THE APPLICATION OF AN ILLUMINATION MODEL TO A MOUNTAINOUS LANDSAT SCENE

by

DAVID B. ELLIOTT

THESIS submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

APPROVED:

Robert M. Haralick

John W. Roach

Linda G. Shapiro

May, 1982
Blacksburg, Virginia

ACKNOWLEDGEMENTS

I would like to thank Dr. Robert Haralick for the guidance he has given me during my research, and for the time he has spent reading and suggesting changes to this thesis. Likewise, I would like to thank the other faculty members of my advisory committee: Dr. John Roach and Dr. Linda Shapiro. Others who have contributed to this work are Dr. James Campbell, Dr. Roger Ehrich, and Shyuan Wang who worked with Dr. Haralick, Dr. Shapiro, and me as a research group to investigate problems that directly relate to this topic. Dr. Layne Watson has also spent many hours working with me to explore various methods for solving some of the large linear systems of equations encountered in this work. To him I also say thanks.

I would like to acknowledge the great contributions made to this effort by my wife, Susan. She has not only been most patient and understanding with my work schedule, but has provided much needed encouragement. I also wish to thank her for typing the majority of the thesis.

CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

LIST OF FIGURES
(CONTINUED)

Chapter I

INTRODUCTION

The multispectral scanners (MSS) of the LANDSAT satellites collect solar radiation of different wavelengths reflected from the earth's surface. Separate bands are produced to record intensity levels of reflected light of different wavelengths. While the different greytone values of a given band of a LANDSAT image of simple terrain are due almost entirely to the various reflectivity values of the features of the earth's surface (such as vegetation, mineral deposits, or bodies of water), the same is not true of areas of complex topography. In mountainous areas, the mixture of light and dark regions on some given band in a LANDSAT image may be due to shadow effects as well as the reflectivity values for those wavelengths of light of the various surface features.

An illumination model is defined in this paper to be a function that produces an image composed of rows and columns of pixels with the properties that follow. As in a LANDSAT produced image, each pixel represents an area over which the value to be measured is integrated. In the illumination model the greytone values approximate intensities of reflected light for a given input surface and certain other

input parameters including the orientation of the reflecting surface, the position of the illumination source and intensity of the illumination from that source. The input surface, refered to above, is an image where the greytone value of each pixel represents the elevation at the center of each surface element. An elevation (or terrain) model is defined as an image in which each pixel value is the elevation of the center of the surface element represented by that pixel.

An example of the use of an illumination model and an elevation model follows. Suppose there is a surface and an image of that surface illuminated by some wavelength of light when the source of illumination has an elevation angle of E1 and an azimuth angle A1 with an intensity of I1. Suppose it is desired to have an image of that surface illuminated by light of the same wavelength from a source with elevation angle E2, azimuth angle A2, and intensity I2. This new image might be used to detect the effects due solely to the position and intensity of the illumination source on an image of that surface. An elevation model can be used to represent the surface. An illumination model can be used to produce this new image given the elevation model and the parameter values E2, A2, and I2. The new image would be a synthetically produced view of the approximated

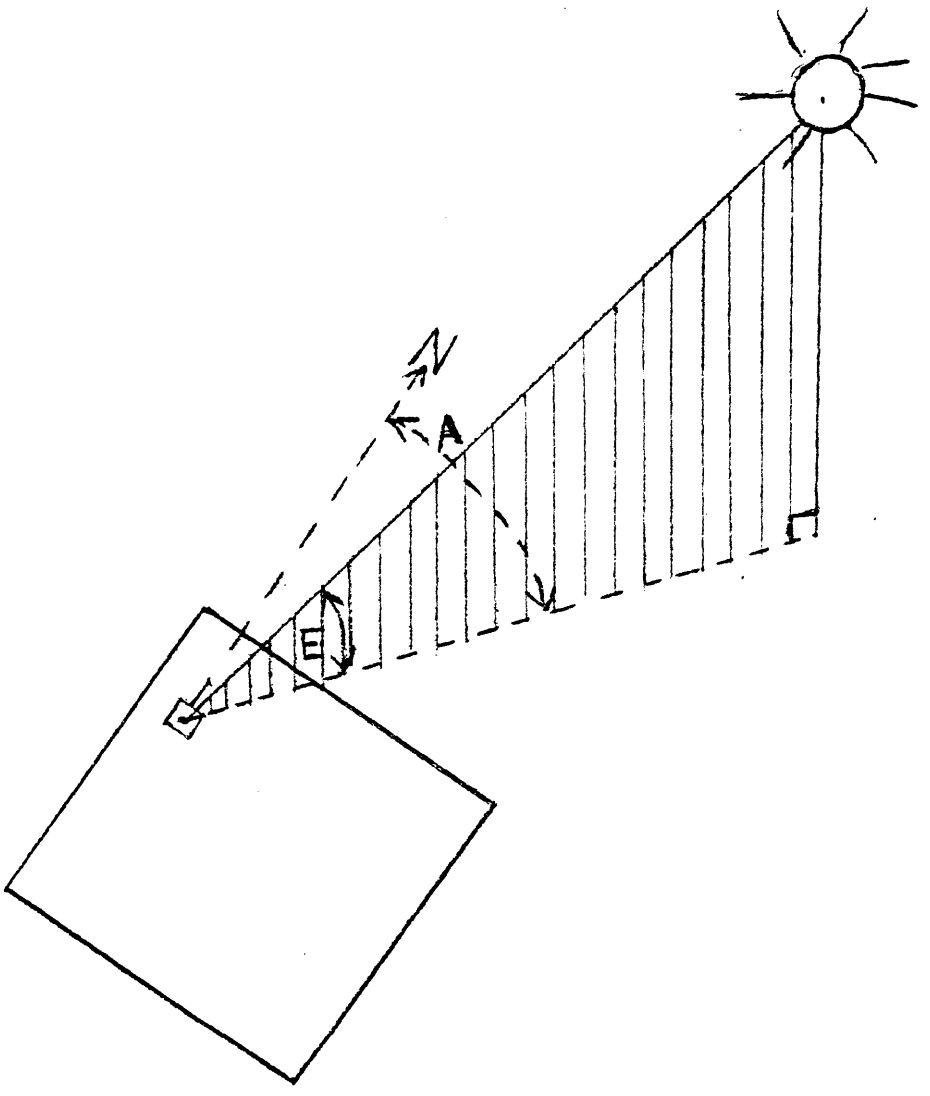


Figure 1:
An illumination source and the angles that define its position. N is true north. E is the elevation angle. A is the azimuth angle.

surface illuminated with the same wavelength of light, but under different conditions.

This thesis describes some of the work that was involved in developing an illumination model and using it to help explain many of the features observed in a LANDSAT image of a mountainous area, especially the shadowing effects. Such a model is a useful tool for distinguishing the features observed in remotely sensed data due to the albedo of the surface material from those features formed by shadowing or shading due to the position of the sun. Chapter II contains a brief review of literature which was helpful in providing ideas, terminology, and a basis for this work. The illumination model including assumptions made and parameters used is defined in Chapter III. Chapter IV tells how the model was implemented on a computer to produce model images that can be compared to real LANDSAT scenes. The application of this illumination model to a particular LANDSAT scene is described in Chapters V and VI. Chapter VII presents some conclusions for this work including an evaluation of this illumination model. The appendices contain the documentation of the GIPSY commands mentioned throughout the thesis and a run file showing how the commands were used in this work.

Chapter II

REVIEW OF RELATED LITERATURE

Since LANDSAT 1 (then called ERTS-1) was launched on 23 July 1972, there has been a great deal of research using the data it produced. The images recorded by satellite have been used to find faults [Campbell, 1976], locate areas of possible shallow and deep ground water [Lucchitta, 1975], measure the motion of glaciers [Post, et al., 1976], identify potential oil-rich lands [Venkataramanan, 1979], observe land changes due to strip-mining [Anderson and Schubert, 1979], detect sources of air pollution [Brown and Karn, 1979], select road alignments [Krinsley, 1979], and for land-use planning [Haralick and Shanmugan, 1973; Welch, et al., 1979; Anderson, 1979]. Computers have been used to compare multiple images of the same area which have significant differences due to different sun positions and various atmospheric variations [Kim and Smith, 1979; Kriebel, 1976; Potter, 1977]. Many techniques for noise removal and other cosmetic procedures and enhancements have been developed for use with satellite remote sensing data as well [Viollier and Baussart, 1979; Billingsley, et al., 1975; Billingsley and Goetz, 1973].

There is work recorded in the literature that relates more directly to this research. It has been reported that certain atmospheric factors [Dave, 1979] and the elevation angle of the sun [Kitcho, 1979] have significant effects on the quality and detail observed in LANDSAT imagery. Anuta [Anuta, 1976] has done work to register topographic data to LANDSAT images in order to enable more accurate and productive analysis of areas with significant topographic relief. Additionally, illumination models have been used in the production of computer generated pictures where it has been found that shading effects are very important in producing realistic images [Phong, 1975].

In [Haralick, 1980], Haralick presents a model to analyze the effects of sun position, camera position, and shadowing on the observed reflectance of a row crop. The illumination model described in this thesis is based in part on that work. Through early experiments in the research described in this thesis, it was determined that at least three components are required to adequately model the effects observed in the particular LANDSAT image to which this illumination model was applied. The idea that there are three possible sources of radiation incident upon some surface in mountainous terrain (direct solar radiance, sky radiance, and adjacent slope radiance) is also stated in

[Kimes and Kirchner, 1981]. The assumption that the scattering of the radiation reflected from the earth's surface is Lambertian is made in the model of this thesis as well as in [Haralick, 1980]. In [Smith, et al., 1980], an analysis of this assumption is made for a specific study area in Colorada. The Lambertian assumption will be discussed in Chapter III.

Other papers which have been useful in providing background information, terminology, and an understanding of the problems encountered in this research, but not noted already, include [Temps and Coulson, 1977] [Horn and Bachman, 1978] [Mitchell, 1978] [Horn, 1977].

Chapter III

THE ILLUMINATION MODEL

This chapter describes the components of the real world that were considered in the formulation of the illumination model. Because of the complexity of the real world, assumptions have been made in modeling it. Some of these are discussed in the first section, others are mentioned in the following sections as the physical phenomena with that they are associated are discussed. The second section describes the sources of radiation which reach a surface and how the complexity of the terrain is involved in that process. The geometry of the terrain also has an effect on the brightness of a surface as it is observed from above. This will be explained in the last section of this chapter.

3.1 BASIC MODEL ASSUMPTIONS

A very common assumption that is made in the literature when modeling the imaging process is that each of the elements of the surface being observed obeys Lambert's cosine law. That is, each element of the surface "appears equally bright when observed from any direction" [Van Nostrands, 1976]. Smith, Lin, and Ranson [Smith, et al., 1980] discuss the validity of this assumpton for a single

mountainous area in Colorado using empirical data and a model proposed by Minnaert [Minnaert, 1941]. The data used were pixels selected from a LANDSAT image in direct sunlight, densely covered with ponderosa pine, on slopes with a primarily northerly aspect. Their conclusions stated that the Lambertian assumption is not strictly valid for the data chosen. However, they stated that for a restricted range of effective incident angles, the assumption is more nearly valid. An effective incidence angle for a surface element is the angle between the normal to that surface element and the ray of sunlight illuminating it. In the LANDSAT image to which the illumination model in this thesis is applied, the position of the sun is such that the pixels which are on the lit side of mountains have effective incidence angles within the restricted range discussed in [Smith, et al., 1980]. Thus, the Lambertian assumption is made for the illumination model in this thesis as it is in [Haralick, 1980], [Horn and Bachman, 1978], and others.

The illumination model developed in this work is also based on a distant, point source. Since the sun is the illumination source for LANDSAT imagery, this assumption seems reasonable. This distant, point source or "parallel ray" assumption implies that all surface elements with the same normal vectors have the same effective incidence angles.

A further assumption is that the view angle of images (both produced by LANDSAT and synthetically by the model) is vertical, i.e. a top view. This is an appropriate assumption for the LANDSAT imagery since the angle that it varies from vertical is only a few degrees. The assumption is made for the images produced by the model so that no surface elements are hidden by other surface elements because of the view angle.

3.2 SOURCES OF INCIDENT RADIATION

Incident radiation is the energy (light) that falls upon a given element of a surface. In complex terrain, there are three possible sources of incident radiation 1) direct solar radiance, 2) sky radiance, and 3) adjacent slope radiance [Kimes and Kirchner, 1981]. Through comparison of illumination model produced images with actual LANDSAT data, it was determined that all three sources are required to adequately model the particular LANDSAT image to which the illumination model was applied. In the illumination model, the contributions of each of these sources is summed to approximate the observed intensity of reflected light for each pixel. That is,

$$\text{pixel value} = R * (DS + D + B),$$

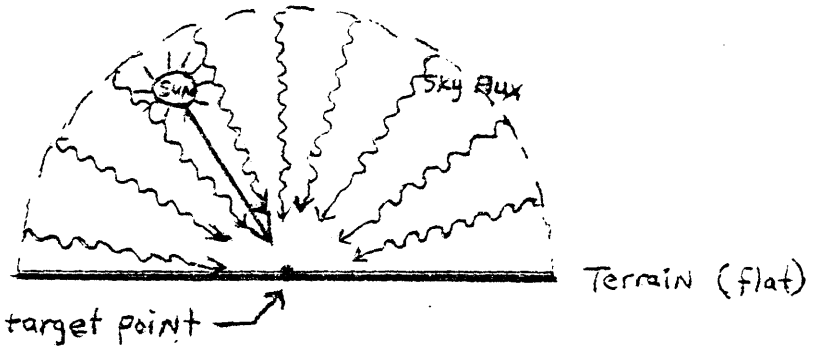
where R is the reflectivity value of the surface element represented by the pixel. DS, D, and B are the intensities of reflected light from direct solar radiance, diffuse radiance, and backlighting, respectively.

Direct solar radiance is the simplest and the most important of the three sources. It consists of the rays of direct sunlight (not reflected) that illuminate an element of a surface. In the model in this work, no loss to atmospheric scattering or absorption is assumed. That is, it is assumed that rays from the sun with intensity I move through the atmosphere unaffected and illuminate each pixel that is not in shadow. The brightness or intensity of a pixel illuminated with direct radiance alone is computed according to Lambert's cosine rule:

direct radiance intensity of pixel = DS * r * cos(i) ,

where r is the reflectivity value of the element of the surface and i is the effective incidence angle. Direct solar radiation accounts for about 80% to 90% of the observed intensity of a pixel oriented toward the sun.

