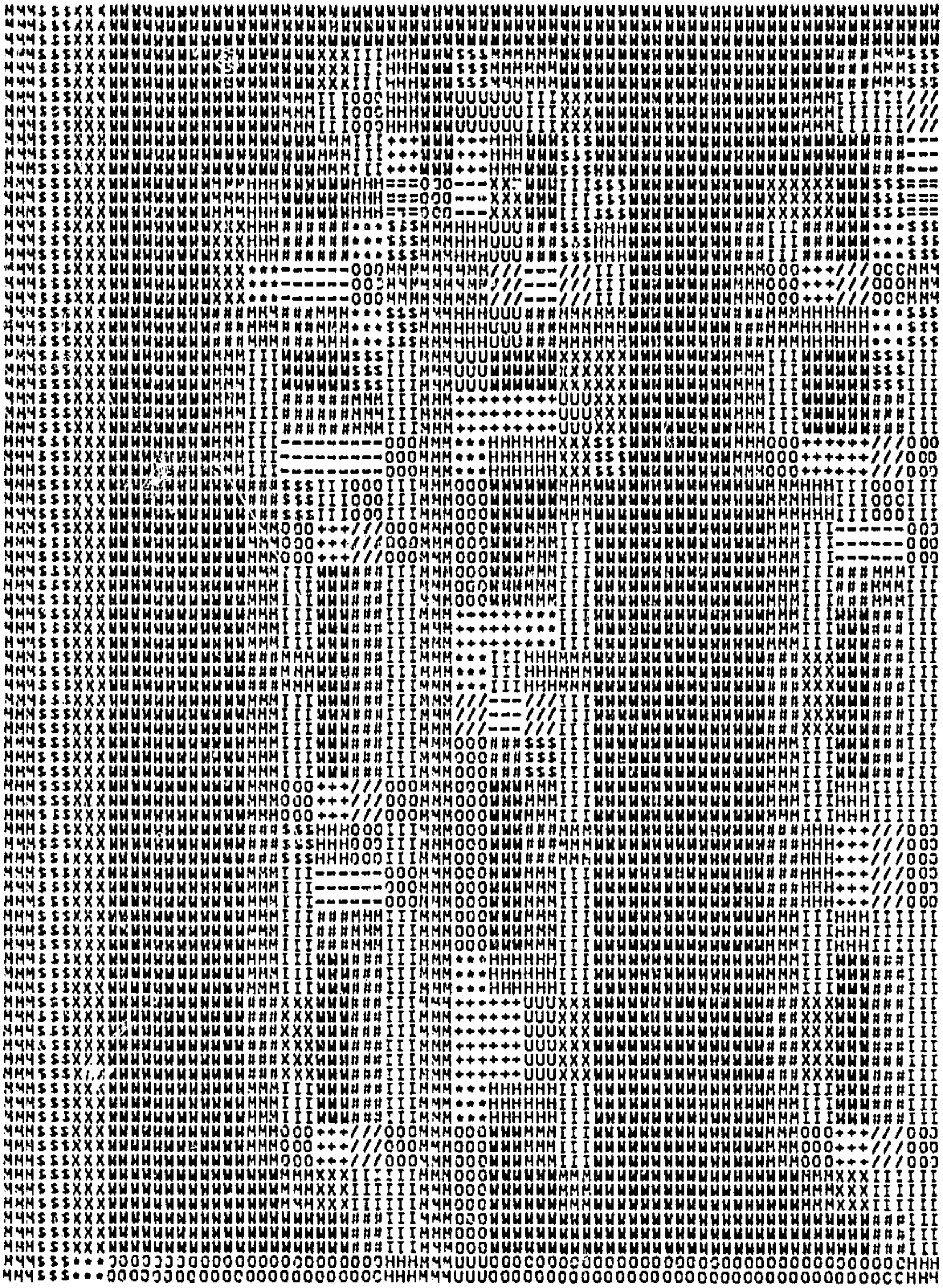


Figure 4.6 Portion of Graymap for Resolution 4, pixel size approximately 5m X 5m.

clustering, supervised classification, texture analysis, etc. The results of these classifications could then be used to evaluate the potential performance of remote sensing systems having different resolution (pixel size) characteristics.

The purpose of the above exercise is to illustrate the input and output levels of reflectance (which depend on the MTF and PSF of the sensor). The reflectance values should be converted to radiance values and then quantitized using the quantizaton characteristics of the particular sensor being analyzed. The user will then be able to examine the actual digital values that can be expected from the sensor.

4.5 Analysis of Atmospheric Effects and Scene Noise

Atmospheric effects can be analyzed by attenuating the radiance values and adding atmospheric path radiance using program ATMOS. The transformation of the radiance values using this program can precede the calculation of the transfer function and the classification/graymap steps. Land cover signature differences due to different atmospheric conditions can be examined by applying different atmospheric effects to the same scene and comparing classifications. This same technique can be used to examine the effect of random noise on the signatures by adding/subtracting randomly generated numbers to the pixel radiance values or to the original scene.

5.0 REFERENCES

Brock, G. C. 1970. Image Evaluation for Aerial Photography. The Focal Press, New York, 258 p.

Christenson, J. S., B. J. David, V. J. Gregg, H. M. Lachowski, and R. L. McKinney. 1977. Landsat urban area delineation. NASA/Goddard Space Flight Center, Info. Trans. Lab., Intralab Project 75-3.

Christenson, J. W., D. L. Dietrich, H. M. Lachowski, M. L. Stauffer, and R. L. McKinney. 1978. Urbanized area analysis using Landsat data. Presented at the Harvard Computer Laboratory's International User's Conference on Computer Mapping, July 23-28, 1978, Cambridge, Mass.

Christenson, J. W. and H. M. Lachowski. 1977. Urban area delineation and detection of change along the urban-rural boundary as derived from Landsat digitized data. NASA/Goddard Space Flight Center, Greenbelt, MD, X-923-77-245.

Computer Sciences Corporation. 1978. Landsat image differencing as an automated land cover change detection technique (Interim Report), prepared for NASA/Goddard Space Flight Center, Greenbelt, MD, Contract No. NAS5-24350.

Dye, R. H. 1975. Restoration of Landsat images by discrete two-dimensional deconvolutions. Proc. 10th Intl. Symp. on Remote Sensing of Environment, ERIM, Ann Arbor, MI, October, 1975, pp. 725-735.

Friedman, S. Z. 1980. Mapping urbanized area expansion through digital image processing of Landsat and conventional data, Jet Propulsion Lab., Pasadena, CA, JPL Publ. 79-113.

Norwood, Virginia T. 1974. Balance between resolution and signal-to-noise ratio in scanner design for earth resources systems. Proc. of SPIE, Vol. 51, Aug. 14-20, 1974, San Diego, CA, pp. 37-42.

Riordan, C. J. 1980a. Non-urban to urban land cover change detection using Landsat data. Summary Report for NASA/Goddard Space Flight Center, Greenbelt, MD, under Contract No. NAS5-26127.

Riordan, C. J. 1980b. Signal/noise and MTF analysis of land cover change detection using Landsat data. Unpublished dissertation, Colorado State University, Fort Collins, CO.

Root, R. R. and L. D. Miller. 1971. Identification of urban watershed units using remote multispectral sensing. Environmental Resources Center, Completion Report Series No. 29, Colorado State University, Fort Collins, CO.

Watt, A. D., R. M. Coon, and V. J. Zurick. 1953. Reduction of adjacent channel interference by shaping of transmitter keying waveform, NBS Rept. No. 2415, Central Radio Propagation Lab., Boulder, Colorado.

APPENDIX A

USEPS' GUIDE AND PROGRAM LISTINGS

USERS' GUIDE

Programs included here are:

SCENE,MAKWTS,PIXPRO,REPEAT,SUBSET,TRN,BMDDAT,BMD02T,GRARHO,GRAPIX,ATMOS

Briefly, program SCENE creates a digital scene; program MAKWTS computes filter weights based on the sensor Point Spread Function to apply to the digital scene samples; program PIXPRO applies the weights to the digital scene and aggregates scene samples into pixels; program REPEAT takes the output pixels and repeats the values to match the scale of the input scene; program SUBSET creates a subset of the original scene data, to match the area covered by the output pixels; program TRN takes transects from the subscene to use in comparing the spatial frequency content with the pixels; program BMDDAT combines the repeated pixels from program REPEAT and transect data from program TRN into one data set for input to the BMD02T power spectral and transfer function program; program BMD02T computes the power spectra of the original scene and output scene and computes the transfer function between the two; program GRARHO calculates graylevels for graymapping the pixels; program GRAPIX graymaps the pixels output from program PIXPRO using graylevels output from program GRARHO; and program ATMOS adds the effects of atmospheric transmittance and path radiance to the output pixel values. A flow chart of the above procedure is given in Figure A-1.