Sky radiance, otherwise known as diffuse radiation, is a rather small part of the total incident radiation on a surface element. Diffuse radiation is direct solar radiation that has been reflected by some surface element and illuminates the surface after being scattered one or



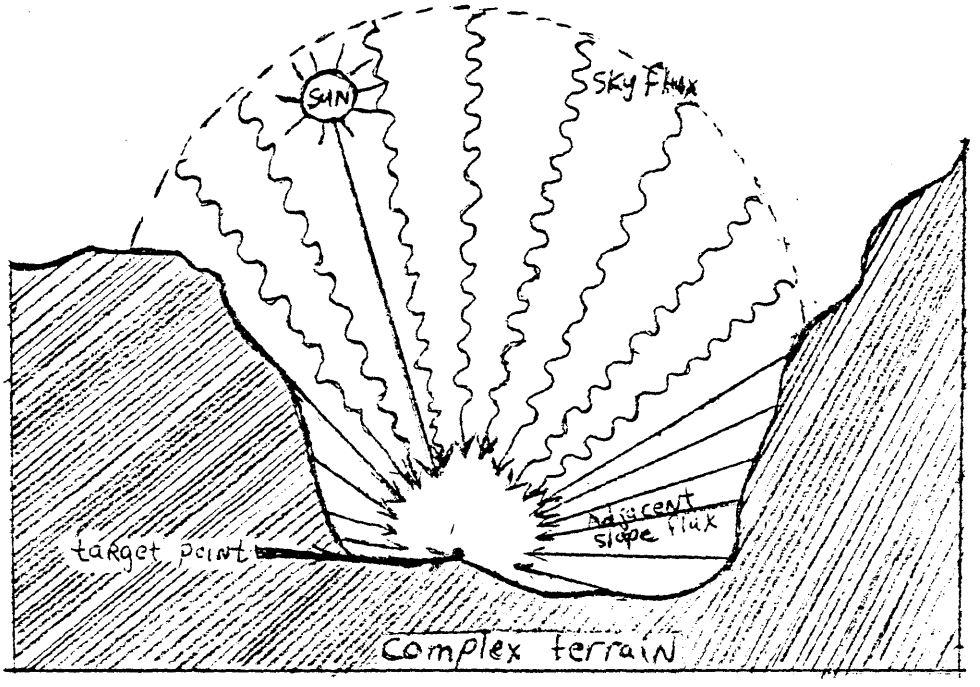


Figure 2:
Three sources of incident radiation direct solar radiance, sky radiance (diffuse light), and adjacent slope radiance (backlighting).

more times by the atmosphere [Temps and Coulson, 1976]. The diffuse component is about 14 percent of the total flux on a surface normal to the solar beam, and almost 100 percent of that on a surface shaded from the direct solar beam [Temps and Coulson, 1976]. Though diffuse light is not uniformly distributed over the dome of the sky, a common assumption made in this model is that the sky radiance incident on a surface element can be approximated by a constant.

Adjacent slope radiance is reflected radiation from surrounding terrain that illuminates a surface element. This source of radiance is only significant in mountainous terrain. Obviously, calculation of all adjacent slope radiances incident upon each surface element in an image containing complex terrain would be very time consuming. To model this phenomonon, a secondary point illumination source having an azimuth 180 degrees different from the primary source, but with a much smaller intensity is applied to the surface. The sum of the effects of these two "suns" is used to approximate the backlighting effects of adjacent slope radiance in the valleys observed in the LANDSAT image to which this model was applied.

3.3 GEOMETRY AND REFLECTANCE

The geometry or shape of the surface being observed has a great bearing on the reflected light intensities represented in an image. In fact, in LANDSAT images intensity fluctuations due to varying surface gradients often swamp the intensity fluctuations due to variations in surface cover (different reflectivity values of different materials). [Horn and Bachman, 1978].

The effects of adjacent slopes have previously been mentioned. The two most important ways in which surface geometry can affect image intensities are due to Lambert's cosine law and shadowing.

Lambert's cosine law has already been mentioned in reference to the effects of direct solar radiance. Here, the part of that relation of most importance is the incidence angle i. The incidence angle is the angle between a ray of the sun incident on an element of the surface and the vector normal to the element of the surface. Since the intensity of all the rays of the sun are the same, the intensity of a pixel in an image is largely determined by the orientation of that element. A pixel oriented normal to the incident ray will appear brightest. The same pixel would appear less bright if oriented differently.

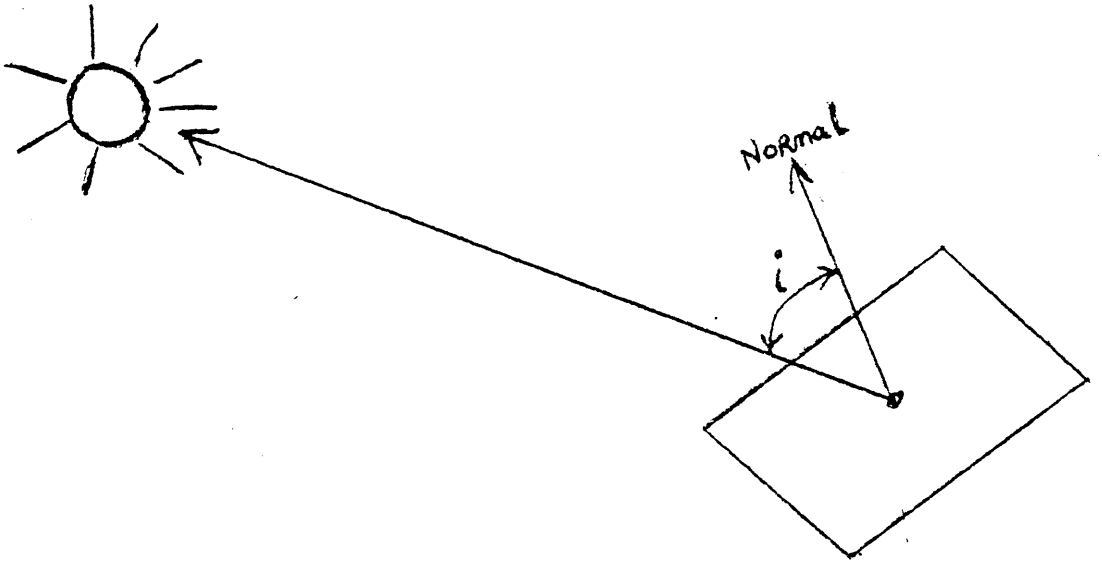


Figure 3:
The incidence angle, i, is the angle between a ray of the illumination source and the vector normal to the illuminated surface.

Shadowing occurs when some feature of the surface blocks the direct solar radiance for some surface element. The geometry of the surface and the position of the sun are the factors that determine the shadowing in an image. A pixel in shadow need not be completely black (intensity value of Ø). In this illumination model, diffuse radiance and adjacent slope radiance may also be incident upon that pixel.

The reflectivity value r in Lambert's cosine relation is a ratio of radiation reflected to radiation incident on a surface element. There are different reflectivity values for different materials and for different wavelengths of energy. For instance, in band 7 (near-infrared) of the LANDSAT MSS images, green vegatation reflects very brightly. Water bodies, on the other hand, appear very dark in band 7. The illumination model allows for a reflectivity parameter. In initial experiments, however, constant reflectivity value is assumed for all materials.

Chapter IV

IMPLEMENTATION OF THE MODEL

This chapter describes the computer programs that implement the illumination model of Chapter III. The programs are commands in the GIPSY image processing system [Krusemark, 1980] and are written in RATFOR. Two commands are used to implement the illumination model. They are described in the first and second sections that follow. The third section discusses how these same commands are used in the implementation of backlighting effects.

4.1 TERRAIN SHADOWING EFFECTS

SHADOW is the GIPSY command that marks pixels of the image which will appear in shadow due to interactions between the geometry of the surface and the position of the illumination source. The input to this program is an elevation model and the parameters that define the position of the illumination source and a scaling factor for the elevation model values. The output of the program is a binary image the same size as the input file (elevation image) with shadowed pixels marked '0' and all other pixels marked '1'.

4.1.1 Input Data

The elevation model (input file) for the surface for which a synthetically illuminated image is to be produced must be an SIF (standard image format, a GIPSY convention) file in integer mode and row format. That is, one row of the image per record, and the record contains an integer value representing the elevation at the center of each pixel in that row. The ground distance multiplier is a parameter the user is asked for and is used to make the units in the vertical direction (elevation values) the same as those in the horizonal plane (row and column directions). The widths of a row and column are assumed to be equal, i.e. a pixel is assumed to represent a square area of the surface that the elevation model image represents. The ground distance multiplier is a floating point (real) value.

Three other input parameters are requested from the user by the SHADOW program. All three define the position of the illumination source and are floating point values. The elevation angle of the source is measured in degrees up from the horizontal and the azimuth angle of the source is measured in degrees clockwise from true north. Since it is not always the case that the top of the elevation model image is oriented due north, the third parameter is required. The user enters the number of degrees that the

image is skewed to the right. Again, this angle is measured in degrees clockwise from north. (It is common for LANDSAT images to be skewed 12 degrees.) These three parameters define the position of the illumination source relative to the elevation model image.

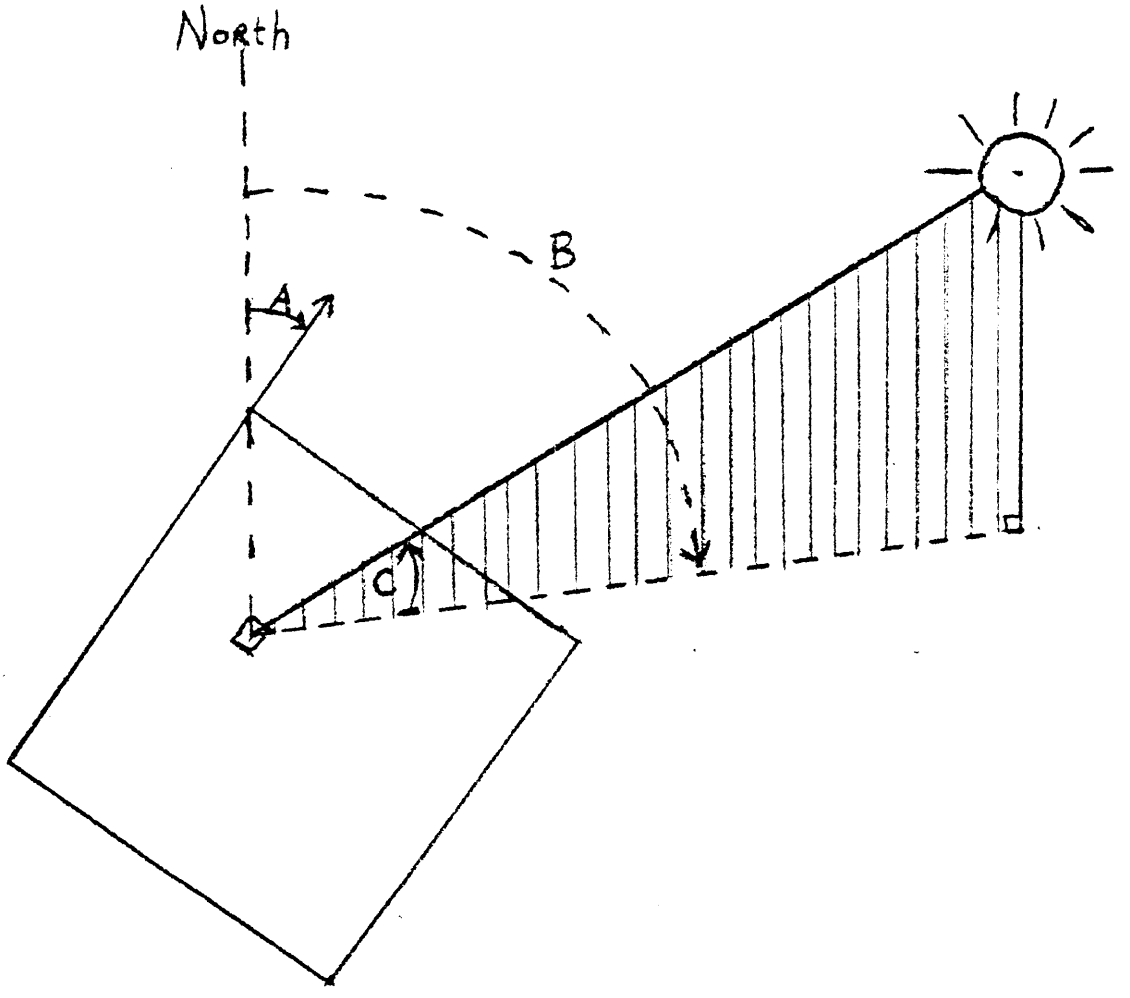


Figure 4:
A diagram showing the image orientation angle, A, the azimuth angle of the source, B, and the elevation angle of the source, C.

4.1.2 Implementation Details

The SHADOW program works by creating an output image and examining each pixel to determine if the direct solar radiance is blocked from that pixel. The pixels in the output image have a one-to-one correspondence with the pixels in the input elevation image.

A ray from the illumination source is assumed to touch the center of each pixel at an angle with the horizontal equal to the elevation angle input by the user. Therefore, the height of that ray of light can be calculated above the other surface elements (pixels) along the azimuth from the pixel whose condition (in shadow or not) is to be determined to the edge of the image. If the elevation of any of these surface elements along the azimuth specified as a parameter is greater than the height of the ray, then the pixel is in shadow.

Doing the procedure for every pixel in an image can be quite time consuming. This procedure also requires a lot of I/O (input/output) operations since a large image cannot generally be stored in internal memory at once. To reduce some of the time and I/O operations, a few implementation details can be considered.

First, there is a maximum integer pixel elevation value that can occur in the elevation model image. This value is

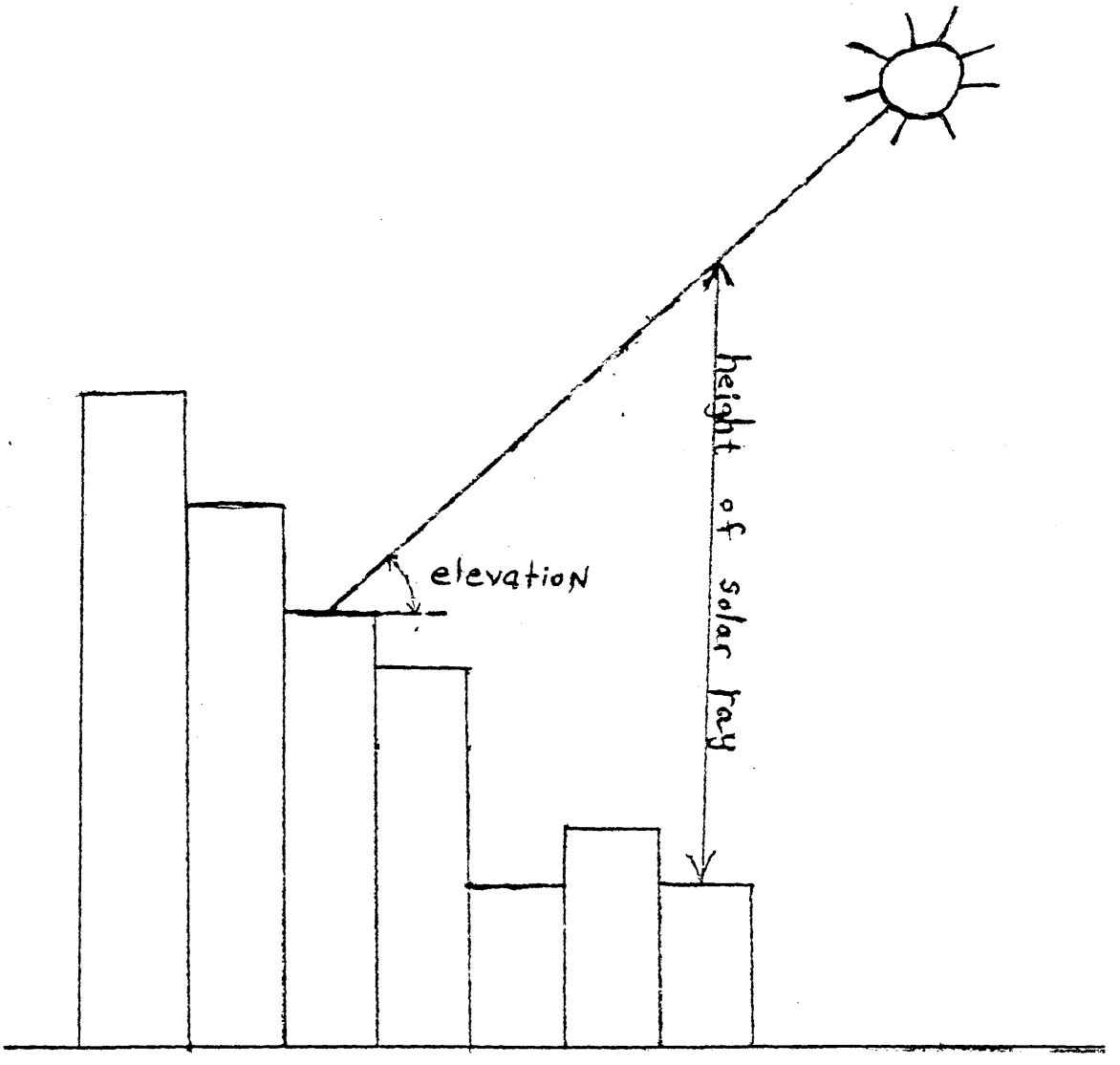


Figure 5:
A diagram showing the path of a ray of light from the source to a particular surface element that is illuminated.

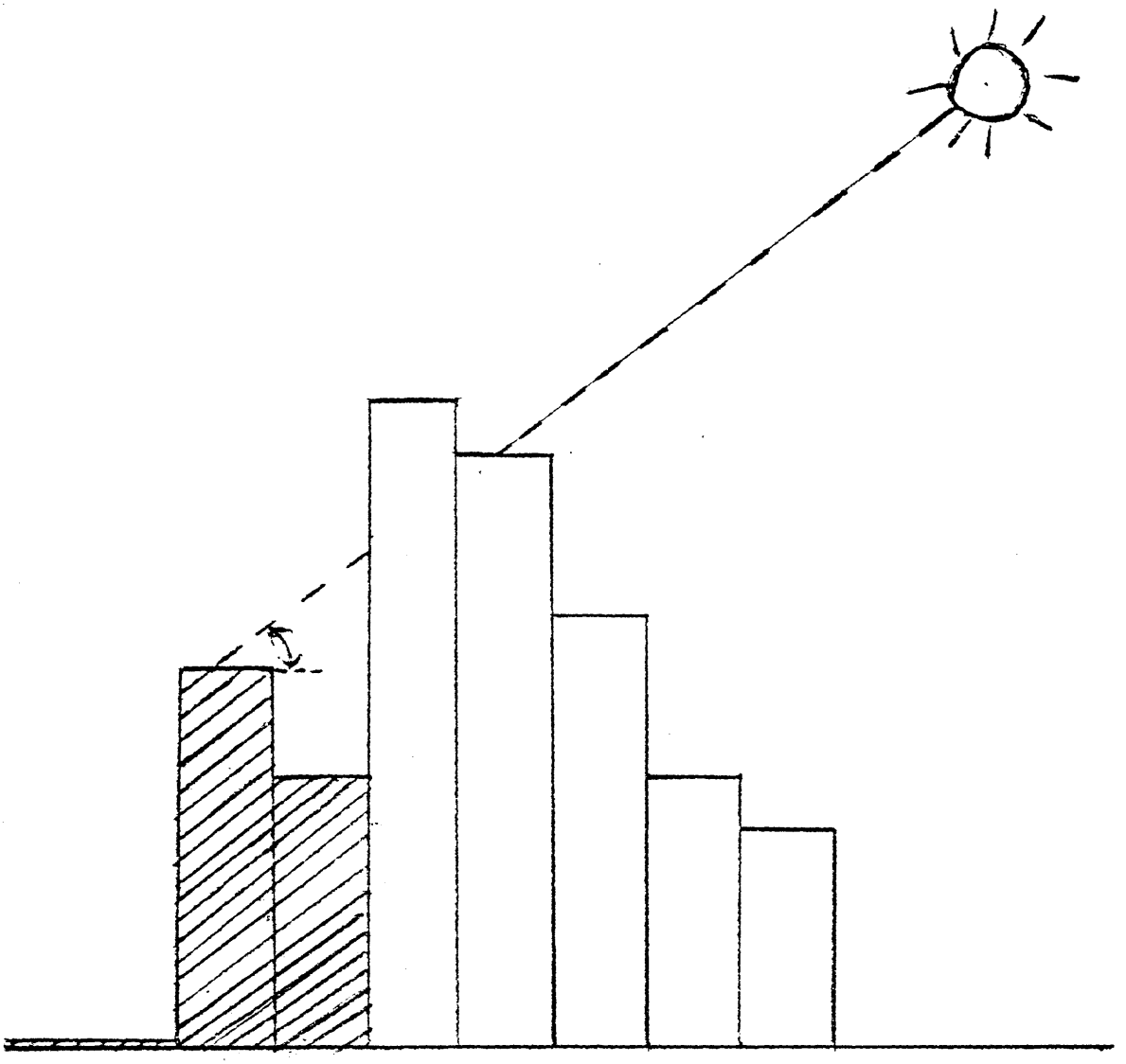


Figure 6:
A diagram showing a surface element in shadow.

recorded in the header record for that SIF file. There is no need to examine pixels over which the ray of interest has exceeded that maximum elevation. Eliminating the processing of pixels in this way will reduce computation time.

To reduce the amount of I/O operations, a buffer can be provided into which as much of the elevation image will be stored as internal storage space allows. By ordering the operations so that all references to a row in the buffer are made before that row is over-written, even more I/O operations can be saved. All of these measures are incorporated into the computer code of the SHADOW command.

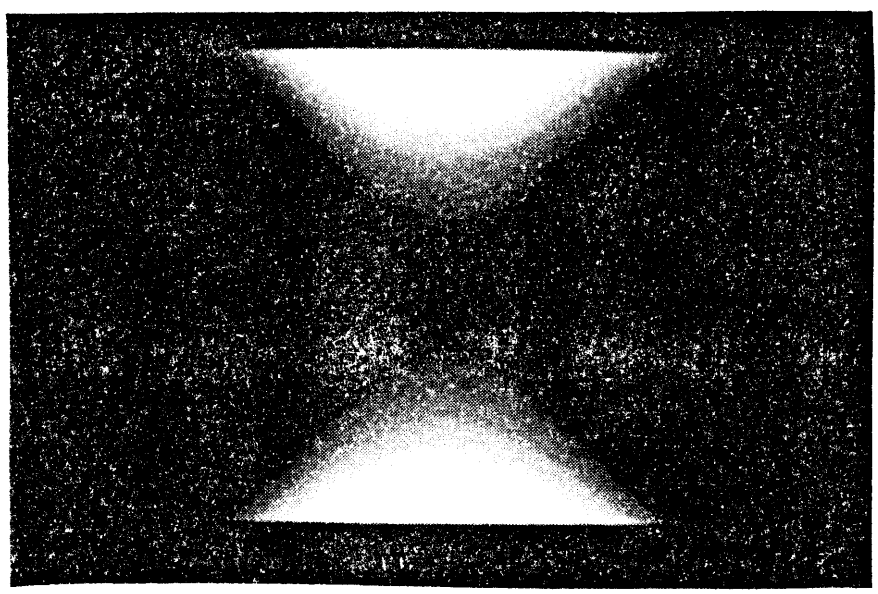


Figure 7:
A greytone representation of a hyperbolic paraboloid. The bright values represent large values in the height direction. This image is stored in a file named 'HYPPAR.SUR' and was created by the SURFAC command.

```
SHADOW HYPPAR.SUR > HYPPAR.SUN

    Enter orientation of top of image :  Ø.
    Enter azimuth of sun              : 9Ø.
    Enter ground distance multiplier  :  1.
    Enter elevation angle of sun      : 45.
```

Figure 8:
An example showing the use of the SHADOW command.

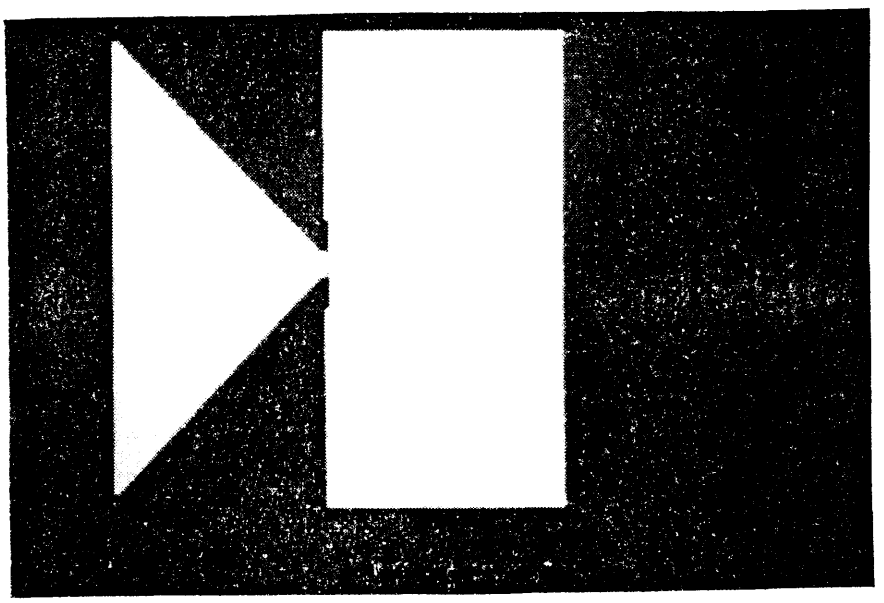


Figure 9:
The binary output image from the SHADOW command in the previous figure. The black represents pixels in shadow, the white represents lit pixels.

4.2 INCIDENT RADIATION AND REFLECTIVITY

The first step in the implementation of the illumination model presented in the previous section considers only the relative elevation and location of each surface element (pixel) and the position of the illumination source to mark pixels that are in shadow. The second step, described in this section, uses the information from the first step and additionally considers the 3-D orientation of each surface element and the strength of the radiation source to model the various intensities of reflected light that is observed when a surface is illuminated. The GIPSY command SHADE performs this second step. This program takes two or three SIF files as input as well as input parameters. These will be described below. The output of the program is an image that represents the intensities of reflected light for the surface to which the illumination model is being applied. The output and the calculations that produce it are described later in this chapter.

4.2.1 Input Data

The first input file is the SIF file produced by the SHADOW command in step one. This image indicates for each pixel whether it is in shadow or direct illumination. The second input file provides the information necessary to

define the 3-D orientation of each surface element in the scene being modeled. This information is represented by two values for each pixel. One value, ALPHA, is the slope of the pixel in the row direction. The other value, BETA, is the slope of the pixel in the column direction. There is a GIPSY command named CNSFCT which computes these values for a given elevation model of a surface. SHADE accepts the multiband (more than one value per pixel) output of the CNSFCT command as its second input file.

Both of the two input files of the SHADE command just described are required. A third input file must be provided only if the 'R' user flag is set in the command line invoking SHADE, (setting user flags is a mechanism used in the GIPSY system to select optional features of a command). The R-flag in the SHADE command allows the user to specify the albedo (reflectivity value) of each pixel in the image. The albedo is represented by an integer percentage (Ø-1ØØ) of the incident light of a given wavelength that is reflected by the pixel. The third input file to the SHADE command contains these values for the wavelength of light being modeled for each pixel in the image. If the R-flag is not set, no third input file is used and it is assumed that for all pixels 1ØØ percent of the incident light is reflected.

```
CNSFCT  HYPPAR.SUR > HYPPAR.SFT

     Enter window size for
      row by column          : 3 3
     Enter number of bits    :  16
```

Figure 1Ø:
An example using the CNSFCT command.

The SHADE command has five input parameters for which the user is prompted. Two of these parameters, the elevation angle of the source and the ground distance multiplier, are exactly as they are defined in the discussion of the SHADOW command in a previous section. A third parameter, the corrected azimuth angle of the source, is the sum of two other parameters described for the SHADOW command: the azimuth angle of the source and the skew angle of the image.

Two other input parameters of the SHADE command are source intensity and the diffusion ratio. The source intensity is the measure, in any units the user desires, of the intensity of the radiation from the source that reaches the surface. The source intensity parameter is an integer (fixed point) value.

The diffusion ratio is defined as the amount of diffuse radiation (sky light), in the same units as the source intensity defined above, divided by the total source intensity. It is a dimensionless (unitless) fraction of the total radiation that is to be used to represent the constant diffuse radiance defined in Chapter III. The five parameters and the input files provide the data needed to produce the output image described below.

4.2.2 Calculations to Produce the Output

The output file for the SHADE command is a single band (one value per pixel) image the same size as the input images. At each pixel, the intensity of the observed (reflected) light of a given wavelength is represented by a value in the same units as the source intensity input parameter. The calculation of this value is different for pixels in shadow than it is for pixels that are directly illuminated.

This program uses a three dimensional vector space Rows x Columns x Height (elevation) to define the relative positions of the surface elements and the illumination source. A vector that points in the direction of the illumination source from the center of a pixel is computed. Note that since the illumination model assumes parallel rays radiate from the distant, point illumination source, the same vector can be used to point to the source from all pixels. This vector can be written as the ordered triple:

[-SIN(90-A)*SIN(90-E), -COS(90-A)*SIN(90-E), COS(90-E)],

where A is the value of the azimuth angle of source parameter and E is the value of the elevation angle of source parameter.

A vector normal to the surface element represented by each pixel can be calculated also. It can be written as the ordered triple:

(-Row Slope , -Column Slope , 1),

where Row Slope and Column Slope are the values ALPHA and BETA, respectively taken from the second input file and scaled using the ground distance multiplier parameter. The cosine of the angle of incidence of a ray of direct radiation with a pixel can be computed for each pixel by taking the dot product of the vector normal to pixel and the vector pointing to the illumination source.

The intensity value in the output image for each pixel not in shadow is:

$$R * (COS(i) * I + D).$$

For those pixels in shadow the output intensity value is: R*D. In both relations, R is the reflectivity value of that pixel / 100, i is the incidence angle for that pixel defined in Chapter III, I is the source intensity parameter value, and D is the amount of diffuse radiation given by the diffusion ratio parameter value * the source intensity parameter value.

All the output intensity values are truncated to an integer value.

```
SHADE HYPPAR.SUN, HYPPAR.SFT > HYPPAR.SHD

     Enter corrected sun azimuth      : 90.
     Enter sun elevation angle        : 45.
     Enter diffusion ratio            : .10
     Enter sun intensity              : 100
     Enter ground distance multiplier : 1.0
```

Figure 11:
An example using the SHADE command.

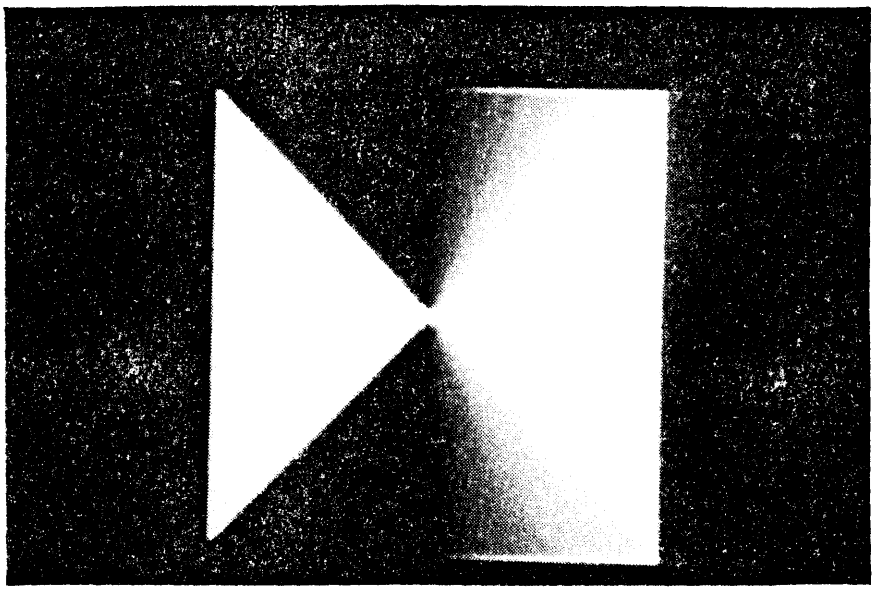


Figure 12:
A greytone representation of the output image from the example in the previous figure.