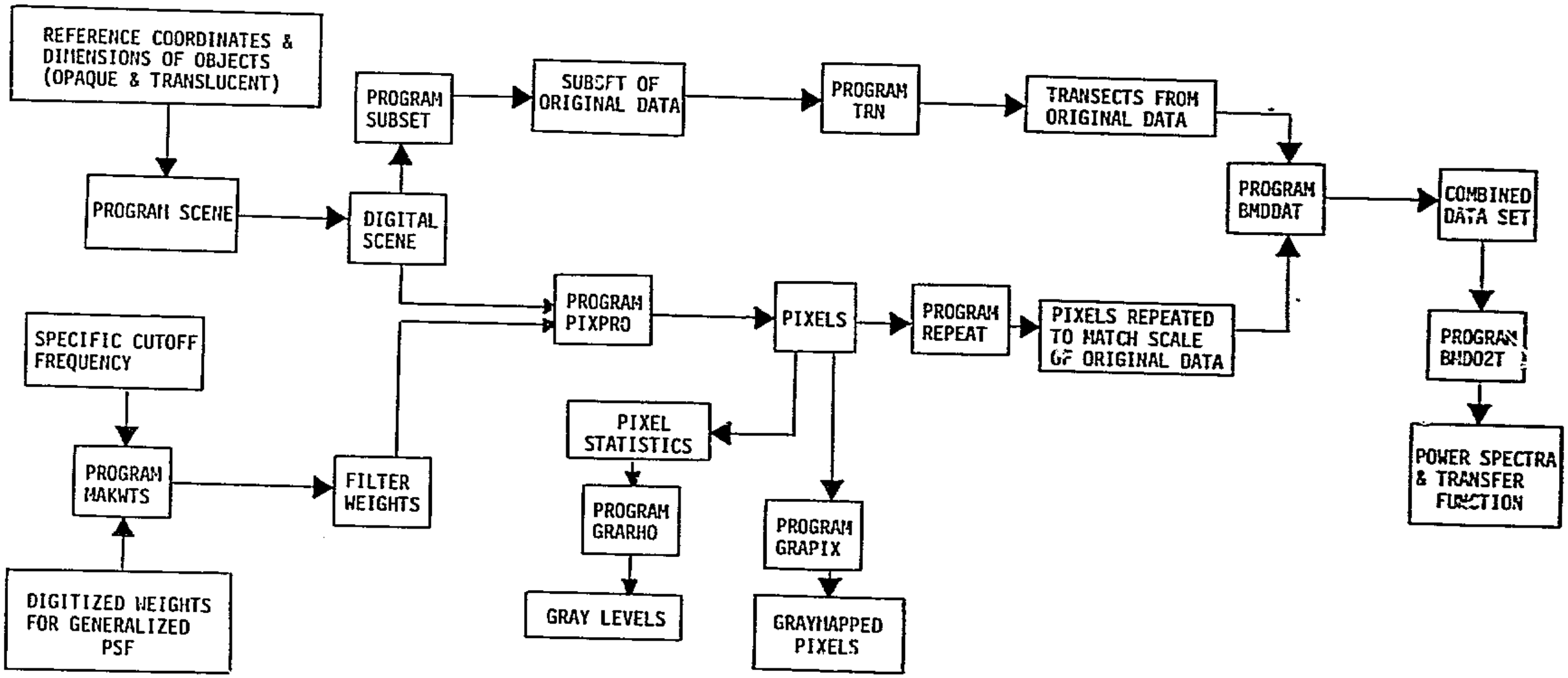


Figure A-1. Flow of data through programs.

PROGRAM SCENE

This program creates objects within a scene (array) by reading-in (from a plan or picture) object number, object location (coordinates of the upper left corner for rectangles, center for circles, etc.), object shape (rectangle, circle, etc.), object size (length, width, radius, etc.), and cover type, and then filling the scene array with the object using this data. The size of the plan or picture is specified in inches, the reference coordinates for objects are given in inches from the top and left of the plan, and the object dimensions are specified in inches. The number of cells per inch is read-in, which specifies the scale of the resulting scene (or, the distance covered by each scene array element). Objects at an angle from the top and left side of the plan can be subdivided into rectangular parts which are parallel with the sides of the plan. At present, the program processes only circles and rectangles, but additional shapes could be accommodated with new subroutines.

The scene array is first filled with the most common cover type by assigning a five letter cover code name (e.g. GRASS) to each array element. Parameters of opaque objects are then read-in to the program which fills the appropriate array elements with the object cover code names. Finally, parameters of translucent objects are read-in to the program, and the program combines the cover codes for the underlying cover and translucent cover. The result is a digital scene array with each element assigned an opaque or combination opaque-translucent cover code.

To run the program:

1. Measure the length and width of the plan or picture in inches (DATLEN, DATWID).
2. Choose a cell size that will divide the length and width of the plan into the number of desired cells per inch (NCELSZ). Check the dimensions of NSCENE in program SCENE to make sure they accommodate the resulting number of cells in the length and width directions (NROW and NCOL).
3. NCELSZ, DATWID, and DATLEN are read-in using TAPE5, format 100, in program SCENE.
4. Create files for opaque and translucent objects. These files must contain object number, object part number, distance from left edge of plan to object reference point (inches), distance from top of plan (inches) to object reference point, object shape, object length if rectangle (inches), object width if rectangle (inches), object radius if circle (inches), and cover type (five letter code name). The reference point is the upper left corner for rectangles and the center for circles. Follow format 100 in program SCENE. The file of opaque object parameters should be attached to program SCENE as TAPE1, and the translucent object parameters as TAPE2. When program SCENE adds translucent objects, it combines the first three letters of the underlying cover name with the first two letters of the translucent cover name. Check the code names chosen for uniqueness.
5. Determine the dominant ground cover and specify this cover in program SCENE in the "DO 4" do-loop.

Output from Program SCENE:

Output of program SCENE is written to TAPE4 and TAPE7. TAPE4 will contain the scene written as a string of five letter code names with the scene column varying fastest (e.g. 1,1 1,2 1,3 2,1 2,2 2,3 ...). TAPE7 contains information on the scene dimensions, cell size, etc. (parameters of the scene array). The user can also follow the loop indices, if desired, by removing the comment specification ("C" in column 1) on the writes to TAPE7 which are set off in asteriks. These tapes must be saved as files.

PROGRAM MAKWTS

This program calculates the filter weights to assign to the scene elements based on the Point Spread Function (PSF) of the sensor. A generalized PSF is digitized and read-in as TAPE1. A cutoff frequency for a particular sensor is read-in and the program creates weights for scene elements as a function of distance from the center element and the distance covered per element in the scene array. The number of resulting weights depends on the distance represented by each scene element (DISAMP). The generalized PSF should always be digitized at an interval less than or equal to the distance per scene element (DISAMP).

To run the program:

1. Digitize the generalized PSF (must be digitized at an interval less than or equal to the distance represented by each scene element (DISAMP)). A generalized PSF for the research reported here is described in Section 3.2. These values are read-in to array XPSF using format 101.

2. Choose the cutoff frequency (CUTFRE).
3. The cutoff frequency (CUTFRE), number of digitized PSF values (NUMPTS), and distance represented by each scene element in meters (DISAMP) are read-in on TAPE5, using format 100.
4. Check the dimensions of the arrays: NUP dimensions should equal XPSF dimension plus one; XPSF dimension must be greater than or equal to the number of digitized PSF values; WEITS dimension can be set the same as for XPSF.

Output from Program MAKWTS:

The weights output from program MAKWTS are written to TAPE2 and printed after the program listing. TAPE2 must be saved as a file. The number of weights is printed in output, along with other parameters calculated in the program. The output weights are used in program PIXPRO to create a spatial filter, representing the sensor PSF, which is passed over the scene as the scene elements are aggregated into pixels.

PROGRAM PIXPRO

This program calculates pixels values by applying the filter weights output from program MAKWTS and aggregating the scene elements into pixels.

To run the program:

1. TAPE1 is the scene output from program SCENE and TAPE2 contains the weights output from program MAKWTS.

2. The scene array must be dimensioned at least as great as "NROW" and "NCOL" output from program SCENE. The weights arrays for rows (WEITR) and columns (WEITC) must be dimensioned at least as great as the number of weights output from program MAKWTS plus one (NUMWTS + 1 since the center element weight is labeled "0").
3. Before the program is run, the scene array cover types names must be changed to reflectance or radiance values. This can be accomplished with a system text editor (i.e. all occurrences of a particular cover name are replaced with a particular numerical value), or with a program which changes names to chosen numbers. It is the user's choice as to what scheme to use in assigning numerical values to the scene.
4. Program parameters are read-in using TAPE5 and format 100. Since the program as presently written uses a symmetrical PSF, the number of row and column weights (NUMWTR and NUMWTC) are the same and are equal to NUMWTS output from program MAKWTS. NSCLEN and NSCWID are the length and width in number of rows and columns output from program SCENE. A "zero" read-in for NSYM specifies that the PSF is symmetrical. The pixel width and length in columns and rows (PIXWID and PIXLEN) are specified in meters. The distance represented by each scene sample as output from program SCENE (DISAMP) is also specified in meters.

Output from Program PIXPRO:

Output pixels from the program are written to TAPE9 with each line of output having a pixel row and column specification plus a pixel value. This tape must be saved as a file. Program results printed after the program listing include: number of weights, pixel width and length, scene width and length in column and row dimensions, distance covered by each scene element, number of samples per width and length of the pixels, number or rows and columns of pixels output, and pixel row and column specification with each pixel value.

PROGRAM REPEAT

This program repeats the pixel values output from program PIXPRO to make the distance covered by each value equal to the distance covered by one original scene value (to make the scale equal for input and output values). The series of pixel values and series of original scene values will then have the same number of elements for computing the transfer function in program BMDO2T.

To run the program:

1. (a) Read-in the number of times the value must be repeated to match the scale of the original scene data - NUMREP (this is equal to NPXWID and NPXLEN from output of program PIXPRO for square pixels).

   (b) Read-in the starting (NRSTRT) and ending (NREND) row numbers for the rows of pixels needed to match the portion of the subscene output from program SUBSET, and used in program TRN to extract transects of original data. NUMREP, NRSTRT, and NREND are read-in using TAPE5 and format 99.

2. Pixel values are read-in using TAPE1 and format 100.

Output from Program REPEAT:

Output consists of a string of repeated pixel values which matches the scale of the original scene. The total number of values created is printed after the program listing. These values are combined with a series of original scene values output from program TRN using program BMDDAT, and the combined set is input to program BMD02T for calculation of the power spectra and transfer function. The repeated pixel values are written to TAPE2 and must be saved as a file.

PROGRAM SUBSET

This program creates a subset scene from the original scene that will match the area covered by the pixels output from program PIXPRO (edges of the original scene are truncated because the filter extends beyond the edge of each pixel). Thus, dimensions of the subset scene are based on the size of the pixels and the number of filter weights needed to create a pixel.