4.3 IMPLEMENTING BACKLIGHTING

Backlighting, or adjacent slope radiance as it is described in Chapter III, is most significant in mountainous terrain with steep slopes and deep valleys. To implement the model of backlighting given in Chapter III requires more than one application of the SHADOW and SHADE commands and one other GIPSY command as explained below.

To produce an image that models a surface being illuminated by two or more sources, the commands SHADOW and SHADE must be run for each source separately. The input files will be the same for each separate run. The input parameters which are used to define the position and intensity of the source in both SHADOW and SHADE will vary with the different sources being modeled. To produce a single image that models the effects of all these sources at the same time, the results of each of these runs must be summed.

There is a GIPSY command named ARITHM that makes summing the effects of two or more illumination sources simple. ARITHM performs a binary arithmetic operation (selected by user flag) on two input images of the same size and produces an output image as the result. To add two images together, the A-flag is set in the command line calling ARITHM. The greytone value of each pixel from the first input image is

added to the greytone value of the corresponding pixel of the second input image to produce the value at each pixel of the output image.

The backlighting model in Chapter III requires two illumination sources. SHADOW and SHADE are run for the primary source, perhaps the sun, as described earlier in this chapter. Then SHADOW and SHADE are run using the azimuth angle of the primary source + 180 degrees as the azimuth angle of the source and a smaller source intensity. The output image from the two SHADE runs are taken as the two input images to the ARITHM command. The output image from ARITHM is the illumination of the surface including backlighting effects.

```
SHADOW HYPPAR.SUR > HYPPAR.SN1
     Enter orientation of top of image :  Ø.
     Enter azimuth of sun              : 9Ø.
     Enter ground distance multiplier  :  1.
     Enter elevation angle of sun      : 45.


SHADE HYPPAR.SN1, HYPPAR.SFT > HYPPAR.SD1
     Enter corrected sun azimuth       :  9Ø.
     Enter sun elevation angle         :  45.
     Enter diffusion ratio             :  .1Ø
     Enter sun intensity               :  1ØØ
     Enter ground distance multiplier  :  1.Ø


SHADOW HYPPAR.SUR > HYPPAR.SN2
     Enter orientation of top of image :  Ø.
     Enter azimuth of sun              : 27Ø
     Enter ground distance multiplier  :  1.
     Enter elevation angle of sun      : 45.


SHADE HYPPAR.SN2, HYPPAR.SFT > HYPPAR.SD2
     Enter corrected sun azimuth       :  27Ø
     Enter sun elevation angle         :  45.
     Enter diffusion ratio             :  .1Ø
     Enter sun intensity               :   1Ø
     Enter ground distance multiplier  :  1.Ø


ARITHM HYPPAR.SD1, HYPPAR.SD2 > HYPPAR.LIT (A)
     Decrease number of bits ?   :  N
```

Figure 13:
An example using the SHADE and SHADOW command two times to implement backlighting.

Chapter V

APPLICATION TO A LANDSAT SCENE

The purpose of the work described in this thesis was to develop an illumination model and to apply that model to a LANDSAT image of a mountainous area to separate the effects of shadowing from other phenomenon. The illumination model and its implementation have been discussed in the previous chapters. This chapter describes a particular LANDSAT image and the area it represents in the first section. Then, the elevation (or terrain) model that is used to represent the topography of that area is described. Finally, the results of applying these models to this LANDSAT area are discussed.

5.1 DESCRIPTION OF STUDY AREA

The LANDSAT image to be examined in this work was produced 13 April 1975 by the LANDSAT-I MSS (Multi Spectral Scanner) (scene id: 5360-14502; path 18, row 34). This scene represents an 185 km x 185 km area appearing on the Charleston, West Virginia/Ohio USGS 1:250,000 quadrangle (NJ 17-5). A small segment of this large image was chosen for study. It is located in southeastern West Virginia and includes an area covered by the Ansted, West Virginia and Lockwood, West Virginia 7.5 minute quadrangle topographic

maps. The size of the study area is about 6.3 km x 4.4 km or 80 x 80 pixels in the LANDSAT image.

The topography of this area is very complex. Valleys are narrow with steep sides. For example, the Gauley River, visible at points on our LANDSAT image, follows a valley that is typically 150 meters deep but only 100 meters wide. Uplands often consist of only ridge crests; although plateau-like upland regions are present, they are not continuous or extensive. The relief in the area (difference in elevation detween hilltops and surrounding lowlands) averages about 600 feet. The area receives 40 inches of precipitation each year and is forested with a dense cover of deciduous trees. Most of the trees are without leaves at the time the LANDSAT image was made, especially in the upper elevations. The valleys have some newly emerged leaves and grasses [Wang, et al, 1982].

The LANDSAT image shows the study area illuminated from the southeast (azimuth = 119 degrees) at an elevation of 45 degrees above the horizon. Band 7 of the MSS data which records reflected energy with wavelengths .8 - 1.1 μm is used in this work since shadows appear dark and clearly defined and there is little evidence of scattering or atmospheric degradation of the data in that band in this image. A photograph of the LANDSAT image of the study area is shown in Figure 15.

Figure 14:
A topographic map of the study area.

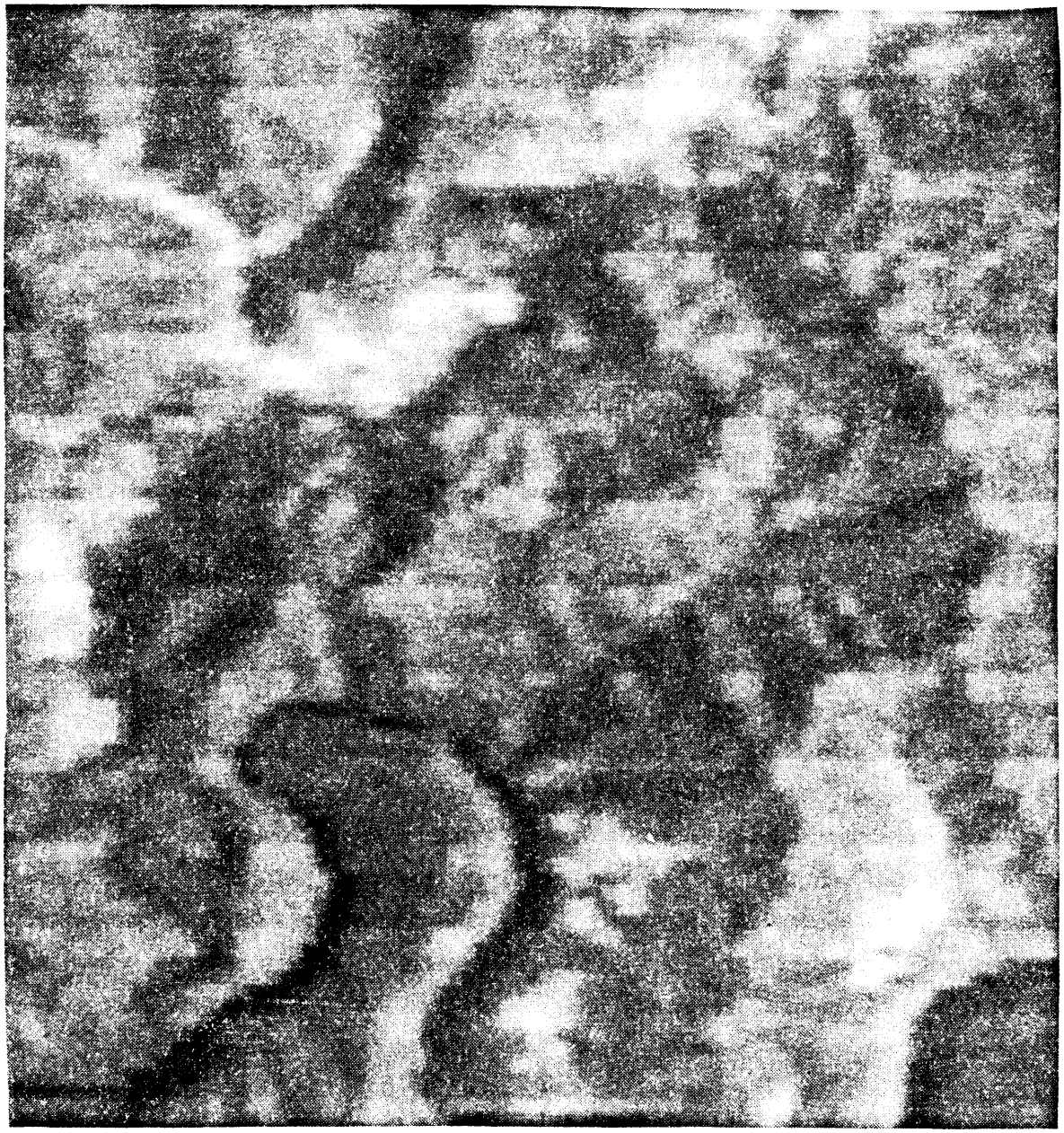


Figure 15:
A picture of band 7 of the LANDSAT image of the study area.

5.2 CREATING A DIGITAL ELEVATION MODEL

The greytone values from a band of the MSS data in a LANDSAT image represent the intensity of some wavelength of reflected energy, but not elevation data about the area being imaged. In order to simulate the shadowing effects caused by the sun illuminating a mountainous area, an elevation model that represents the topography of that area must be computed. A common method for obtaining an elevation model for an area imaged by satellite is to digitize a topographic map of the area and register the image to the digital elevation model. Rather than use a topographic map, in this research the digital elevation model is constructed using only the LANDSAT data. The following sub-sections describe how elevation information is derived from the LANDSAT image, experiments with different elevation models and various implementations, as well as the resulting elevation model images.

5.2.1 Finding Ridges and Valleys

Since the position of the sun at the time the LANDSAT image was created is known and it can be assumed that some of the contrast between bright and dark regions in band 7 image of the LANDSAT MSS data is due to shadowing in this mountainous area, it is possible to identify many pixels in

the image as being located on a ridge or in a valley. A segmentation technique to locate valley and ridge pixels comes from the work of S. Wang [Wang, et al, 1982]. A brief summary of the process follows.

The band 7 LANDSAT image is segmented into regions by greytone value. Each region consists of adjacent pixels with greytone values in the same range. The boundaries (pixels on the outside edge of a region) are examined to determine if they may be valleys or ridges. This determination is made by examining the greylevel values of the regions on which the boundaries border and considering the position of the illumination source. For example, when the sun is in the east, a boundary that is dark on the left and bright on the right is labeled as a ridge. The reverse (dark on right and bright on left) will be labeled as a valley. Where east-west boundaries exist, they are labeled as valleys unless they are adjacent to ridges on both ends. This technique is used to produce two binary images (ridge and valley maps) one for ridge pixels, the other for valley pixels.

5.2.2 From Ridges and Valleys to Terrain Model

Once ridges and valleys have been identified, it is possible to create a digital terrain (elevation) model for an area. An elevation model as defined in Chapter I is "an image in which each pixel value is the elevation at the center of the surface element represented by that pixel." The method for creating a digital elevation model from the ridge and valley maps is described here. An image the same size as the portion of the LANDSAT scene representing the study area is created with pixels corresponding to the valley pixels from the valley map given a value of Ø and pixels corresponding to the ridge pixels in the ridge map being given a value of 1ØØ. These pixel values will remain fixed. The next step is to assign values to all the pixels in the elevation model not marked as ridge or valley. To do this, some surface is calculated which has the function value of Ø or 1ØØ at the center of the surface element represented by the valley and ridge pixels, respectively, and each "variable pixel" in the elevation model image is assigned the function value at the center of the surface element it represents. Since an infinite number of interpolation surfaces can be generated for a given set of ridge and valley pixels, only surfaces with certain properties are to be considered. Two such surfaces are

described in the next sub-section. It should also be noted that since all the ridge pixels have the same elevation value, as do all the valley pixels, the elevation model created by this method should be called a "relative elevation (terrain) model."

5.2.3 Interpolating Surfaces

In this research so far, two different surfaces have been used in creating an elevation model. That is, two different conditions have been placed on the interpolating surface in addition to the constraint that the ridge and valley pixels values are given as constants (boundary values). The first condition is that the interpolating surface must have the given boundary values and have a Laplacian of zero at all non-boundary value points. This will be referred to as the Laplacian surface. The second condition that was used is that the interpolating surface must have the given boundary values and minimize the quadratic variation function. This will be referred to as the quadratic variation surface. In this section, these two surfaces are defined more completely. A later sub-section describes various techniques for computing them.

The elevation models given by both of these surfaces can be computed by solving a system of linear equations. The system for the Laplacian surface is written as:

$$A x = b.$$

The vector x is the solution to this linear system and represents the values to be assigned to each "variable" (non-boundary value) pixel in the elevation model. The A matrix is defined by applying the digital Laplacian mask operator to each variable pixel. A mask operator is applied to a pixel by placing the mask over the image so that the central (large positive) mask value is directly over the pixel whose value is to be computed. The pixel value is changed to make the sum of the mask values times the corresponding image values under them equal to zero.

For the Laplacian surface only, Neumann boundary conditions are enforced along the outside rows and columns of the elevation model image. That is, the outer-most row or column is repeated so that the mask operator can be applied to the outside pixels. There is one row in A for each variable pixel in the elevation model and one coefficient value in that row for each variable. A is a sparse matrix since no variable is constrained by more than four other variables (due to the definition of the digital Laplacian mask operator).

The b vector is the right hand side of each of the linear equations in the system. The constants on the left hand side of each equation (that result from applying the

$$\begin{array}{ccc} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{array}$$

Figure 16:
A digital Laplacian mask operator.

Example image			Binary image		
1	2	3	Ø	1	Ø
4	5	6	Ø	Ø	Ø
7	8	9	1	Ø	Ø

Variables and constants (Neumann boundaries)

$$\begin{array}{ccccc} x_1 & x1 & 2 & x2 & x2 \\ x_1 & x1 & 2 & x2 & x2 \\ x_3 & x3 & x4 & x5 & x5 \\ 7 & 7 & x_6 & x7 & x7 \\ 7 & 7 & x_6 & x7 & x7 \end{array}$$

RESULTING LINEAR SYSTEM

A

x_1	x2	x3	x4	x5	x6	x7
2	Ø	-1	Ø	Ø	Ø	Ø
Ø	4	Ø	Ø	-1	Ø	Ø
-1	Ø	3	-1	Ø	Ø	Ø
Ø	Ø	-1	4	-1	-1	Ø
Ø	-1	Ø	-1	3	Ø	-1
Ø	Ø	Ø	-1	Ø	3	-1
Ø	Ø	Ø	Ø	-1	-1	2

x = b

x_1	2
x_2	2
x_3	7
x_4	2
x_5	Ø
x_6	7
x_7	Ø

Figure 17:
A linear system created using the digital Laplacian mask.

Laplacian operator to a variable pixel that has a "known" pixel 4-adjacent to it) are carried to the right hand side and appear in b. For equations representing variable pixels not 4-adjacent to "known" pixels, the corresponding b element is zero.

The system for the quadratic variation surface is written as:

$$Q\,x = b.$$

The x and b vectors have the same meaning as in the Laplacian case and are constructed similarly. The Q matrix is likewise similar to the A matrix of the Laplacian. Instead of using Neumann boundary conditions at the edge of the image, the quadratic variation surface is defined by using special masks to fit the rows and columns near the outside edges. The six masks are rotated as necessary and applied to the only appropriate variable pixels of the elevation image to define Q. Mask two is applied to corner pixels, mask three is applied to pixels in the outside row or column that are adjacent to a corner pixel, mask four is applied to other pixels in the outside rows and columns, mask five is applied to pixels in the next-to-the outside row and columns that are 8-adjacent to corner pixels, mask six is applied to other pixels in the next to the outside rows and columns, and mask 1 is applied to all other variable pixels in the image.

```
          2
      4 -16   4
  2 -16  40 -16   2
      4 -16   4
          2
```

```
          2
         -8
          8  -8   2
```

Figure 18:
Mask one (top) and mask two (bottom)
of the quadratic variation method.

```
      2
   4 -12   4
  -8  20 -12   2
```

```
      2
   4 -12   4
 2 -12  22 -12   2
```

Figure 19:
Mask three (top) and mask four (bottom)
of the quadratic variation method.

```
          2
     4  -16    4
   -12   36  -16    2
     4  -12    4
```

```
          2
     4  -16    4
 2  -16   38  -16    2
     4  -12    4
```

Figure 20:
Mask five (top) and mask six (bottom)
of the quadratic variation method.

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

Figure 21:
An example showing the index of the mask used
for each pixel location in a small 6 row by 7 column image.

These masks were published in a paper by Grimson [Grimson, 1981]. Matrix Q, like matrix A, is a sparse matrix.

5.2.4 Iteration Methods Compared

The linear system of equations from which the elevation model image can be quite large. For example, for an elevation model image with 100 rows and 100 columns and 100 known pixels (corresponding to ridge or valley pixels) the linear system has 9,900 variables and the A and Q matrices have 9,900 x 9,900 or 99,110,000 entries. Iterative techniques are commonly used to solve large linear systems. Three iterative methods were tested and compared for a small linear system with 95 variables, and a larger problem with 4591 variables. The three methods are Gauss-Seidel iteration, Successive Over-Relaxation (SOR) iteration, and a tridiagonal technique to be described later. For each of the iteration methods, there is an iteration formula that describes how to compute new values for all the variables given a previous set of values. The iteration continues until some convergence criterion is met. The convergence criterion used in this work is that the maximum change in values of all variables between two iterations be smaller than a given constant (called epsilon).

In describing the first two iteration techniqus, the following definitions are used:

$$A = D - L - U ,$$

where A is the coefficient matrix for the system (corresponding to the A or Q matrix in our application to the surface interpolation problem). D, L, and U matrices are all the same size as A, and D consists of the diagonal elements of A and zeros elsewhere, L consists of the negative of the strictly lower triangular elements (excludes the diagonal elements) of A and zeros elsewhere, and U consists of the negative of the strictly upper diagonal elements of A and zeros elsewhere. Gauss-Seidel iteration is defined as:

$$x^{(k+1)} = (D-L)^{-1} U x^{(k)} + (D-L)^{-1} b ,$$

where $k >= 0$. $x^{(0)}$ is the initial value of the variables in the linear system and b is the right hand side of the equations in the system. The Gauss-Seidel method is implemented in the GIPSY command INTERP. The command takes as input two images. The first image is the initial value image in which the variables may have any pixel values and the known pixels have their known, constant values. The second input is a binary image the same size as the first input image in which the value of '1' in a pixel location indicates that the corresponding pixel in the first input

image is a known pixel, and hence its value will not be changed. The user defines the mask operator to be used in the iterations and also the stopping criteria (the maximum number of iterations and the epsilon described before). The output is an image, the same size as the input images, with the known pixels having their constant values and all variable pixels having their values from the last iteration performed.

Using the Gauss-Seidel method with an epsilon value of 1.Ø, the small problem took 42 iterations to converge. The maximum change in a variable value between the last two iterations was .995Ø26 with an average change in variable values of .653990. The spectral radius (maximum eigenvalue) of the iteration matrix, $(D-L)^{-1}U$, was calculated to be .95Ø6328 for this small problem. For the larger problem, convergence took 26 iterations, had a maximum change value of .988121, and average change value of .Ø882335 for the same epsilon value. The spectral radius was not calculated because of the size of the linear system and great amount of computation that would be involved in this calculation.

The Successive Over-Relaxation iteration formula is:

$$x^{(k+1)} = (D-wL)^{-1} \{(1-w)D+wU\} x^{(k)} + w(D-wL)^{-1} b ,$$

where $k >= Ø$ and $Ø < w < 2$ is the relaxation factor. D, U, L, x, and b are defined the same as for the Gauss-Seidel case. Another way to define the SOR method is:

$$x^{(k+1)} = (1-w)x^{(k)} + w\ GS\ ,$$

where GS is the $x^{(k+1)}$ vector that would be calculated by the Gauss-Seidel method for this iteration. When the A matrix (or Q matrix in the case of the quadratic variation system) is positive definite and consistently ordered ! (and it is for both the Laplacian and quadratic variation cases) then the optimal relaxation factor value can be computed by:

$$w = 2 / \{ 1+SQRT(p(J)^2)\} ,$$

where p(J) is the maximum eigenvalue of the iteration matrix for the Jacobi iteration method, $D^{-1}(L+U)$. Using this relaxation factor value, the SOR method should converge at least as fast as the Gauss-Seidel technique for that linear system. The GIPSY program RELAX4 is used to calculate this optimal relaxation value. The input file is the binary image that is described as the second input file to the INTERP program. The size of the problem for which the optimal relaxation can be computed with RELAX4 is restricted in that the entire Jacobi iteration matrix, $D^{-1}(L+U)$, must reside in memory at once since an EISPACK routine is used to compute the eigenvalues. The output from the RELAX4 command is a sequential file containing the optimal relaxation value and other text.

The optimal relaxation factor computed for the SOR method for the small test problem was 1.636408. The INTERP program with the R-flag set was used to perform the SOR iterations on the small problem with all other input conditions identical to the Gauss-Seidel test. The number of iterations for convergence was 24. The maximum change in a variable value was .993843, with an average change of .412097. The spectral radius of the SOR iteration matrix, $(D-wL)^{-1}\{(1-w)D + wU\}$, was .636408. So, the SOR iteration method showed a great improvement over the Gauss-Seidel for the small problem that was tested.

Unfortunately, the optimal relaxation factor could not be calculated using a reasonable amount of computer resources for the large test problem. Attempts to use the relaxation factor computed for the similar but smaller problem gave no advantage in performance over the Gauss-Seidel method for the large test problem. In fact, the relaxation factor values of 1.636408, 1.6, 1.7, and 1.8 were tried for the large problem and performance was sometimes worse than for the Gauss-Seidel method with no relaxation factor. It was concluded from these experiments that to gain the advantage that the SOR technique can give, the optimal relaxation factor must be computed for each specific problem to which it is to be applied.

One other iteration method was tested against the Gauss-Seidel and SOR methods. The Tridiagonal method as it will be called in this paper uses the following definitions:

$$A = N - P,$$

where A is as it was defined for the two other methods. The matrix N has the main diagonal, the super diagonal, and the sub-diagonal elements of A and zeros elsewhere. Both N and P are the same size as the A matrix. The P matrix contains the negative of all the non-tridiagonal elements of A and zeros in its super, main and sub-diagonal elements. Because of the nature of the Laplacian mask operation, as well as the masks for the quadratic variation, the N matrix is tridiagonal and symmetric. The tridiagonal iteration method takes advantage of this property of the N matrix and the fact that the LINPACK package has an efficient routine, DPTSL, for solving tridiagonal symmetric systems. The iteration formula for the tridiagonal method is

$$x^{(k+1)} = N^{-1}P\, x^{(k)} + N^{-1}b\ ,$$

where $x^{(n)}$ and b are defined the same way as for the Gauss-Seidel method. The GIPSY program which performs the tridiagonal method is TRIDAG. TRIDAG takes the same input files as described for the INTERP command and optionally computes the spectral radius of the iteration matrix for this method, $N^{-1}P$.

The tridiagonal method was tested using the same small and large test problems that were used to test the Gauss-Seidel and SOR methods. The results showed no advantage over the Gauss-Seidel technique and took more computer time to converge. The results for all tests described are shown in Tables 1 and 2.

TABLE 1

The iteration methods are compared for the small problem

METHOD	ITERATIONS	MAX DIFF	AVE DIFF	SP RADIUS
Gauss-Seidel	42	.995026	.653990	.9506328
SOR (w=1.636)	24	.993843	.412097	.6364080
Tridiagonal	43	.978874	.518923	.9529950

TABLE 2

The iteration methods are compared for the large problem

METHOD	ITERATIONS	MAX DIFF	AVE DIFF
Gauss-Seidel	26	.988121	.0882335
SOR (w=1.1)	30	.992550	.0570704
SOR (w=1.2)	33	.975332	.0382721
SOR (w=1.3)	33	.961666	.0301344
SOR (w=1.4)	31	.983982	.0264942
SOR (w=1.5)	29	.979816	.0224705
SOR (w=1.6)	27	.921153	.0180106
SOR (w=1.636)	26	.916611	.0173174
SOR (w=1.7)	25	.868000	.0267043
SOR (w=1.8)	29	.847610	.0575774
SOR (w=1.9)	49	.990700	.1368760
Tridiagonal	27	.959248	.077255

5.2.5 Another Method Tested

Besides the comparisons of the Gauss-Seidel, SOR, and Tridiagonal iteration methods described in the previous section, one other scheme was tested. The gradient projection method is described in [Grimson, 1981] where it is used to compute an interpolating surface that minimizes the quadratic variation. This section describes an implementation of this algorithm and compares it to the Gauss-Seidel iteration technique used to compute the quadratic variation interpolating surface.

The basic notion of the gradient projection algorithm is described in [Grimson, 1981] as (1) find an initial position on the objective function; (2) find the gradient at this position; (3) compute a direction vector by modifying the direction of the gradient by some rule; and (4) move to (compute) the lowest position on the objective surface along this direction and repeat again from step (2). Let S be defined as the current surface approximation with the given values at the ridge and valley pixels, and D be the direction vector image (or the negative of the gradient computed at each pixel in S with the value of D set to zero for the "known" pixels). The gradient projection algorithm to compute the interpolating surface through the ridge and valley pixels that minimizes the quadratic variation is

given in [Grimson, 1981] as: (Ø) start with an initial approximation surface, S; (1) compute the gradient of the objective function (minimizing the quadratic variation over the surface computed) by applying the masks shown a previous section to the pixels in the current S image. The direction vector D is computed as the negative of the results of convolving the appropriate mask with each pixel in S and setting the value of D to zero at the ridge and valley pixels; (2) compute ALPHA, a scalor, which specifies the amount to move along the direction vector D; (3) refine the surface approximation, S, by incrementing each pixel of the current S with the scaled direction vector (ALPHA*D) so that S_new(I,J):=S_current(I,J)+ALPHA*D(I,J), where (I,J) is the row and column location of each pixel; (4) return to step (1) and continue iterating until the magnitude of all components of D are smaller than some scalor value epsilon (the stopping criterion).

These steps of the gradient projection algorithm are implemented by a two-pass method to compute the quadratic variation interpolating surface. That is, each iteration that is done requires that each pixel in S be read and processed twice. In the first pass for each iteration, the appropriate mask is applied to each pixel and the D value for each pixel is computed. The ridge and valley pixels do

not have the mask applied since the D value is alwas going to be set to zero. In this first pass the value of the scalor ALPHA is also computed. The second pass is necessary for each iteration because the value ALPHA is used to compute the new S pixel values and the entire first pass must be completed before the value of ALPHA is known. In the second pass, the new value of S is computed for each pixel using the assignment formula:

$$S_{new(I,J)} := S_{current(I,J)} + ALPHA*D(I,J),$$

where (I,J) is the row and column location of each pixel. The stopping criteria for this implementation of the algorithm is similar to those described for the INTERP command which implements the Gauss-Seidel method. That is, iterations are done until either a given number of iterations have been performed, or until the maximum change in all S pixel values between two iterations is less than some given epsilon value. These criteria are used so that the Gauss-Seidel and the gradient projection iterative methods can be easily compared.

The Gauss-Seidel iteration method and the gradient projection method were compared by computing the quadratic variation surface for the large problem (4591 variables) used in previous tests. The stopping criterion used for both methods was the same. The maximum difference for all

pixels of S was specified to be less than 1.5. The Gauss-Seidel method achieved this criterion after 32 iterations. The CPU time required for this computation was 122.6 seconds. The gradient projection method described above went 39 iterations before meeting the stopping criterion and took 306.8 seconds. The difference in CPU time for these two methods computing the same problem is due to the two passes required for each iteration with the gradient projection method as opposed to only one pass per iteration for the Gauss-Seidel method. The two passes required more input and output operations. Another measure is the number of arithmetic operations required to compute a new value for a pixel using the two methods. Counting the operations required to compute the value for a pixel not located along the edge of the S image shows that the Gauss-Seidel method requires 14 additions, 13 multiplications, and 2 decisions. The gradient projection method requires 65 additions, 41 multiplications, and 8 decisions to compute the value of a pixel for each iteration. Because it took fewer iterations to meet the stopping criterion and because each iteration requires less CPU time due to the fewer number of operations, the Gauss-Seidel iteration method is judged better to compute the quadratic variation interpolation surface for the elevation model in this thesis.

5.3 INITIAL EXPERIMENTS

Well over 300 illumination model images have been produced as new features were added to the model, different computational techniques were tried, and various parameter values were tested. This section describes the sequence of steps used to produce one illumination model image and to explain the parameter values used. Briefly these steps are: pre-processing the LANDSAT data, creating the ridge and valley map, calculating the elevation model image, and building the illumination model image using shadowing, shading and backlighting effects. The final sub-section shows how the model of the study area is compared with the LANDSAT image of that area.

5.3.1 Preprocessing the LANDSAT Data

The first set of operations performed on the LANDSAT data was to read it off of magnetic tape, put it into the GIPSY SIF format, and extract the part of the large image that contains the study area of interest. The GIPSY commands BINSIF, DSPLY, and SBIMG were used to do this. The next operation was to remove the striping using the GIPSY DSTRPE command.

The programs which create the illumination model assume that each pixel represents a square area. Since the pixels in

Figure 22:
LANDSAT image of the study area after de-striping.

the LANDSAT image are not square, the image was re-sampled using the WARP command. The 80 pixel x 80 pixel study area became a 113 pixel x 80 pixel area, with each pixel representing a 56m x 56m square area instead of the 79m x 56m rectangular LANDSAT pixels. The new file will be refered as 'B73A.WRP'.

Figure 23:
LANDSAT image of the study area after re-sampling.

5.3.2 Ridge and Valley Image

After preprocessng the input data, the ridges and valleys were located by Shyuan Wang using the segmentation scheme described earlier in this chapter. The image resulting from his work must be modified to be used as input to the program that computes the elevation model image. The output from the segmentation process, a two-band file with valley pixels marked in band one and ridge pixels marked in band two, is used to produce two SIF files, both the same size as the 113 pixel x 80 pixel study area image. File one has the value 0 at valley pixels, 100 at ridge pixels, and the value 50 at all other pixels. The second file is a binary file with a value of 1 at all ridge or valley pixels and a 0 value at all other pixels. These two files are created using the GIPSY command MKCHK and BOOL and will be refered to as '050100.SIF' and 'RDGVAL.BIN', respectively.