To run the program:

1. Dimension SCENE to match the dimensions of the original scene and read-in the original scene using TAPE1 and format 100.
2. Read-in, using TAPE5 and format 99: a) the number of weights in the row and column directions used to create a pixel - NUMWTR and NUMWTC; b) the number of rows and columns of pixels - NROWPX and NCOLPX; and c) the length and width of the pixels in number of samples from the original scene - NPXLEN and NPXWID. All of these values are output from program PIXPRO.

Output from Program SUBSET:

Output from the program is a string of values representing a subset of the original data which matches the area covered by the pixels output from program PIXPRO. These values are written to TAPE2 using format 100 and must be saved as a file. Information printed after the program listing includes the starting row and column in the original scene and the number of rows and columns in the output subscene.

PROGRAM TRN

This program extracts one-dimensional, horizontal transects from a subset of the original scene data for later comparison of the frequency content with pixels output from program PIXPRO.

To run the program:

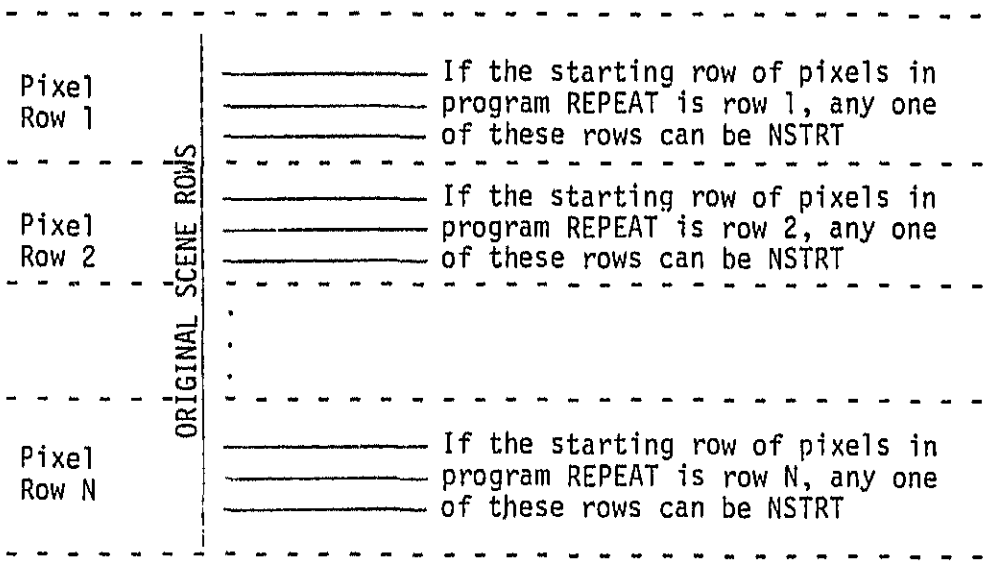
1. Dimension SCENE to the dimension of the original scene data (or use NROWS and NCOLS output from program SUBSET to set these dimensions).
2. Read-in the number of rows and columns in the subset of the original scene data using TAPE5 and format 99. These values were output from program SUBSET.
3. Read-in the subscene that was output from program SUBSET using TAPE1 and format 100.
4. Read-in NSTRT, NSTOP, INTERV using TAPE5 and format 101. These values are determined in the following manner:

   NSTRT can be any one of several rows in the subset data which corresponds to the first row of pixels chosen in program REPEAT to be comapred with the original scene data. This is due to the fact that one row of pixels represents several rows of original scene data (the pixel is an aggregate of several rows and columns of original scene data). If the first row of pixels output from program PIXPRO is the starting row in program REPEAT, then NSTRT can be arbitrarily chosen to be ½ of the pixel length (NPXLEN) from program PIXPRO.

NSTOP is the number of rows in the subscene needed to extract 1,000 values (BMDO2T takes only 1,000 points per series) which depends on the number of samples per row in the input subset of original data (NCOL).

INTERV is the number of samples per pixel (NPXLEN from program PIXPRO).

A schematic of the relationship of starting pixel row, in program REPEAT, to NSTRT is given below.



Output from Program TRN:

The transect values are written as a series of numbers to TAPE2 using format 100 and must be saved as a file. These values are combined with output from program REPEAT using program BMDDAT, and the combined data set is input to program BMD02T. The number of values written to TAPE2 is printed after the program listing.

PROGRAM BMDDAT

This program takes transect data from the original scene (output from program TRN) and combines it with repeated pixel values (output from program REPEAT) to form a data set for input to program BMD02T.

To run the program:

1. Read-in the total number of points in each set (NUMPTS) from TAPE5. NUMPTS is the lesser of the number of points output from program REPEAT and TRN. NUMPTS cannot exceed 1,000 due to the limits of program BMD02T. This limit may be changed if another power spectra program is used that handles more points per series.
2. Read-in the transect data on TAPE1 using format 100.
3. Read-in the pixel values on TAPE2 using format 200.

Output from Program BMDDAT:

The output values, representing two series (one following the other) of points, is output to TAPE3 and must be saved. The number of points per series is printed after the program listing. The data set is input to program BMD02T.

PROGRAM BMDO2T

This program is taken from Biomedical Programs (Dixon, 1971) and performs power spectra analyses and calculates transfer functions between series of points. The BMD manual details the input and output for the program and gives examples.

PROGRAM GRARHO

This program calculates arbitrary graymapping intervals for later use in program GRAPIX. To run the program, the user specifies the minimum graymap level (XMIN), the number of levels to calculate (NUMRHO), and the interval between levels (XINT). NUMRHO is limited by the number of symbols available in program GRAPIX for graymapping. XMIN, NUMRHO, and XINT are read-in from TAPE5 using format 100. Many other schemes could be used for calculating graylevels, such as histogram divisions, etc. This choice is left to the user.

Output from Program GRARHO:

Output from the program is a series of graylevels which is written to TAPE1 using format 200 and must be saved as a file. This file is used as input to program GRAPIX.

PROGRAM GRAPIX

This program graymaps the pixels output from program PIXPRO at the same scale as the input scene so comparisons of spatial detail can be made.

To run the program:

1. Read-in NWID, NCOL, NCOLPX, and NUMRHO using TAPE5 and format 100. NWID and NCOL are the number of rows and columns of elements in the original scene that would be needed to equal the size of one pixel (i.e. NPXWID and NPXLEN from program PIXPRO). NCOLPX is the number of pixels per row output from program PIXPRO. NUMRHO is the number of graymap levels output from program GRARHO.
2. Read-in array RHO which contains the graymap levels output from program GRARHO, using TAPE1 and format 101.

Output from Program GRAPIX:

Output from the program is a graymap of the pixels at the same scale as the input scene. This format allows comparison of the amount of spatial detail (modulation) that was transferred from the original scene to the output scene (pixels).

PROGRAM ATMOS

This program adds the effect of the atmosphere to the pixels output from program PIXPRO. A transmission coefficient and path radiance value are read-in to the program. These values could also be applied directly to the original scene reflectance or radiance values.

To run the program:

1. Read-in a transmission coefficient "TRANS" and path radiance value "PATHRA" using TAPE5 and format 100.
2. Read-in the pixel values (and their row and column designations) from TAPE1 using format 101.

Output from Program ATMOS:

Output of the program is a string revised pixel value (and their row and column identification). Output is written to TAPE2 and must be saved as a file.

REFERENCES

Dixon, W. J., Ed. 1971. BMD Biomedical Computer Programs. U. of California Publications in Automatic Computation No. 2, U. of California Press, Berkeley, California.

PROGRAM LISTINGS

PROGRAM SCENE


```
/EOR
      PROGRAM SCENE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2,TAPE3,TAPE4,TAPE7)
C
C.....TAPES 1 AND 2 ARE INPUT DATA (OPAQUE AND TRANSLUCENT OBJECTS
C     RESPECTIVELY), TAPE3 IS AN INTERMEDIATE TAPE, TAPE4 IS OUTPUT,
C     AND TAPE7 IS USED TO CHECK ARRAY DIMENSION, CELL SIZE, ETC.
C
C
C.....THIS PROGRAM CREATES OBJECTS WITHIN A SCENE (ARRAY) BY
C     READING IN AN OBJECT NUMBER, LOCATION (COORDINATES OF THE UPPER-LEFT
C     CORNER FOR RECTANGLES, AND CENTER FOR CIRCLES, ETC.), THE SHAPE, THE SIZE
C     OF THE OBJECT (LENGTH, WIDTH, RADIUS), AND THE COVER TYPE, AND
C     THEN FILLING THE SCENE ARRAY WITH THE OBJECT.
C
C.....VARIABLE LIST:
C
C     DATLEN  LENGTH OF SCENE IN INCHES
C     DATWID  WIDTH OF SCENE IN INCHES
C     DISLFT  DISTANCE (INCHES) FROM LEFT EDGE OF PLAN TO REFERENCE POINT
C     DISTOP  DISTANCE (INCHES) FROM TOP OF PLAN TO REFERENCE POINT FOR OJBECT
C     NCELSZ  NUMBER OF CELLS PER INCH
C     NCOL    NUMBER OF COLUMNS IN SCENE
C     NCOMB   COMBINATION OF OPAQUE AND TRANSLUCENT OBJECT NAMES
C     NCOVER  COVER TYPE
C     NROW    NUMBER OF ROWS IN SCENE
C     NSCENE  ARRAY FOR STORING OBJECTS
C     NSHAPE  SHAPE OF OBJECT
C     NUMOB   OBJECT NUMBER
C     NUMPRT  OBJECT PART NUMBER
C     RAD     RADIUS (INCHES) OF CIRCULAR OBJECT
C     WID     WIDTH (INCHES) OF RECTANGULAR OBJECT
C     XLEN    LENGTH (INCHES) OF RACTANGULAR OBJECT
C
C     ********************************************************
      COMMON NSCENE(100,170), NCELSZ, NCOL, NROW
C
C.....READ IN THE VALUES FOR CELL SIZE, SCENE WIDTH AND SCENE LENGTH USING
C     FORMAT 100.
C
      READ(5,100)NCELSZ, DATWID,DATLEN
C
C.....CALCULATE THE NUMBER OF ROWS AND COLS IN THE SCENE ARRAY (SHOULD BE
C     LESS THAN OR EQUAL TO THE DIMENSIONS SPECIFIED IN COMMON).
C
      NCOL=INT((DATWID*FLOAT(NCELSZ))+.5)
      NROW=INT((DATLEN*FLOAT(NCELSZ))+.5)
C
C**********
      WRITE(7,599)
 599  FORMAT(*THE SCENE PARAMETERS ARE*)
      WRITE(7,600) NCELSZ,DATWID,DATLEN,NCOL,NROW
 600  FORMAT(*NCELSZ=*,I5,1X,*DATWID=*,F7.3,1X,*DATLEN=*,F7.3,1X
     +*NCOL=*,I5,1X,*NROW=*,I5)
C**********
C
C.....FILL THE SCENE WITH THE DOMINANT COVER.
C
      DO 4 I=1,100
         DO 3 J=1,170
            NSCENE(I,J)=5HGRASS
 3       CONTINUE
 4    CONTINUE
 1    READ(1,101) NUMOB,NUMPRT,DISLFT,DISTOP,NSHAPE,XLEN,WID,RAD,NCOVER
```

```
C
C.....IF FINISHED WITH OPAQUE OBJECTS, GOTO TRANSLUCENT OBJECTS (500).
C
      IF(EOF(1)) 500,2
C
C**********
C     WRITE(7,601) NUMOB,NUMPRT,DISLFT,DISTOP,NSHAPE,
C    +XLEN,WID,RAD,NCOVER
 601  FORMAT(2I4,2F5.1,A5,3F5.1,A5)
C**********
C
C.....CALCULATE THE COL NUMBER FOR THE OBJECT REFERENCE COORDINATE.
C
 2    J=INT((DISLFT*FLOAT(NCELSZ))+.5)
C
C**********
C     WRITE(7,602) J
 602  FORMAT("J=",I5)
C**********
C
      IF (J .LE. 0) J=1
C
C**********
C     WRITE(7,603) J
 603  FORMAT("J=",I5)
C**********
C
C.....CALCULATE THE ROW NUMBER FOR THE OBJECT REFERENCE COORDINATE.
C
      I=INT((DISTOP*FLOAT(NCELSZ))+.5)
C
C**********
C     WRITE(7,604) I
 604  FORMAT("I=",I5)
C**********
C
      IF (I .LE. 0) I=1
C
C**********
C     WRITE(7,604) I
C**********
C
C.....ASSIGN THE COVER TYPE TO THE OBJECT REFERENCE COORDINATE.
C
      NSCENE(I,J) = NCOVER
C
C**********
C     WRITE(7,605) NSCENE(I,J)
 605  FORMAT("NSCENE(I,J)=",A5)
C**********
C
C.....CALL THE APPROPRIATE SUBROUTINE DEPENDING ON THE OBJECT SHAPE.
C
      IF (NSHAPE .EQ. 4HRECT)
     +       CALL RECT(NUMOB,NUMPRT,XLEN,WID,NCOVER,I,J)
      IF (NSHAPE .EQ. 4HCIRC)
     +       CALL CIRC(NUMOB,NUMPRT,RAD,NCOVER,I,J)
C
      GOTO 1
C
 100  FORMAT(I5,2F10.4)
 101  FORMAT(2I5,2F10.4,A4,3F10.4,A5)
C
C.....READ IN A TRANSLUCENT OBJECT PARAMETERS AND FOLLOW SAME PROCEDURE
C     AS FOR OPAQUE OBJECTS ABOVE.
C
 500  READ(2,101) NUMOB,NUMPRT,DISLFT,DISTOP,NSHAPE,XLEN,WID,RAD,NCOVER
      IF(EOF(2)) 301,30
 30   J=INT((DISLFT*FLOAT(NCELSZ))+.5)
      IF (J .LE. 0) J=1
      I=INT((DISTOP*FLOAT(NCELSZ))+.5)
      IF (I .LE. 0) I=1
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C**********
C      WRITE(7,200) I,J,NSCENE(I,J),NCOVER
 200  FORMAT("FILLING TAPE3 WITH",2I5,A3,A2)
C**********
C
C.....CALL THE APPROPRIATE SUBROUTINE DEPENDING ON OBJECT SHAPE.
C
      IF (NSHAPE .EQ. 4HRECT)
     +     CALL TRECT(NUMOB,NUMPRT,XLEN,WID,NCOVER,I,J)
      IF (NSHAPE .EQ. 4HCIRC) CALL TCIRC(NUMOB,NUMPRT,RAD,NCOVER,I,J)
C
      GOTO 50
C
C.....NOW REWIND TAPE 3 AND REPLACE OPAQUE OBJECTS WITH COMBINATION
C      OF OPAQUE AND TRANSLUCENT OBJECTS.
C
 501  REWIND 3
C
 60   READ(3,410) I,J,NCOMB
      IF(EOF(3)) 502,70
 70   NSCENE(I,J) = NCOMB
 410  FORMAT(2I5,A5)
      GOTO 60
C
 502  CONTINUE
C**********
C      WRITE(7,800)
 800  FORMAT("NOW WRITE RESULTS ON TAPE4")
C**********
C
C
      WRITE(4,102) ((NSCENE(I,J),J=1,NCOL),I=1,NROW)
 102  FORMAT(A5)
      STOP
      END
C
      SUBROUTINE RECT(NUMOB,NUMPRT,XLEN,WID,NCOVER,I,J)
      COMMON NSCENE(100,170), NCELSZ,NCOL,NROW
C
C.....THIS SUBROUTINE IS USED TO FILL THE SCENE WITH RECTANGULAR
C      OPAQUE OBJECT.
C.....THE IF CHECKS ON K1 AND K2 ARE TO MAKE SURE THE SCENE ARRAY BOUNDS
C      ARE NOT EXCEEDED.
C
C**********
C      WRITE(7,606)
 606  FORMAT("WENT TO SUBROUTINE RECT")
C**********
C
C**********
C      WRITE(7,607) NUMOB,NUMPRT,XLEN,WID,NCOVER,I,J
 607  FORMAT(2I4,2F5.1,A5,2I4)
C**********
C
      K1=I+INT((XLEN*FLOAT(NCELSZ))+.5)-1
      IF(K1 .EQ. 0) K1=1
      IF(K1 .GT. NROW) K1=NROW
C
C**********
C      WRITE(7,666) K1
 666  FORMAT("K1=",I5)
C**********
C
      K2=J+INT((WID*FLOAT(NCELSZ))+.5)-1
      IF(K2 .EQ. 0) K2=1
      IF(K2 .GT. NCOL) K2=NCOL
C
C**********
C      WRITE(7,667) K2
 667  FORMAT("K2=",I5)
C**********
C
```

```
      DO 10 IND1=I,K1
          DO 5 IND2=J,K2
             NSCENE(IND1,IND2) = NCOVER
C
C**********
C                WRITE(7,608) IND1,IND2,NCOVER
 608             FORMAT(2I5,A5)
C**********
C
 5           CONTINUE
 10   CONTINUE
      RETURN
      END
C
      SUBROUTINE CIRC(NUMOB,NUMPRT,RAD,NCOVER,I,J)
      COMMON NSCENE(100,170), NCELSZ,NCOL,NROW
C
C.....THIS SUBROUTINE IS USED TO FILL THE SCENE WITH CIRCULAR
C     OPAQUE OBJECTS.
C
C.....THE IF CHECKS ON I1,I2,J1,J2,NROWT,NROWB,K1, AND K2 ARE TO MAKE
C     SURE THE SCENE ARRAY BOUNDS ARE NOT EXCEEDED.
C
C
C**********
C     WRITE(7,607)
 607  FORMAT("WENT TO SUBROUTINE CIRC")
C**********
C
C**********
C     WRITE(7,608) NUMOB,NUMPRT,RAD,NCOVER,I,J
 608  FORMAT(2I4,F5.1,A5,2I5)
C**********
C
C     I AND J ARE THE CENTER COORDS OF THE CIRCLE; RAD IS RADIUS.
C
      NSIDE1=INT((RAD*FLOAT(NCELSZ))+.5)
C
C**********
C     WRITE(7,444) NSIDE1
 444  FORMAT("NSIDE1=",I5)
C**********
C
C     FILL THE FOUR DISTANCES FROM THE CENTER FIRST.
      NDEX=NSIDE1-1
      IF(NDEX .EQ. 0) RETURN
      I1=I-NDEX
      IF(I1 .LE. 0) I1=1
      I2=I+NDEX
      IF(I2 .GT. NROW) I2=NROW
      DO 10 IND1=I1,I2
             NSCENE(IND1,J)=NCOVER
 10   CONTINUE
      J1=J-NDEX
      IF(J1 .LE. 0) J1=1
      J2=J+NDEX
      IF(J2 .GT. NCOL) J2=NCOL
      DO 20 IND2=J1,J2
             NSCENE(I,IND2)=NCOVER
 20   CONTINUE
C
C     IND3 IS USED TO COUNT NUMBER OF ROWS MUST CONSIDER BETWEEN
C     CENTER AND EDGE OF CIRCLE = RADIUS-2 SINCE KNOW TOP AND MIDDLE.
      L1=NSIDE1-2
C
C**********
C     WRITE(7,445) L1
 445  FORMAT("L1=",I5)
C**********
      IF(L1 .EQ. 0) RETURN
C
```

```
      DO 50 IND3=1,L1
C
C     USE P-THEOREM TO FIND THIRD SIDE
C
            NSIDE2=NSIDE1-IND3
            X1=FLOAT(NSIDE1)**2.
            X2=FLOAT(NSIDE2)**2.
            X3=((X1-X2)**.5)+.5
            NUMCOL=INT(X3)
            NROWT=I-(NSIDE1-(IND3+1))
            IF(NROWT .LE. 0) NROWT=1
            NROWB=I+(NSIDE1-(IND3+1))
            IF(NROWB .GT. NROW) NROWB=NROW
            K1=J-NUMCOL
            IF(K1 .LE. 0) K=1
            K2=J+NUMCOL
            IF(K2 .GT. NCOL) K2=NCOL
C
            DO 40 K3=K1,K2
                  NSCENE(NROWT,K3)=NCOVER
C**********
C                 WRITE(7,700) NROWT,K3,NCOVER
 700              FORMAT(2I5,A5)
C**********
C
                  NSCENE(NROWB,K3)=NCOVER
C
C                 WRITE(7,701) NROWB,K3,NCOVER
 701              FORMAT(2I5,A5)
C
 40         CONTINUE
 50   CONTINUE
      RETURN
      END
C
      SUBROUTINE TRECT(NUMOB,NUMPRT,XLEN,WID,NCOVER,I,J)
      COMMON NSCENE(100,170), NCELSZ,NCOL,NROW
C
C.....THIS SUBROUTINE IS USED TO WRITE THE LOACTIONS OF RECTANGULAR
C     TRANSLUCENT OBJECTS ONTO TAPE3 FOR SUBSEQUENT IMPOSITION ON THE
C     OPAQUE OBJECTS.
C
C.....THE IF CHECKS ON K1 AND K2 ARE TO MAKE SURE THE SCENE ARRAY BOUNDS
C     ARE NOT EXCEEDED.
C
C
C     WRITE(7,608)
 608  FORMAT(*WENT TO SUBROUTINE TRECT*)
C
      WRITE(3,100) I,J,NSCENE(I,J),NCOVER
C
      K1=I+INT((XLEN*FLOAT(NCELSZ))+.5)-1
C
C**********
C     WRITE(7,997) K1
 997  FORMAT(*K1=*,I5)
C**********
C
      IF(K1 .EQ. 0) K1=1
      IF(K1 .GT. NROW) K1=NROW
      K2=J+INT((WID*FLOAT(NCELSZ))+.5)-1
C
C**********
C     WRITE(7,998) K2
 998  FORMAT(*K2=*,I5)
C**********
C
      IF(K2 .EQ. 0) K2=1
      IF(K2 .GT. NCOL) K2=NCOL
C
      DO 10 IND1=I,K1
            DO 5 IND2=J,K2
                  WRITE(3,100) IND1,IND2,NSCENE(IND1,IND2),NCOVER
 5          CONTINUE
 10   CONTINUE
```

```
 100  FORMAT(2I5,A3,A2)
C
      RETURN
      END
C
      SUBROUTINE TCIRC(NUMOB,NUMPRT,RAD,NCOVER,I,J)
      COMMON NSCENE(100,170), NCELSZ,NCOL,NROW
C
C.....THIS SUBROUTINE IS USED TO WRITE THE LOCATIONS OF CIRCULAR
C     TRANSLUCENT OBJECTS ONTO TAPE3 FOR SUBSEQUENT IMPOSITION ON
C     THE OPAQUE OBJECTS.
C
C.....THE IF CHECKS ON I1,I2,J1,J2,NROWT,NROWB,K1, AND K2 ARE TO MAKE
C     SURE THE SCENE ARRAY BOUNDS ARE NOT EXCEEDED.
C
C
C**********
C     WRITE(7,609)
C609  FORMAT("WENT TO SUBROUTINE TCIRC")
C**********
C
      WRITE(3,100) I,J,NSCENE(I,J),NCOVER
C
      NSIDE1=INT((RAD*FLOAT(NCELSZ))+.5)
C
C**********
C     WRITE(7,303) NSIDE1
C303  FORMAT("NSIDE1=",I5)
C**********
C
      NDEX=NSIDE1-1
C
      IF(NDEX .EQ. 0) RETURN
C
      I1=I-NDEX
      IF(I1 .LE. 0) I=1
      I2=I+NDEX
      IF(I2 .GT. NROW) I2=NROW
C
      DO 10 IND1=I1,I2
         WRITE(3,100) IND1,J,NSCENE(IND1,J),NCOVER
 10   CONTINUE
C
      J1=J-NDEX
      IF(J1 .LE. 0) J1=1
      J2=J+NDEX
      IF(J2 .GT. NCOL) J2=NCOL
C
      DO 20 IND2=J1,J2
         WRITE(3,100) I,IND2,NSCENE(I,IND2),NCOVER
 20   CONTINUE
C
      L1=NSIDE1-2
C
C**********
C     WRITE(7,304) L1
C304  FORMAT("L1=",I5)
C**********
C
      IF(L1 .EQ. 0) RETURN
C
C
      DO 50 IND3=1,L1
         NSIDE2=NSIDE1-IND3
         X1=FLOAT(NSIDE1)**2.
         X2=FLOAT(NSIDE2)**2.
         X3=((X1-X2)**.5)+.5
         NUMCOL=INT(X3)
         NROWT=I-(NSIDE1-(IND3+1))
         IF(NROWT .LE. 0) NROWT=1
         NROWB=I+(NSIDE1-(IND3+1))
         IF(NROWB .GT. NROW) NROWB=NROW
         K1=J-NUMCOL
         IF(K1 .LE. 0) K1=1
         K2=J+NUMCOL
         IF(K2 .GT. NCOL) K2=NCOL
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
            DO 40 K3=K1,K2
                  WRITE(3,100) NROWT,K3,NSCENE(NROWT,K3),NCOVER
                  WRITE(3,100) NROWB,K3,NSCENE(NROWB,K3),NCOVER
 40         CONTINUE
 50   CONTINUE
C
 100  FORMAT(2I5,A3,A2)
C
      RETURN
      END
/EOR
    8   21.1875    12.500
/EOF
```