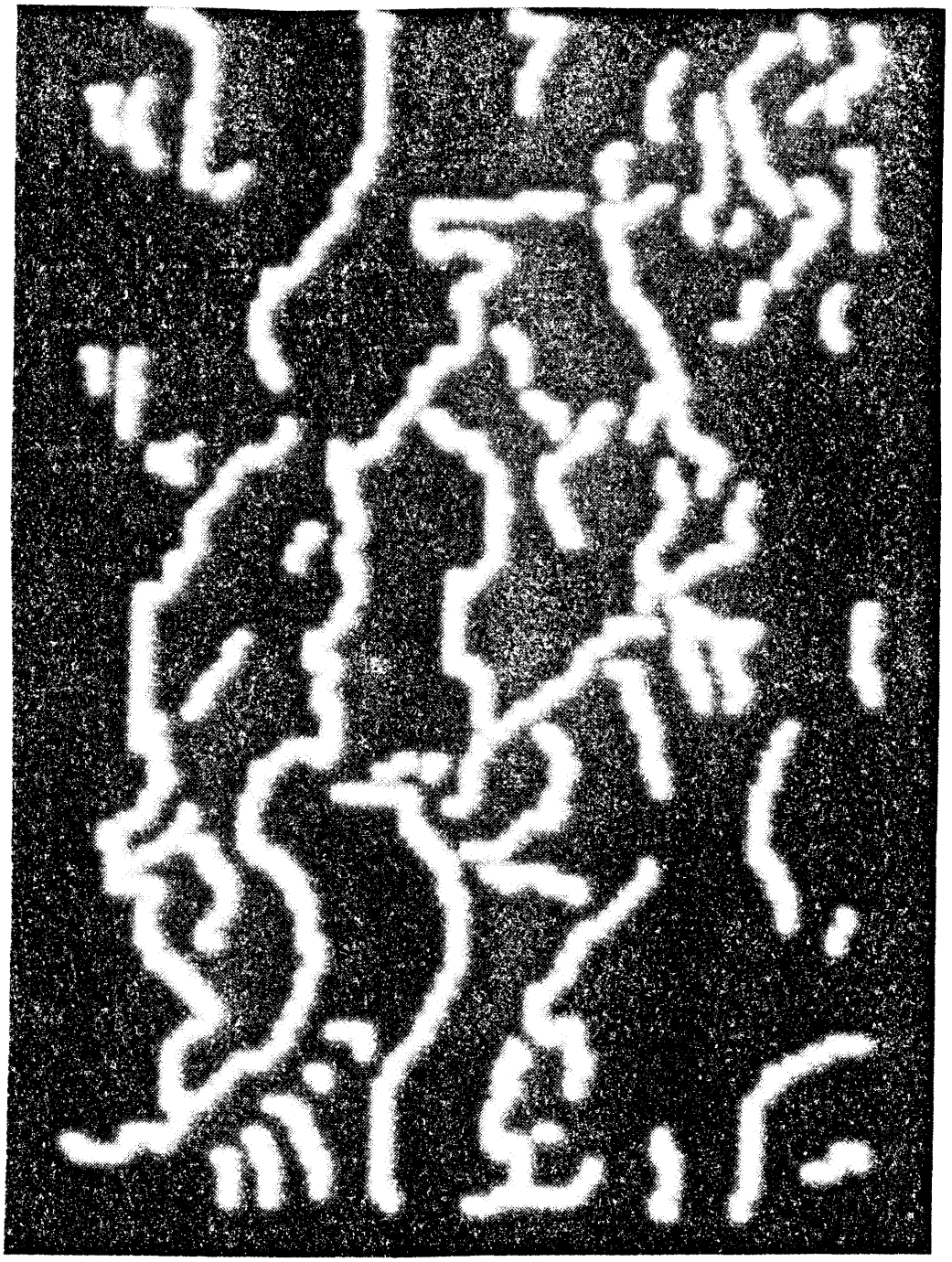


Figure 24:
Binary image showing valleys in black.

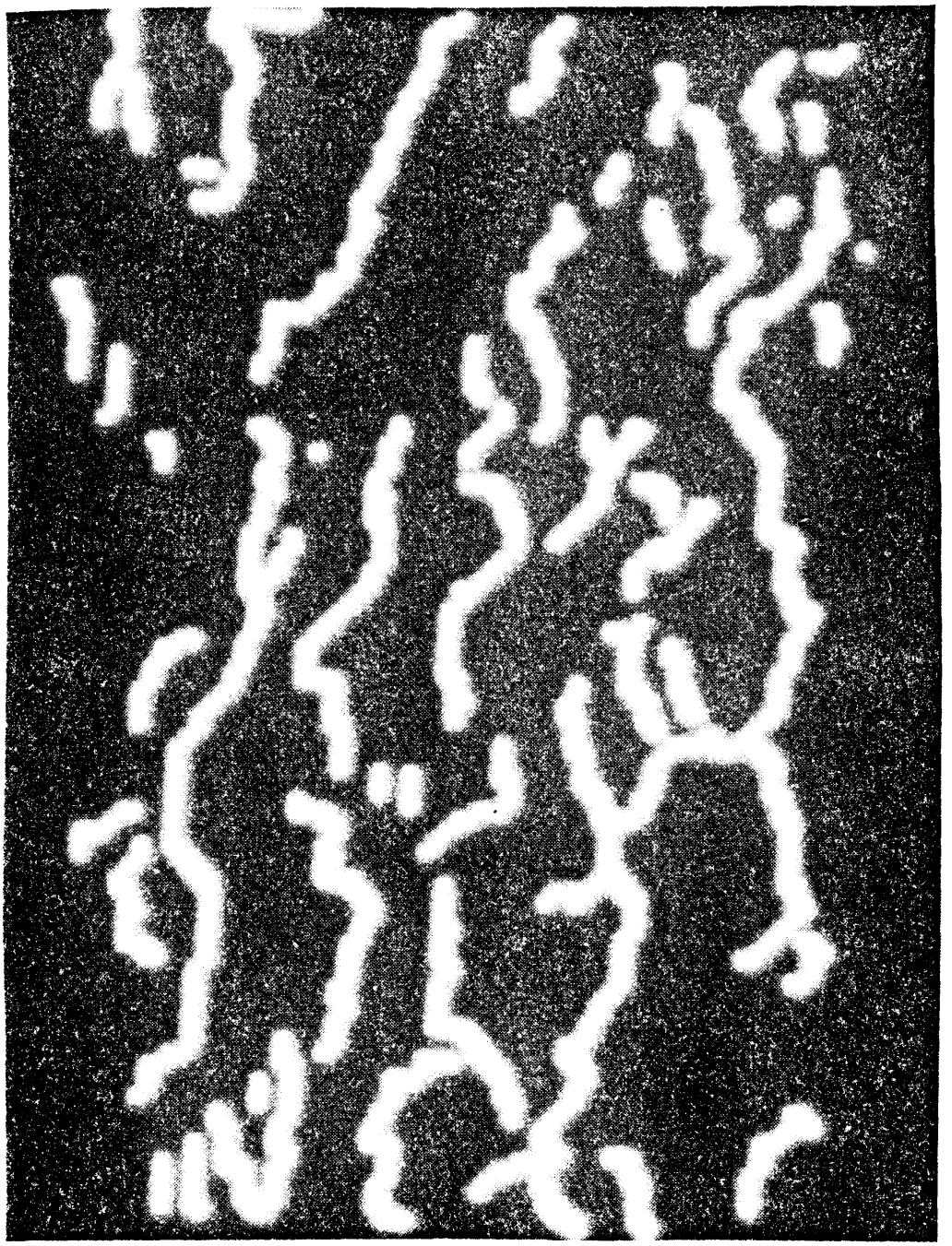


Figure 25:
Binary image showing ridges in black.

5.3.3 Elevation Model Image

Because the resulting image seems to more correctly represent the surface of our study area when visually compared to a topographic map, the elevation model created using the quadratic variation interpolation was chosen over that created using digital Laplacian techniques. The GIPSY program QVINTP combines Gauss-Seidel iteration and the mask operators which define the quadratic variation surface. The two input files used are '050100.SIF' and 'RDGVAL.BIN', respectively. The output file is named 'ELEVQV.OUT' and the B-flag is set to indicate that some pixels have "known" values and are not to be changed. The stopping criteria are specified by the number of iterations to perform and the maximum difference for any variable pixel between two iterations. The iterations stop when either (1) the specified number of iterations has been performed, or (2) no pixel changed between two iterations by more than the maximum difference value specified, which ever is satisfied first. The stopping criteria values used for this particular problem are 100 iterations and a maximum difference of 1.5. The QVINTP program stopped after 34 iterations, with a maximum difference of 1.479253. The average difference for all variable pixels between the last two iteration was .198347. The resulting image is stored in

SIF format in 'ELEVQV.OUT', the output file. Before using the 'ELEVQV.OUT' as an input file into the illumination model programs, the ARITHM command is used to add a constant 28 to all of the relative elevation values to make the minimum value Ø. The resultant image file 'ELEVQV.SIF', is displayed with bright values representing the higher elevation values and dark values representing the lower elevations in Figure 26.

Figure 26:
Digital elevation model image with dark representing low values and bright representing high values.

5.3.4 Illumination Model Image

The relative elevation model image, 'ELEVQV.SIF', is the input image file to the SHADOW command. The output file, 'QVSUN1.BIN', is a binary output file that has pixels that receive direct illumination marked with a '1' and pixels in shadow marked with a 'Ø'. The image orientation angle parameter for this LANDSAT image of the study area is 12 degrees. The azimuth of the sun at the time the LANDSAT image was created was 119 degrees. The ground distance multiplier for this image is an approximation using the known pixel size (56m x 56m) of the geometrically corrected image and the estimate that the average relief (distance from valley floors to ridge tops) is about 6ØØ feet. This estimate was made using measurements from the topographic maps of this area. To compute the factor to be multiplied to distances in the row and column direction to be in the same units as the elevation values in 'ELEVQV.SIF', it is assumed that the average relief value of the study area surface (6ØØ feet) is represented by 1ØØ units in the elevation model image (valleys here relative elevation value of Ø and ridges have relative elevation value of 1ØØ). Therefore, the number of these same units across a pixel (56 meters) is:

56 meters/6feet = 2204.72 inches/72 inches = 30.621118

The elevation of the sun above the horizon when the LANDSAT image was created was 45 degrees. The output image from this use of the SHADOW command is shown in Figure 27 with bright areas representing pixels that receive direct illumination and dark areas representing pixels in shadow with the sun at the 119 degree azimuth and 45 degree elevation.

Before using the SHADE command to approximate the intensities of reflected light that will be observed due to the orientation of each lit pixel with respect to the illumination source, the command CNSFCT is used to compute the 3-D orientation of each pixel. The central neighborhood slope facet model treats each pixel as if it were a plane oriented in row x column x greytone (elevation) space. A parameter fitting of the form

greytone value = ALPHA * Row + BETA * Column + G + error

is done within a rectangular window (specified by the user) around each pixel. Thus, ALPHA is the slope in the row direction of the pixel plane and BETA is the slope in the column direction. G is the independent term and error is the fitting error. The input file used to compute the orientation of each pixel in the Row x Column x Height (elevation) space is 'ELEVQV.SIF', the elevation model. The multiband SIF output file from CNSFCT is 'ELEVQV.SFT' and

Figure 27:
Binary image showing the shadowed pixels according to the illumination model in black for the study area.

contains the ALPHA value of each pixel in band 2, the BETA value of each pixel in band 3, and the scaling factors that may have been used when storing ALPHA and BETA as integer values in the descriptor records. A window size of 3 x 3 was used in the CNSFCT command.

Given the output from the SHADOW command, 'QVSUN1.BIN', as the first input file and 'ELEVQV.SFT' as the second input file, the SHADE command can be run to produce the output file 'QVSHD1.SIF'. The corrected azimuth of the sun parameter is 107 degrees (119 degrees - 12 degree image orientation), and the elevation angle of the sun is 45 degrees. The ratio of diffuse radiation to direct illumination is .14. This value is used because of the estimate of diffuse radiance made by Temps and Coulson [Temps and Coulson, 1976]. The intensity of the source parameter of 500 was chosen arbitrarily. The interesting properties of the illumination model will not be the intensity value at a particular pixel, but the relationship among the pixel locations and their values. The ground distance multiplier is 30.621118 as for the SHADOW command. A greytone representation of the intensities from the output file 'QVSHD1.SIF' is shown in Figure 28 for the sun at the 119 degree azimuth and 45 degree elevation with no backlighting.

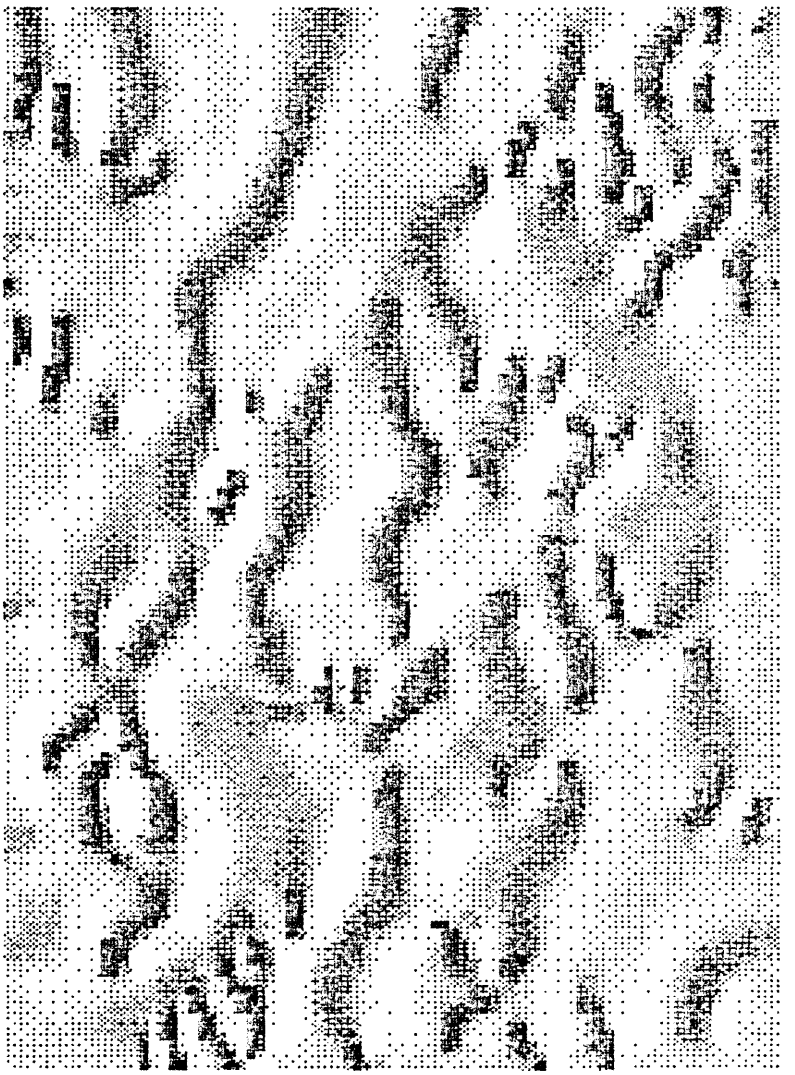


Figure 28:
The image 'QVSHD1.SIF', showing the light intensities from the primary source (119 degree azimuth and 45 degree elevation) including a constant diffuse radiance.

As mentioned in Chapter IV, backlighting is implemented by two applications of the SHADOW and SHADE commands. The SHADOW command has input file 'ELEVQV.SIF' (the same as the previous use of SHADOW) and the output file is 'QVSUN2.BIN'. The parameters for the application of the SHADOW command are the same as those for the previous use of SHADOW, except that the sun azimuth is 299 degrees (119 degrees + 180 degrees) for this case. Likewise, the second use of the SHADE command is very similar to the first. The input files are 'QVSUN2.BIN' and 'ELEVQV.SIF' and the output file is 'QVSHD2.SIF'. The corrected sun azimuth is 187 degrees (119 degrees - 12 degrees). The intensity value of the source is 100. This value was chosen after many experiments showed it yielded the backlighting effects that compared most favorably to those in the LANDSAT image. No other parameter values are different from the previous use of SHADE.

The results of the two uses of the SHADE command, 'QVSHD1.SIF' and 'QVSHD2.SIF', are added together (pixel by pixel) by the ARITHM command as described in Chapter IV. The resultant image, the illumination model image, is called 'QVILLUM1.SIF'.

```
SHADOW ELEVQV.SIF > QVSUN1.BIN
    Enter orientation of top of image :  12.
    Enter azimuth of sun              : 119.
    Enter ground distance multiplier  :  30.621118
    Enter elevation angle of sun      : 45.


CNSFCT ELEVQV.SIF > ELEVQV.SFT
    Enter window size for
     row by column          : 3 3
    Enter number of bits    :  16


SHADE QVSUN1.BIN, ELEVQV.SFT > QVSHD1.SIF1
    Enter corrected sun azimuth      :  107.
    Enter sun elevation angle        :  45.
    Enter diffusion ratio            :  .14
    Enter sun intensity              :  500
    Enter ground distance multiplier :  30.621118


SHADOW ELEVQV.SIF > QVSUN2.BIN
    Enter orientation of top of image :  12.
    Enter azimuth of sun              : 199
    Enter ground distance multiplier  :  30.621118
    Enter elevation angle of sun      : 45.


SHADE QVSUN2.BIN, ELEVQV.SFT > QVSHD2.SIF
    Enter corrected sun azimuth      :  187
    Enter sun elevation angle        :  45.
    Enter diffusion ratio            :  .14
    Enter sun intensity              :  100
    Enter ground distance multiplier :  30.621118


ARITHM QVSHD1.SIF, QVSHD2.SIF > QVILLUM1.SIF (A)
    Decrease number of bits ?   :  N
```

Figure 29:
The sequence of commands to generate the illumination model.

5.3.5 Comparing the Model and LANDSAT Images

As it was noted in the previous sub-section, the illumination model image is a qualitative result. Since the pixel values in the LANDSAT image and the illumination model image are not in the same range, comparison can be difficult. To compare the images more easily, the REGRES and PQUANT commands are used. REGRES with 'QVILLUM1.SIF' as the first input file and 'B73A.WRP' as the second input file and the M-flag set, produces a sequential output file, called 'QV1WRP.REG' here. For each pixel value in the first input file (the domain), the pixels in the first file with that value are located and the mean value of the corresponding (with same row and column coordinates) pixels in file two is computed. The output file contains two columns of numbers. The left column has values from the first input file, the right column has the mean value computed from the second file.

The PQUANT command with the R-flag set, takes two input files. The first file is the same as the first file for the REGRES command. The second input file is the output file from the REGRES command. The output file from PQUANT is the first input file with each of the values from the left column of the second input file replaced by the corresponding value in the right column. Thus, this REGRES

and PQUANT combination does a mapping of the greytone values from the illumination model image, 'QVILLUM1.SIF', into the greytone value range of the LANDSAT image. The output file from PQUANT will be called 'QVILLUM1.REG'.

After mapping the greytone values of the illumination model into the same range as the greytone values of the LANDSAT image, comparison of the two images by direct methods is possible. The ARITHM command is used to subtract the values of each pixel of 'QUILLUM1.REG', the illumination model image, from the corresponding pixel values of 'B73A.WRP', the LANDSAT image. The absolute value of the output file, 'WRPILL1.DIF', is shown in Figure 32.

```
REGRES QVILLUM1.SIF, B73A.WRP > QVWRP1.REG (M)

PQUANT QVILLUM1.SIF,QVWRP1.REG > QVILLUM1.REG (R)
```

Figure 30:
Command sequence for the regression method.

Figure 31:
The illumination model image after regression.

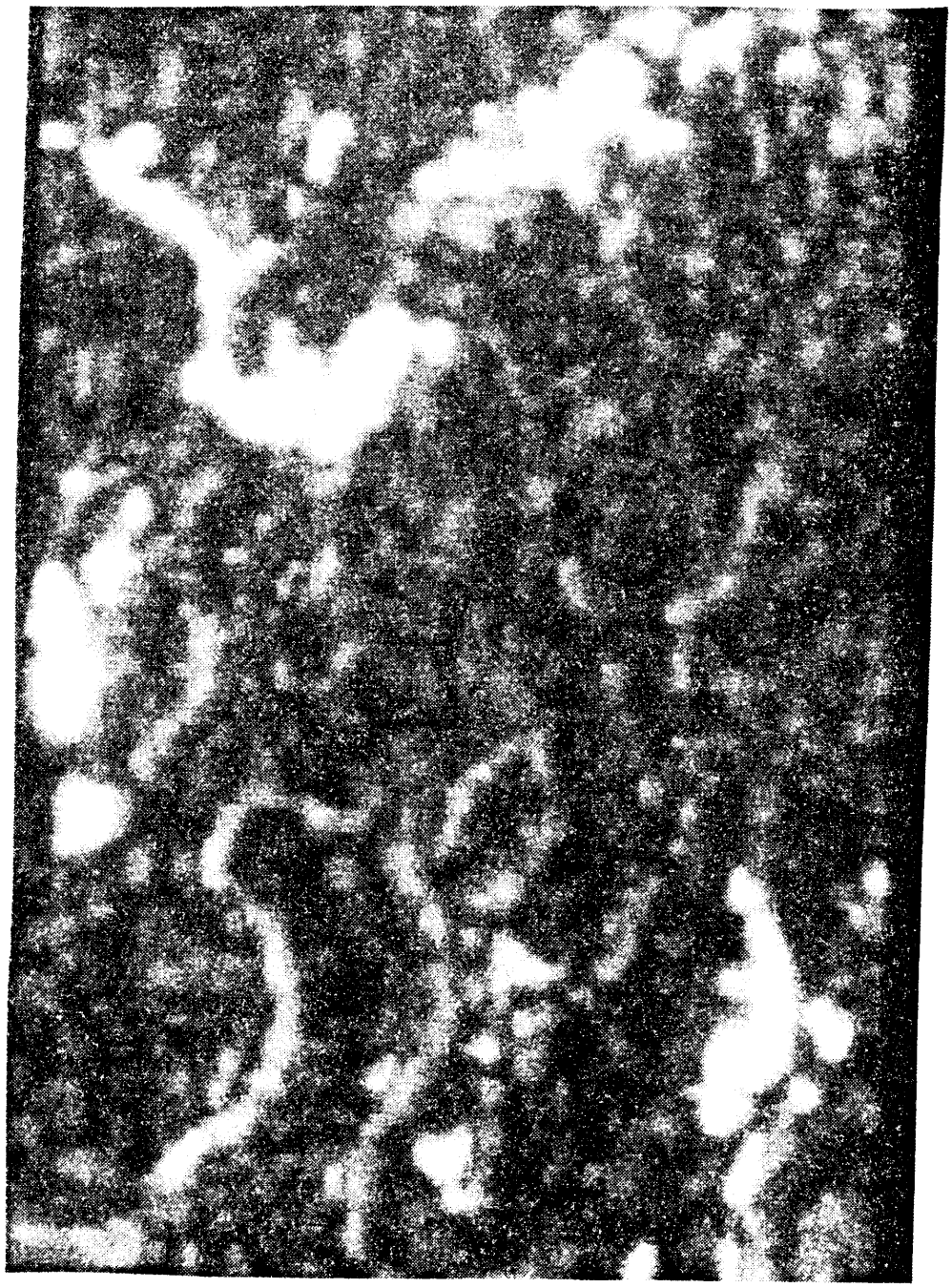


Figure 32:
The absolute value of the difference between the LANDSAT image and QVILLUM1.REG, the regressed illumination image.

Chapter VI

MORE EXPERIMENTS

To be able to understand the differences between the illumination model image and the LANDSAT image, some additional experiments were performed. First, a histogram (plot of frequency versus greytone value) of the difference image 'WRPILL1.DIF', was done using the MHIST command. The negative values represent pixels where the illumination model image values were higher than the corresponding LANDSAT image pixel values. The positive values represent pixels where the illumination values were lower. Of course, the zero values are pixels where the values are the same in both images. Obviously, the pixels with the highest absolute values (both the large negative and large positive) are of most concern. A threshold value of 1Ø is chosen since the histogram seems to indicate that the high positive error values are separated from the average error values there. A second threshold value of -8 is chosen similarly for the high negative error values. Most of the pixels in 'WPRILL1.DIF' with values above the first threshold correspond to very bright areas of the LANDSAT image. Similarly, most of the pixels with values below the second threshold correspond to the very dark areas of the LANDSAT

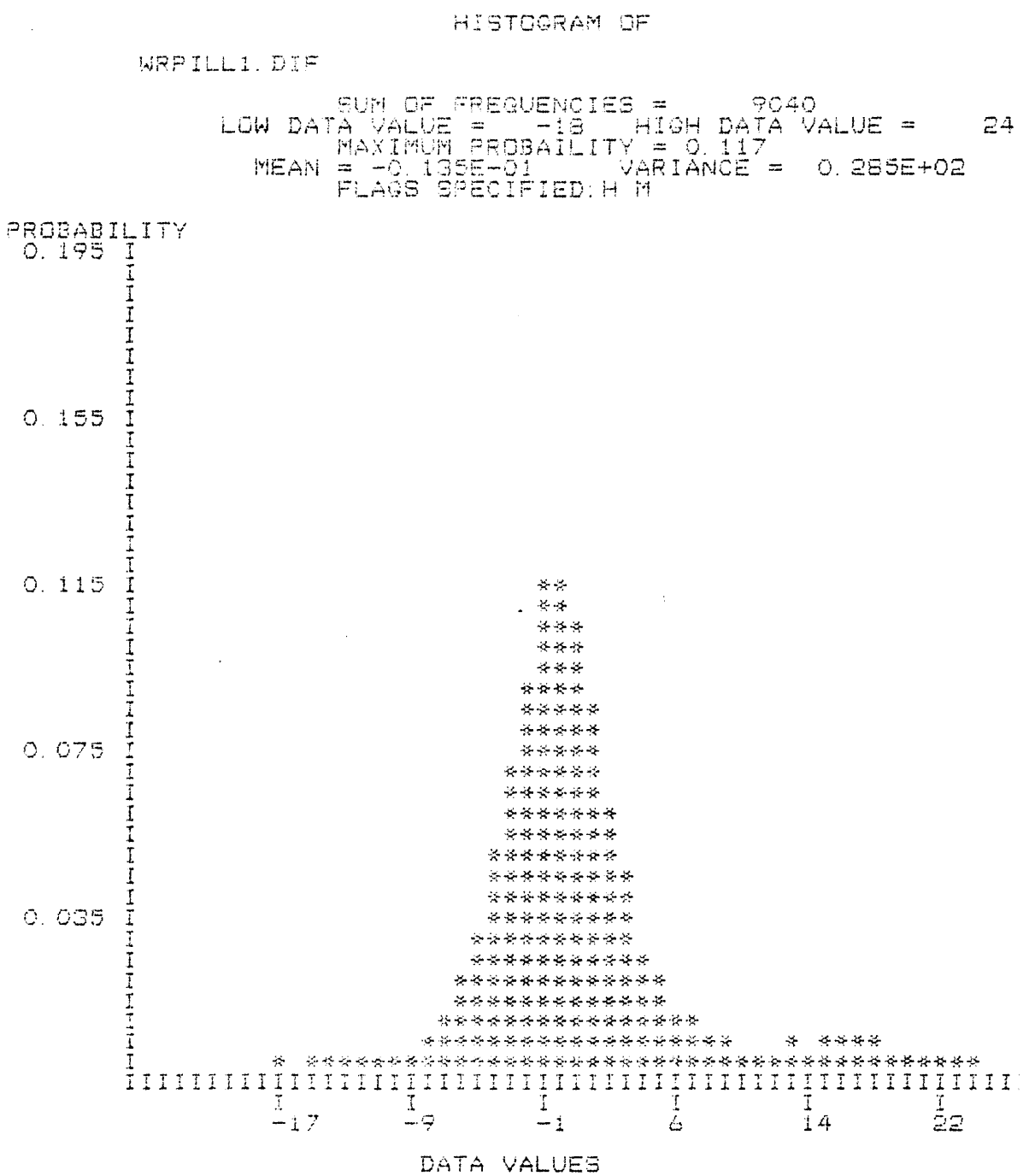


Figure 33:
Histogram of the difference image, 'WRPILL1.DIF'.

image. Since these pixels have values well above and below the mean greytone value of the LANDSAT image, it may be hypothesized that the brightness and darkness of these pixels, that was incorrectly modeled by the illumination model, is due to reflectance properties of those pixels.

The illumination model image, 'QVILLUM1.SIF', was created assuming all pixels have the same reflectance values. If the large positive and large negative difference values in 'WRPILL1.DIF' are due mostly to exceptional reflective properties, then including the pixels from the very bright and very dark areas of the LANDSAT image in the regression process could have also caused some of the "not-so-high" differences shown near the middle of the histogram of 'WRPILL1.DIF'.

To test this possibility, of thresholding operation was done on the LANDSAT image 'B73A.WRP' to extract the pixels from very bright and very dark areas. Two binary images, one showing very bright pixels in black, the other showing the very dark pixels in black are shown in Figures 37 and 38.

The REGRES and PQUANT command sequence is run again, however this time the REGRES command has the O-flag set to indicate that some pixels from the second input file are not to be included in the regression. The output file from this two

Figure 34:
Very bright pixels from 'B73A.WRP', the LANDSAT image.