PROGRAM MAKWTS

```
/EOR
      PROGRAM MAKWTS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2)
C
C.....THIS PROGRAM CALCULATES THE WEIGHTS TO ASSIGN TO THE SCENE ELEMENTS
C     BASED ON THE POINT SPREAD FUNCTION (PSF).  THE NUMBER OF RESULTING
C     WEIGHTS DEENDS ON THE DISTANCE REPRESENTED BY EACH SCENE SAMPLE
C     (DISAMP).  THE PSF SHOULD ALWAYS BE DIGITIZED AT AN INTERVAL .LE.
C     THE DISTANCE PER SCENE SAMPLE (DISAMP).
C
C     TAPE1 IS THE DIGITIZED PSF VALUES, TAPE2 IS THE OUTPUT WEIGHTS.
C
C********** VARIABLE LIST **********
C
C     CENWT    THE WEIGHT ASSIGNED TO THE CENTER SCENE SAMPLE
C     CUTFRE   CUTOFF FREQUENCY (IN METERS)
C     DISAMP   DISTANCE REPRESENTED BY EACH SCENE SAMPLE (IN METERS)
C     DISINT   DISTANCE REPRESENTED BY EACH DIGITIZED PSF VALUE
C     NDROP    NUMBER OF PSF VALUES TO AVERAGE TO DETERMINE THE CENTER
C              SAMPLE WEIGHT
C     NUMPTS   NUMBER OF DIGITIZED PSF VALUES
C     NUMWTS   NUMBER OF WEIGHTS TO BE CALCULATED, OTHER THAN THE WEIGHT
C              FOR THE CENTER SAMPLE
C     VALAVE   NUMBER OF PSF VALUES TO AVERAGE TO EQUAL THE DISTANCE COVERED
C              BY ONE SCENE SAMPLE
C     WEITS    ARRAY FOR STORING CALCULATED WEIGHTS FOR THE SCENE SAMPLES
C     XPSF     ARRAY FOR STORING DIGITIZED PSF VALUES
C
C*****************************************
C
      DIMENSION NUP(51), XPSF(50), WEITS(50)
      READ(5,100) CUTFRE,NUMPTS,DISAMP
      WRITE(6,99) CUTFRE,NUMPTS,DISAMP
 99   FORMAT(1X,*CUTFRE=*,F7.3,*NUMPTS=*,I5,*DISAMP=*,F7.3)
C
C.....READ THE DIGITIZED PSF VALUES INTO ARRAY XPSF (NOTE FORMAT).
C
      READ(1,101) (XPSF(I),I=1,NUMPTS)
 100  FORMAT(F7.3,I5,F7.3)
 101  FORMAT(F5.2)
C
C.....CALCULATE THE DISTANCE REPRESENTED BY EACH DIGITIZED PSF VALUE.
C
      DISINT=CUTFRE/FLOAT(NUMPTS)
      WRITE(6,98) DISINT
 98   FORMAT(1X,*DISINT=*,F10.4)
```

```
C
C.....CALCULATE THE NUMBER OF PSF VALUES TO AVERAGE TO EQUAL THE DISTANCE
C     COVERED BY ONE SCENE SAMPLE.
C
      VALAVE=DISAMP/DISINT
      WRITE(6,97) VALAVE
 97   FORMAT(1X,"VALAVE=",F10.4)
C
C.....CALCULATE THE NUMBER OF PSF VALUES TO AVERAGE TO DETERMINE THE CENTER
C     SAMPLE WEIGHT.
C
      NDROP=INT((VALAVE/2.)+.5)
      WRITE(6,88) NDROP
 88   FORMAT(1X,"NDROP=",I5)
C
C.....CALCULATE THE CENTER WEIGHT.
C
      SUM=0.
      DO 2 I=1,NDROP
         SUM=SUM+XPSF(I)
 2    CONTINUE
      CENWT=SUM/FLOAT(NDROP)
      WRITE(6,84) CENWT
 84   FORMAT(1X,"THE CENTER PIXEL WEIGHT IS:",F10.4)
      WRITE(2,83) CENWT
 83   FORMAT(5H    0,F10.4)
C
C.....CALCULATE THE NUMBER OF WEIGHTS TO BE DETERMINED.
C
      NUMWTS=(NUMPTS-NDROP)/VALAVE
      WRITE(6,96) NUMWTS
 96   FORMAT(1X,"NUMWTS=",I10)
C.....NOW MULTIPLY VALAVE BY 1 TO NUMWTS TO GET UPPER LIMIT OF INTERVALS.
      NUP(1)=NDROP
      DO 5 I=1,NUMWTS
         NUP(I+1)=INT((I*VALAVE)+.5)+NDROP
 5    CONTINUE
      NUPTOP=NUMWTS+1
      IF(NUP(NUPTOP) .GT. NUMPTS) WRITE(6,90)
 90   FORMAT(1X,"EXCEEDED NUMBER OF INPUT WEIGHTS")
      WRITE(6,85)
 85   FORMAT(1X,"UPPER LIMIT OF VALUES TO AVERAGE ARE:")
      WRITE(6,95) (NUP(I),I=1,NUPTOP)
 95   FORMAT(1X,I5)
C
C.....NOW AVERAGE THE APPROPRIATE PSF VALUES TO OBTAIN THE WEIGHTS.
C
      J=NDROP
      K=0
      DO 10 JJ=1,NUMWTS
         NN=0
         SUM=0.
         NDIF=NUP(JJ+1)-NUP(JJ)
         WRITE(6,94) NDIF
 94      FORMAT(1X,"NDIF=",I5)
         DO 7 KK=1,NDIF
            NN=NN+1
            J=J+1
            SUM=SUM+XPSF(J)
 7       CONTINUE
         SUM=SUM/NN
         K=K+1
         WEITS(K)=SUM
 10   CONTINUE
C
C.....SUM THE PSF VALUES THAT ARE LEFT IF THERE ARE AT LEAST VALAVE/2 OF THEM.
C
      LFTWTS=NUMPTS-NUP(NUPTOP)
      WRITE(6,93) LFTWTS
 93   FORMAT(1X,"LFTWTS=",I5)
      IF(FLOAT(LFTWTS) .LT. VALAVE/2.) GOTO 40
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      K=K+1
      SUM=0.
      DO 39 I=1,LFTWTS
         J=J+1
         SUM=SUM+XPSF(J)
 39   CONTINUE
      SUM=SUM/VALAVE
      WEITS(K)=SUM
C
C.....WRITE OUT THE WEIGHTS AND SAVE THEM ON TAPE2.
C
 40   WRITE(6,200)
      WRITE(6,201) ((I,WEITS(I)),I=1,K)
      WRITE(2,202) ((I,WEITS(I)),I=1,K)
 200  FORMAT(1X,"THE WEIGHTS ARE:")
 201  FORMAT(1X,I5,F10.4)
 202  FORMAT(I5,F10.4)
      STOP
      END
/EOR
   30.0    50   1.905
/EOF
```

PROGRAM PIXPRO

```
/EOR
      PROGRAM PIXPRO(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2,TAPE3,TAPE9)
C
C.....THIS PROGRAM CALCULATES PIXEL VALUES BY APPLYING THE WEIGHTS OUTPUT
C     FROM PROGRAM MAKWTS AND AGGREGATING THE SCENE SAMPLES INTO PIXELS.
C     TAPE1 IS THE SCENE OUTPUT FROM PROGRAM SCENE,
C     TAPE2 IS THE WEIGHTS OUTPUT FROM PROGRAM MAKWTS, AND TAPE9 IS THE
C     PIXELS OUTPUT FROM THIS PROGRAM.
C
C.....THE SCENE ARRAY MUST BE DIMENSIONED AT LEAST AS GREAT AS "NROW" AND
C     "NCOL" OUTPUT FROM PROGRAM SCENE.  THE WEIGHTS ARRAYS (WEITR & WEITC)
C     MUST BE DIMENSIONED AT LEAST AS GREAT AS THE NUMBER OF WEIGHTS OUTPUT
C     OUTPUT FROM PROGRAM MAKWTS + 1 (I.E., NUMWTS+1, SINCE THE CENTER WEIGHT
C     IS LABELLED 0).
C     BEFORE THIS PROGRAM IS RUN, THE SCENE ARRAY COVER TYPE CODE NAMES
C     MUST BE CHANGED TO REFLECTANCE OR RADIANCE VALUES.  THIS CAN BE
C     ACCOMPLISHED WITH A SYSTEM TEXT EDITOR (I.E., ALL OCCURRENCES OF A
C     PARTICULAR COVER CODE NAME ARE REPLACED WITH A PARTICULAR NUMERICAL
C     VALUE), OR WITH A PROGRAM WHICH CHANGES NAMES TO CHOSEN NUMBERS.
C     IT IS THE USER'S CHOICE AS TO WHAT SCHEME TO USE IN ASSIGNING
C     NUMERICAL VALUES TO THE COVER CODE NAMES.
C
      DIMENSION SCENE(160,170),WEITR(40),WEITC(40)
      READ(5,100) NUMWTR,NUMWTC,NSCLEN,NSCWID,NSYM,PIXWID,PIXLEN,
     +DISAMP
 100  FORMAT(5I5,3F7.3)
C
      WRITE(6,106) NUMWTR,NUMWTC
 106  FORMAT(1X,"NUMBER OF WEIGHTS IN ROW AND COL DIRECTIONS ARE:
     +",2I10)
      WRITE(6,107) PIXWID,PIXLEN
 107  FORMAT(1X,"PIXEL WID AND LENGTH IN METERS ARE:
     +",2F10.2)
      WRITE(6,109) NSCWID,NSCLEN
 109  FORMAT(1X,"SCENE WIDTH AND LENGTH IN NUMBER OF SAMPLES ARE:
     +",2I10)
```

```
      WRITE(6,109) DISAMP
 109  FORMAT(1X,"DISTANCE PER SAMPLE (IN METERS) IS:
     +",F10.4)
      IF(NSYM .EQ. 0) WRITE(6,110)
 110  FORMAT(1X,"THE PSF IS SYMMETRICAL")
C
      IF(NSYM .EQ. 1) GOTO 500
      READ(2,101) CENWT, (WEITR(I),I=1,NUMWTR)
 101  FORMAT(5X,F10.4)
      READ(1,103) ((SCENE(I,J),J=1,NSCWID),I=1,NSCLEN)
 103  FORMAT(F5.2)
C
      NPXWID=INT((PIXWID/DISAMP)+.5)
      NPXLEN=INT((PIXLEN/DISAMP)+.5)
      WRITE(6,104) NPXWID,NPXLEN
 104  FORMAT(1X,"NUMBER OF SAMPLES PER WIDTH AND LENGTH OF PIXEL ARE:
     +",2I10)
C
      NROWPX=(NSCLEN-(2*NUMWTC))/NPXLEN
      NCOLPX=(NSCWID-(2*NUMWTR))/NPXWID
      WRITE(6,105) NROWPX,NCOLPX
 105  FORMAT(1X,"NUMBER OF ROWS AND COLS OF PIXELS IN OUTPUT ARE:
     +",2I10)
C
C.....WRITE HEADINGS FOR PIXEL OUTPUT.
      WRITE(6,111)
 111  FORMAT(1X,2X,"ROW",2X,"COL",7X,"RHO")
C
C.....NOW START THE PIXEL VALUE CALCULATIONS
C
      MIDPXR=NUMWTR+1
      DO 30 KR=1,NROWPX
           MIDPXC=NUMWTC+1
           DO 29 KC=1,NCOLPX
                SUM=SCENE(MIDPXR,MIDPXC)*CENWT
                SUMWT=CENWT
                DO 28 K1=1,NUMWTR
                     SUM=SUM+(WEITR(K1)*
     +                    (SCENE(MIDPXR,MIDPXC+K1)
     +                    +SCENE(MIDPXR+K1,MIDPXC)
     +                    +SCENE(MIDPXR-K1,MIDPXC)
     +                    +SCENE(MIDPXR,MIDPXC-K1)))
                     SUMWT=SUMWT+(4.*WEITR(K1))
 28             CONTINUE
C
                NUP=0
                DO 27 K2=1,NUMWTR
                     NR=MIDPXR+K2
                     NC=MIDPXC
                     NUP=NUP+2
                     NBAK=0
                     DO 26 K3=1,NUMWTC
                          NBAK=NBAK+2
                          NC=MIDPXC+K3
                          XWEIT=WEITR(K2)*WEITR(K3)
                          SUM=SUM+(XWEIT*(SCENE(NR,NC)
     +                       +SCENE(NR-NUP,NC)
     +                       +SCENE(NR-NUP,NC-NBAK)
     +                       +SCENE(NR,NC-NBAK)))
                          SUMWT=SUMWT+(4.*XWEIT)
 26                  CONTINUE
 27             CONTINUE
                CALL SAVPIX(SUM,SUMWT,KR,KC)
                MIDPXC=MIDPXC+NPXWID
 29        CONTINUE
           MIDPXR=MIDPXR+NPXLEN
 30   CONTINUE
      GOTO 501
 500  WRITE(6,112)
 112  FORMAT(1X,"NOT A SYMMETRICAL POINT SPREAD FUNCTION.")
 501  STOP
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C
      SUBROUTINE SAVPIX(SUM,SUMWT,KR,KC)
      PIXVAL=SUM/SUMWT
      WRITE(9,200) KR,KC,PIXVAL
  200 FORMAT(2I5,F10.4)
      WRITE(6,201) KR,KC,PIXVAL
  201 FORMAT(1X,2I5,F10.4)
      RETURN
      END
/EOR
   37    37  160  170     0 73.500 73.500  1.905
/EOF
```

PROGRAM REPEAT

```
/EOR
      PROGRAM REPEAT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2)
C
C.....THIS PROGRAM REPEATS PIXEL VALUES OUTPUT FROM PROGRAM PIXPRO TO MAKE
C     THE DISTANCE COVERED BY EACH VALUE EQUAL TO THE DISTANCE COVERED BY
C     ONE ORIGINAL SCENE VALUE.  THE SERIES OF PIXEL VALUES AND SERIES OF
C     ORIGINAL SCENE VALUES WILL THEN HAVE THE SAME NUMBER OF ELEMENTS FOR
C     COMPUTING THE TRANSFER FUNCTION IN PROGRAM 3MDO2T.
C     TAPE1 IS THE INPUT PIXELS FROM PROGRAM PIXPRO.
C
C
C.....READ IN THE NUMBER OF SCENE SAMPLES CONTAINED IN THE DISTANCE
C     COVERED BY ONE PIXEL (EQUAL TO "NPIXWID" AND "NPIXLEN" IN PROGRAM
C     PIXPRO).  THIS PROGRAM HANDLES ONLY SQUARE PIXELS AS FORMATTED NOW.
C     ALSO READ IN THE STARTING AND ENDING ROW NUMBERS FOR THE ROWS OF
C     PIXELS NEEDED TO MATCH THE PORTION OF THE SUBSCENE TO BE USED IN
C     PROGRAM TRN TO EXTRACT TRANSECTS.
C
      READ(5,99) NUMREP,NRSTRT,NREND
   99 FORMAT(3I5)
C
      NUMPTS=0
C
    4 READ(1,100) NROW,PIX
      IF(EOF(1)) 500,1
    1 IF(NROW .LT. NRSTRT) GOTO 4
      IF(NROW .GT. NREND) 500,3
    3 DO 12 K=1,NUMREP
         WRITE(2,200) PIX
         NUMPTS=NUMPTS+1
   12 CONTINUE
      GOTO 4
  100 FORMAT(I5,5X,F10.4)
  200 FORMAT(F10.4)
C
C.....WRITE OUT THE TOTAL NUMBER OF VALUES WRITTEN TO TAPE2.
C
  500 WRITE(6,300) NUMPTS
  300 FORMAT(1X,"TOTAL NUMBER OF POINTS OUTPUT=",I7)
C
      STOP
      END
/EOR
    5    6   12
/EOF
```

ORIGINAL PAGE IS
OF POOR QUALITY

PROGRAM SUBSET

```
/EOR
      PROGRAM SUBSET(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2)
C
C.....THIS PROGRAM CREATES A SUBSET SCENE FROM THE ORIGINAL SCENE THAT
C     WILL MATCH THE AREA COVERED BY THE PIXELS OUTPUT FROM
C     PIXRPO.  THE DIMENSIONS OF THE SUBSET AREA WILL BE BASED ON THE
C     SIZE OF THE PIXELS AND THE NUMBER OF WEIGHTS NEEDED TO CREATE A PIXEL.
C
      DIMENSION SCENE(160,170)
      READ(1,100) ((SCENE(I,J),J=1,170),I=1,160)
  100 FORMAT(F5.2)
C
C.....THE INDEX FOR "I" (ROWS) GOES FROM THE CENTER SAMPLE OF THE FIRST
C     PIXEL CREATED FROM THE SCENE, MINUS HALF THE PIXEL SIZE IN NUMBER OF
C     SAMPLES, TO THE NUMBER OF PIXELS * NUMBER OF SCENE SAMPLES PER PIXEL.
C     THE INDEX FOR "J" (COLS) IS CALCULATED THE SAME WAY.  NUMWTR, NUMWTC,
C     NROWPX, NCOLPX, NPXLEN, AND NPXWID ARE TAKEN FROM PROGRAM PIXPRO.
C
      READ(5,99) NUMWTR,NUMWTC,NROWPX,NCOLPX,NPXLEN,NPXWID
   99 FORMAT(6I5)
C
      NCENTR=NUMWTR+1
      NCENTC=NUMWTC+1
      K1=NCENTR-(NPXLEN/2)
      K2=(NPXLEN*NROWPX)-1+K1
      K3=NCENTC-(NPXWID/2)
      K4=(NPXWID*NCOLPX)-1+K3
C
C.....WRITE OUT THE DATA THAT IS NEEDED IN PROGRAM TRN.
C
      WRITE(6,300)
  300 FORMAT(1X,"INPUT TO PROGRAM TRN:")
      WRITE(6,301) K1,K3
  301 FORMAT(1X,"STARTING ROW AND COL IN ORIGINAL SCENE:",2I5)
      NROWS=NPXLEN*NROWPX
      NCOLS=NPXWID*NCOLPX
      WRITE(6,302) NROWS,NCOLS
  302 FORMAT(1X,"NUMBER OF ROWS AND COLS IN SUBSCENE:",2I5)
C
      DO 10 I=K1,K2
           DO 5 J=K3,K4
                WRITE(2,100) SCENE(I,J)
    5      CONTINUE
   10 CONTINUE
      STOP
      END
/EOR
    2    2   52   55    3    3
/EOF
```

PROGRAM TRN

```
/FOR
      PROGRAM TRN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2)
C
C.....THIS PROGRAM EXTRACTS HORIZONTAL (ONE DIMENSIONAL) TRANSECTS FROM A
C      SUBSET OF THE ORIGINAL SCENE DATA FOR LATER COMPARISON OF FREQUENCY
C      CONTENT WITH THE PIXELS OUTPUT FROM PROGRAM PIXPRO.
C
C.....INPUT (TAPE1) IS A SUBSET OF THE ORIGINAL DATA THAT MATCHES THE AREA
C      COVERED BY THE PIXELS OUTPUT FROM PIXPRO (NOTE THAT EDGES OF THE
C      ORIGINAL SCENE HAVE BEEN TRUNCATED TO ACCOUNT FOR THE SAMPLES NEEDED
C      TO CREATE THE VALUES FOR THE EDGE PIXELS).
C      THIS SUBSET OF THE ORIGINAL DATA IS OUTPUT FROM PROGRAM SUBSET.
C
C.....THE STARTING VALUE IN STMT 3 DO LOOP DEPENDS ON THE CHOICE OF THE
C      STARTING POINT FOR THE TRANSECTS BY THE USER, AND THE INTERVALS
C      DEPEND ON THE NUMBER OF TIMES THE PIXELS HAD TO BE REPEATED TO COVER
C      THE SAME DISTANCE AS THE SCENE SAMPLES USED TO CREATED THE PIXELS.
C      NSTOP IS LIMITED BY THE NUMBER OF ROWS OF DATA IN THE SUBSET OF
C      THE ORIGINAL DATA.
C
C.....NSTART CAN BE CHOSEN TO BE 1/2 NPXLEN (FROM PROGRAM PIXPRO); HOWEVER,
C      THE STARTING POINT MUST CORRESPOND TO THE STARTING PIXEL ROW
C      CHOSEN IN PROGRAM REPEAT.
C      NSTOP IS THE NUMBER OF ROWS IN THE SUBSCENE NEEDED TO EXTRACT
C      1000 VALUES (BMDMTF TAKES ONLY 1000 POINTS PER SERIES) WHICH DEPENDS
C      ON THE NUMBER OF SAMPLES PER ROW IN THE INPUT SUBSET OF ORIGINAL DATA;
C      INTERV IS THE NUMBER OF SAMPLES PER PIXEL (NPXLEN FROM PROGRAM PIXPRO).
C
C
C.....NROWS AND NCOLS COME FROM THE NUMBER OF ROWS AND COLS IN THE SUBSCENE
C      AS OUTPUT FROM PROGRAM SUBSET.
C
      DIMENSION SCENE(160,170)
      READ(5,99) NROWS, NCOLS
  99  FORMAT(2I5)
C
      READ(1,100) ((SCENE(I,J),J=1,NCOLS),I=1,NROWS)
      IF(EOF(1)) 500,3
  3   READ(5,101) NSTART, NSTOP, INTERV
 101  FORMAT(3I5)
C
C.....KEEP TRACK OF THE NUMBER OF TRANSECTS RECORDED ON TAPE2 SO THE TOTAL
C      NUMBER OF POINTS IN TAPE2 WILL BE KNOWN.
C
      NSUM=0
      DO 30 I=NSTART,NSTOP,INTERV
            WRITE(2,100) (SCENE(I,J),J=1,NCOLS)
            NSUM=NSUM+1
  30  CONTINUE
C
C.....WRITE OUT THE TOTAL NUMBER OF POINTS THAT WERE OUTPUT TO TAPE2.
C
      NUMPTS=NSUM*NCOLS
      WRITE(6,300) NUMPTS
 300  FORMAT(1X,"TOTAL NUMBER OF POINTS OUTPUT=",I5)
 100  FORMAT(F5.2)
 500  STOP
      END
/EOR
  156  165
   17   35    3
/EOF
```

PROGRAM BMDDAT

```
/EOR
      PROGRAM BMDDAT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2,TAPE3)
C
C.....THIS PROGRAM TAKES TRANSECT DATA FROM THE ORIGINAL SCENE (OUTPUT
C     FROM PROGRAM TRN) AND COMBINES IT WITH THE REPEATED PIXELS (OUTPUT
C     FROM REPEAT) TO FORM A DATA SET FOR INPUT TO PROGRAM BMDMTF.
C     TAPE1 IS THE TRANSECT DATA, TAPE2 IS THE REPEATED PIXELS, AND
C     TAPE3 IS THE OUTPUT DATA SET USED AS INPUT TO BMDMTF.
C
C.....BMDMTF WILL HANDLE ONLY 1000 POINTS PER SERIES.  THEREFORE, DO NOT
C     COMBINE MORE THAN 1000 POINTS FOR EACH OF THE TWO DATA SOURCES
C     (TRANSECTS & PIXELS).  THIS CONTROL IS HANDLED BY SETTING THE LIMITS
C     OF THE DO LOOP TO NUMPTS.  YOU MAY NOT HAVE 1000 POINTS AVAILABLE
C     IN EACH DATA SOURCE, THEREFORE NUMPTS WILL BE .LE. 1000.  THESE LIMITS
C     MAY BE CHANGED IF ANOTHER POWER SPECTRA PROGRAM IS CHOSEN THAT HANDLES
C     MORE DATA POINTS THAN BMD02T.
C
      READ(5,99) NUMPTS
   99 FORMAT(I5)
C
      DO 5 I=1,NUMPTS
           READ(1,100) X
           WRITE(3,200) X
    5 CONTINUE
  100 FORMAT(F5.2)
  200 FORMAT(F10.4)
      DO 15 I=1,NUMPTS
           READ(2,200) X
           WRITE(3,200) X
   15 CONTINUE
C
      WRITE(6,300)
  300 FORMAT(1X,"THE INFORMATION USED IN BMDMTF IS:")
      WRITE(6,301) NUMPTS
  301 FORMAT(1X,"NUMBER OF POINTS PER SERIES=",I5)
C
      STOP
      END
/EOR
 1000
/EOF
```

PROGRAM BMDO2T

```
/JOB
A771,T90,PR100. JCL TO RUN BMD02T TRANSFER FUNCTION
USER,
ROUTE,OUTPUT,ID=30,DC=LP,DEF.
GET,TAPE1=DAT485.
GET,STATLIB/UN=LIBRARY.
CALL(STATLIB(P=BMD02T))
LGO.
REWIND,OUTPUT.
REPLACE,OUTPUT=BMD485.
GOTO,DAYFILE.
EXIT.
DAYFILE,MYFILE.
REPLACE,MYFILE.
/EOR
PROBLMDAT485YES                    2100101  101    1. PIXEL                    171-1
(F10.4)
SELECT YES YES YES  18  01  238T
FINISH
/EOF
```

ORIGINAL PAGE IS OF POOR QUALITY

PROGRAM GRARHO

```
/EOR
      PROGRAM GRARHO(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
      DIMENSION RHO(40)
      READ(5,100) XMIN, NUMRHO, XINT
100   FORMAT(F10.4, I5,F10.4)
      RHO(1)=XMIN
      NUMIT=NUMRHO-1
      DO 5 I=1,NUMIT
         RHO(I+1)=RHO(I)+XINT
5     CONTINUE
      WRITE(1,200) (RHO(J),J=1,NUMRHO)
200   FORMAT(F10.4)
      STOP
      END
/EOR
  11.730      15      .6010
/EOF
```

PROGRAM GRAPIX

```
/EOR
      PROGRAM GRAPIX(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2)
C....................
C     THIS PROGRAM GRAYMAPS THE PIXELS AT THE SAME SCALE AS THE INPUT SCENE.
C     THE GRAYMAP SCALE (VARIABLE "RHO") IS OUTPUT FROM PROGRAM GRARHO
C     AND IS READ-IN TO THIS PROGRAM ON TAPE1.  TAPE2 CONTAINS THE PIXELS
C     OUTPUT FROM PROGRAM PIXPRO.  NCOL AND NWID ARE THE NUMBER OF COLUMNS
C     AND ROWS OF ELEMENTS FROM THE ORIGINAL SCENE USED TO PRODUCE A PIXEL.
C     NCOLPX IS THE NUMBER OF PIXELS PER ROW OUTPUT FROM PROGRAM PIXPRO.
C     NUMRHO IS THE NUMBER OF GRAYMAP LEVELS OUTPUT FROM PROGRAM GRARHO.
C....................
      DIMENSION LINE(125),LINREP(125)
      COMMON RHO(15)
C
      WRITE(6,99)
 99   FORMAT("1","GRAYMAP FOR RESOLUTION 1")
C
      READ(5,100) NWID,NCOL,NCOLPX,NUMRHO,XINT
100   FORMAT(4I5,F10.4)
      NLINE=NWID*NCOLPX
      IF(NLINE .GT. 125) NLINE=125
      READ(1,101) (RHO(I),I=1,NUMRHO)
C
101   FORMAT(F10.4)
3     K3=0
      DO 6 K1=1,NCOLPX
         READ(2,102) PIX
102      FORMAT(10X,F10.4)
         IF(EOF(2)) 50,4
4        CALL SYMBOL(PIX,NSYM,NSYM2)
         DO 5 K2=1,NWID
            K3=K3+1
            LINE(K3)=NSYM
            LINREP(K3)=NSYM2
5        CONTINUE
6     CONTINUE
```

```
      DO 7 I=1,NCOL
           WRITE(6,200) (LINE(JJ),JJ=1,NLINE)
 200       FORMAT(1X,78A1)
           WRITE(6,199) (LINREP(KK),KK=1,NLINE)
 199       FORMAT("+",78A1)
 7    CONTINUE
      GOTO 3
 500  CONTINUE
C
      WRITE(6,898)
 898  FORMAT("0","******  GRAYSCALE  ******")
      WRITE(6,900) RHO(2)
      WRITE(6,899)
      WRITE(6,901) RHO(2),RHO(3)
      WRITE(6,902) RHO(3),RHO(4)
      WRITE(6,903) RHO(4),RHO(5)
      WRITE(6,904) RHO(5),RHO(6)
      WRITE(6,905) RHO(6),RHO(7)
      WRITE(6,906) RHO(7),RHO(8)
      WRITE(6,907) RHO(8),RHO(9)
      WRITE(6,908) RHO(9),RHO(10)
      WRITE(6,909) RHO(10),RHO(11)
      WRITE(6,910) RHO(11),RHO(12)
      WRITE(6,911) RHO(12),RHO(13)
      WRITE(6,912) RHO(13),RHO(14)
      WRITE(6,913) RHO(14),RHO(15)
      WRITE(6,914) RHO(15)
 900  FORMAT(1X,"IF PIXEL .LT.",F10.4,22X,1HW)
 899  FORMAT("+",45X,1HM)
 901  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H#)
 902  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1HM)
 903  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H$)
 904  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1HX)
 905  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1HH)
 906  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1HI)
 907  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1HO)
 908  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1HU)
 909  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H*)
 910  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H+)
 911  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H/)
 912  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H=)
 913  FORMAT(1X,"IF PIXEL .GE.",F10.4,2X,"AND .LT.",F10.4,2X,1H-)
 914  FORMAT(1X,"IF PIXEL .GE.",F10.4,22X,1H.)
      WRITE(6,915) XINT
 915  FORMAT("0","THE DIFFERENCE BETWEEN GRAYMAP LEVELS IS",F10.4)
      STOP
      END
      SUBROUTINE SYMBOL(PIX,NSYM,NSYM2)
      COMMON RHO(15)
      NSYM2=1H
      IF(PIX .LT. RHO(2)) NSYM=1HM
      IF(PIX .LT. RHO(2)) NSYM2=1HW
      IF(PIX .GE. RHO(2) .AND. PIX .LT. RHO(3)) NSYM=1H#
      IF(PIX .GE. RHO(3) .AND. PIX .LT. RHO(4)) NSYM=1HM
      IF(PIX .GE. RHO(4) .AND. PIX .LT. RHO(5)) NSYM=1H$
      IF(PIX .GE. RHO(5) .AND. PIX .LT. RHO(6)) NSYM=1HX
      IF(PIX .GE. RHO(6) .AND. PIX .LT. RHO(7)) NSYM=1HH
      IF(PIX .GE. RHO(7) .AND. PIX .LT. RHO(8)) NSYM=1HI
      IF(PIX .GE. RHO(8) .AND. PIX .LT. RHO(9)) NSYM=1HO
      IF(PIX .GE. RHO(9) .AND. PIX .LT. RHO(10)) NSYM=1HU
      IF(PIX .GE. RHO(10) .AND. PIX .LT. RHO(11)) NSYM=1H*
      IF(PIX .GE. RHO(11) .AND. PIX .LT. RHO(12)) NSYM=1H+
      IF(PIX .GE. RHO(12) .AND. PIX .LT. RHO(13)) NSYM=1H/
      IF(PIX .GE. RHO(13) .AND. PIX .LT. RHO(14)) NSYM=1H=
      IF(PIX .GE. RHO(14) .AND. PIX .LT. RHO(15)) NSYM=1H-
      IF(PIX .GE. RHO(15)) NSYM=1H.
      RETURN
      END
/EOR
   39    39    2   15      .0813
/EOF
```

ORIGINAL PAGE IS OF POOR QUALITY

PROGRAM ATMOS

```
/EOR
      PROGRAM ATMOS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +TAPE1,TAPE2)
C.....THIS PROGRAM CHANGES THE PIXEL VALUES OUTPUT FROM PROGRAM PIXPRO
C     TO REFLECT THE EFFECTS OF THE ATMOSPHERE (TRANSMITTANCE AND PATH
C     RADIANCE).  "TRANS" IS THE TRANSMISSION COEFFICIENT: "PATHRA" IS
C     THE PATH RADIANCE.  THESE TWO VARIABLES ARE READ-IN TO THE PROGRAM
C     USING TAPE5 AND FORMAT 100.  THE PIXEL VALUES OUTPUT FROM PROGRAM
C     PIXPRO ARE READ-IN TO THE PROGRAM USING TAPE1 AND FORMAT 101.
C     OUTPUT PIXELS ARE WRITTEN TO TAPE2 AND MUST BE SAVED AS A FILE.
C.....
      READ(5,100) TRANS,PATHRA
 100  FORMAT(2F10.4)
 1    READ(1,101) NROW,NCOL,PIX
      IF(EOF(1)) 500,2
 2    PIX = (PIX*TRANS)+PATHRA
      WRITE(2,101) NROW,NCOL,PIX
 101  FORMAT(2I5,F10.4)
      GO TO 1
 500  WRITE(6,201) TRANS,PATHRA
 201  FORMAT(1X,"TRANSMISSION COEFF=",F10.4,"PATH RADIANCE=",F10.4)
      STOP
      END
/EOR
        .70        1.
/EOF
```