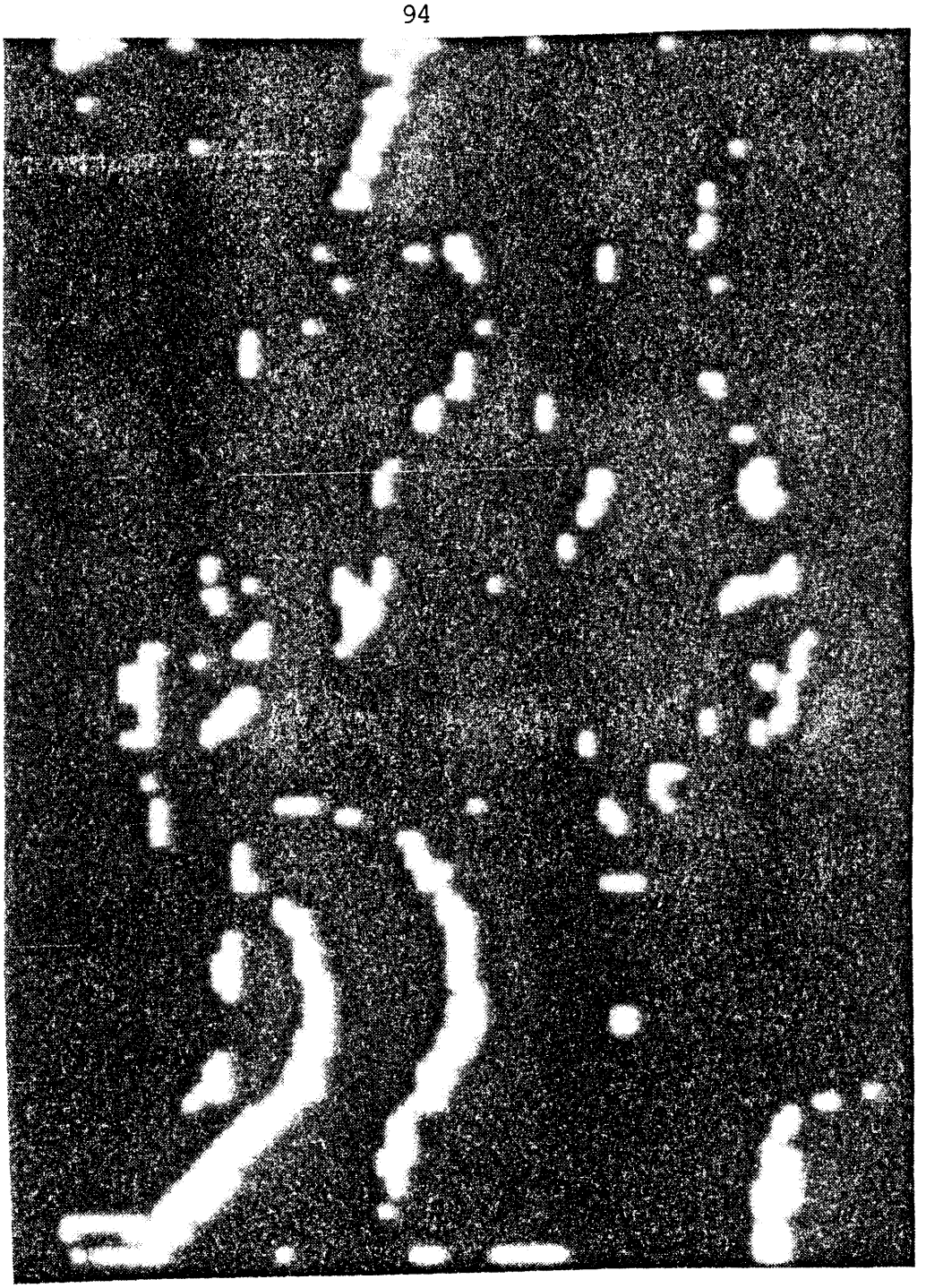


Figure 35:
Very dark pixels from 'B73A.WRP', the LANDSAT image.

command procedure is 'WRPILL1.DF2'. Histograms that include only pixels in the middle error range (between -8 and 10) from both 'WRPILL1.DIF' (difference image between LANDSAT data and the illumination with the first regression scheme) and 'WRPILL1.DF2' (difference image between LANDSAT data and the illumination model using the regression that excluded pixels from very bright and very dark areas in the LANDSAT image) are shown in Figures 37 and 38.

Examining the two histograms shows that the average error for the pixels in the middle error range was -.0335 for the regression scheme that excluded the very bright and very dark pixels, compared with -.855 for the earlier case.

This experiment shows that most of the difference between the illumination model and the LANDSAT data being modeled occurs at these very bright and very dark areas on the LANDSAT scene. Additionally, it shows that in order to reduce the differences over the whole image, these "special" areas must be accounted for in the illumination model.

To try to improve the performance of the illumination model in these "problem areas", a reflectance map is built to give high reflectivity values to pixels corresponding to large positive differences in 'WRPILL1.DIF', and low reflectivity values for pixels corresponding to large negative differences. The values chosen are 100 for high

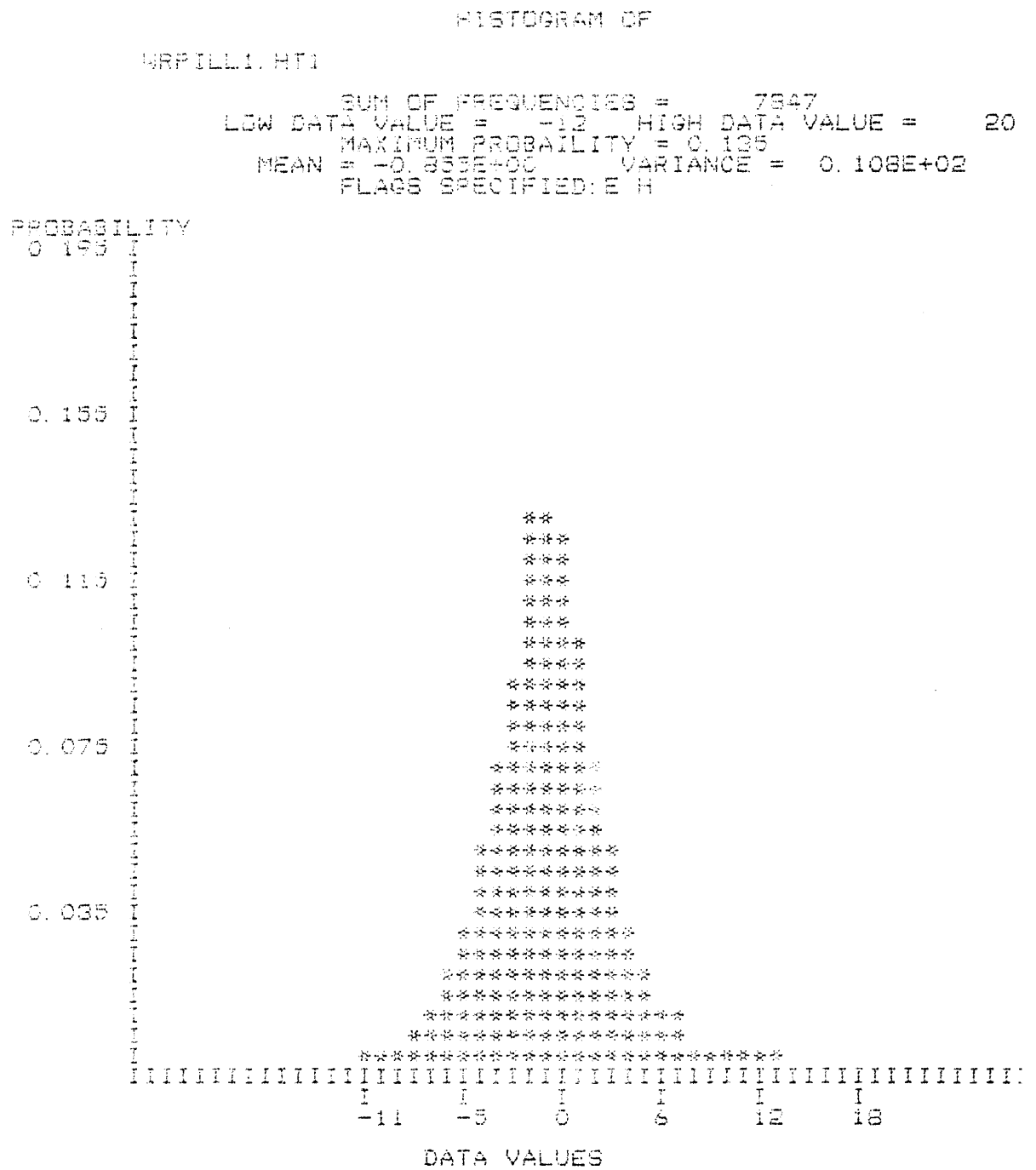


Figure 36:
Histogram of the pixels in the middle range (-8 to 10) of the first difference image, 'WRPILL1.DIF'.

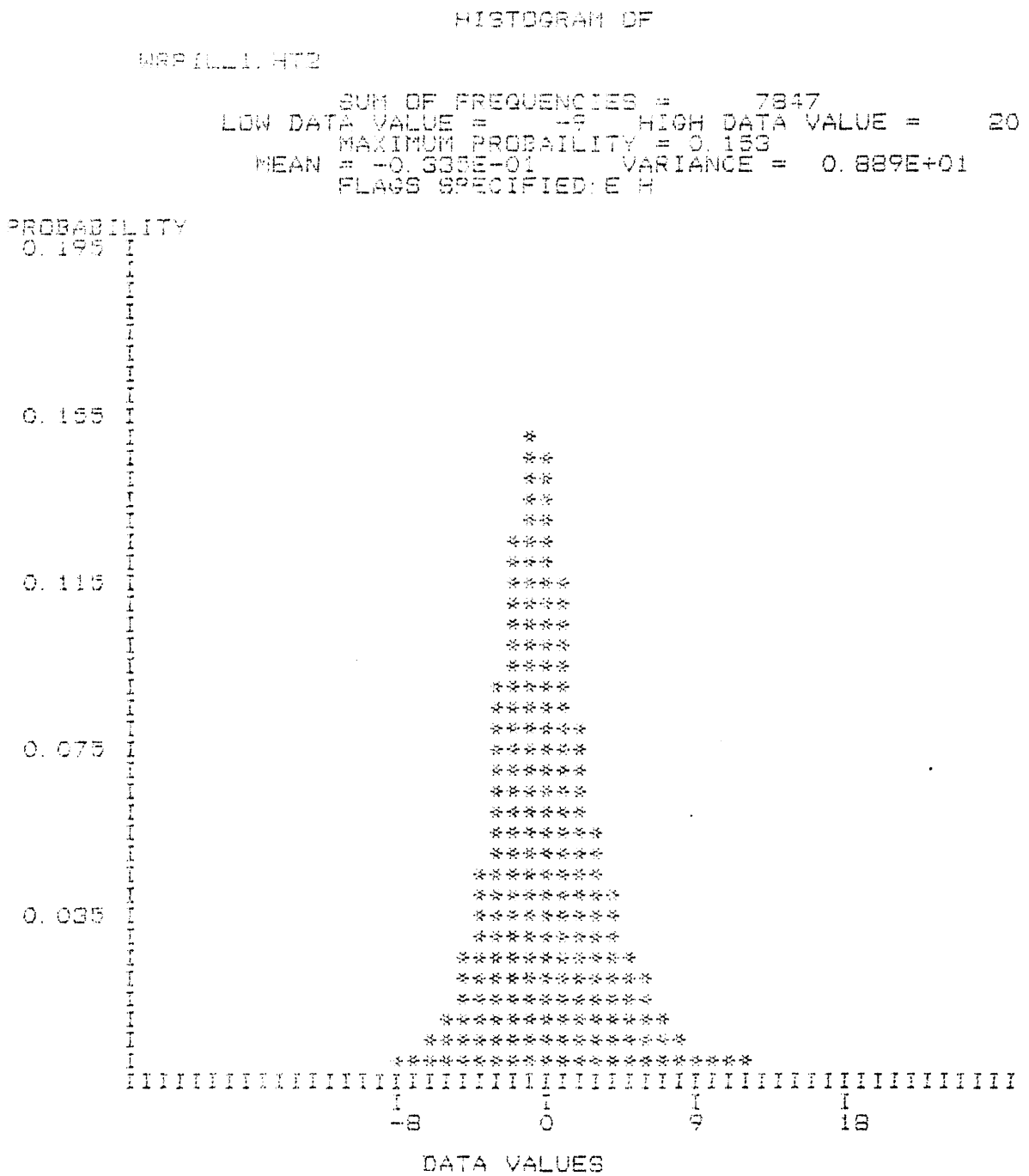


Figure 37:
Histogram of the difference image created using the new regression scheme, 'WRPILL1.DF2'. The same pixels shown in the previous figure are shown here.

reflectivity, 60 for low reflectivity and 80 for all other pixels. SHADE is used again for the sun at both azimuths (119 degrees and 189 degrees) with the same input files and parameters as before, however, the reflectance map 'REFLCT1.RFT' is included and the R-flag is set. The output files for the two applications of SHADE are called ' 'QVSHD12.SIF' and 'QVSHD22.SIF'. The result after using ARITHM to sum these two files is 'QVILLUM2.SIF', the new illumination model image. The REGRES and PQUANT commands are run to map the greytone values of this illumination model image into the range of the greytones of LANDSAT image for comparison as for the first illumination model. The difference between the model and the LANDSAT image is computed using ARITHM and the result is called 'WRPILL2.SIF'. The absolute value of the difference image and the histogram of 'WRPILL2.DIF' (difference between the LANDSAT data and the new illumination model image) are shown in Figures 39 and 40, respectively. The root mean square difference calculated over all pixels is 3.8874.

The average error for all pixels and the error in the "problem areas areas" is lower for this illumination model that includes the reflectivity information derived from the experiments described earlier.

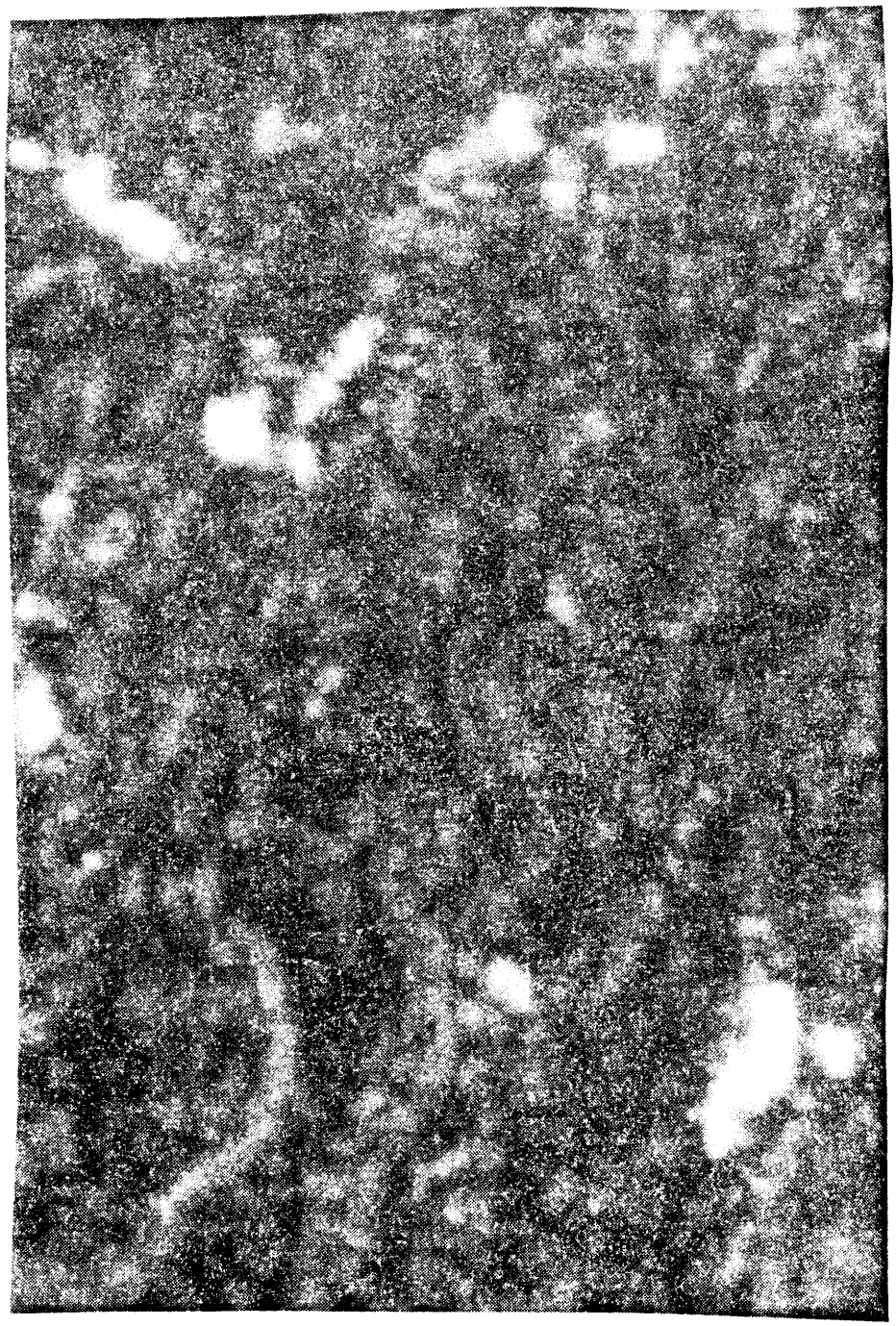


Figure 38:
The absolute value of the differences between the LANDSAT image and the new illumination model including the reflectivity map.

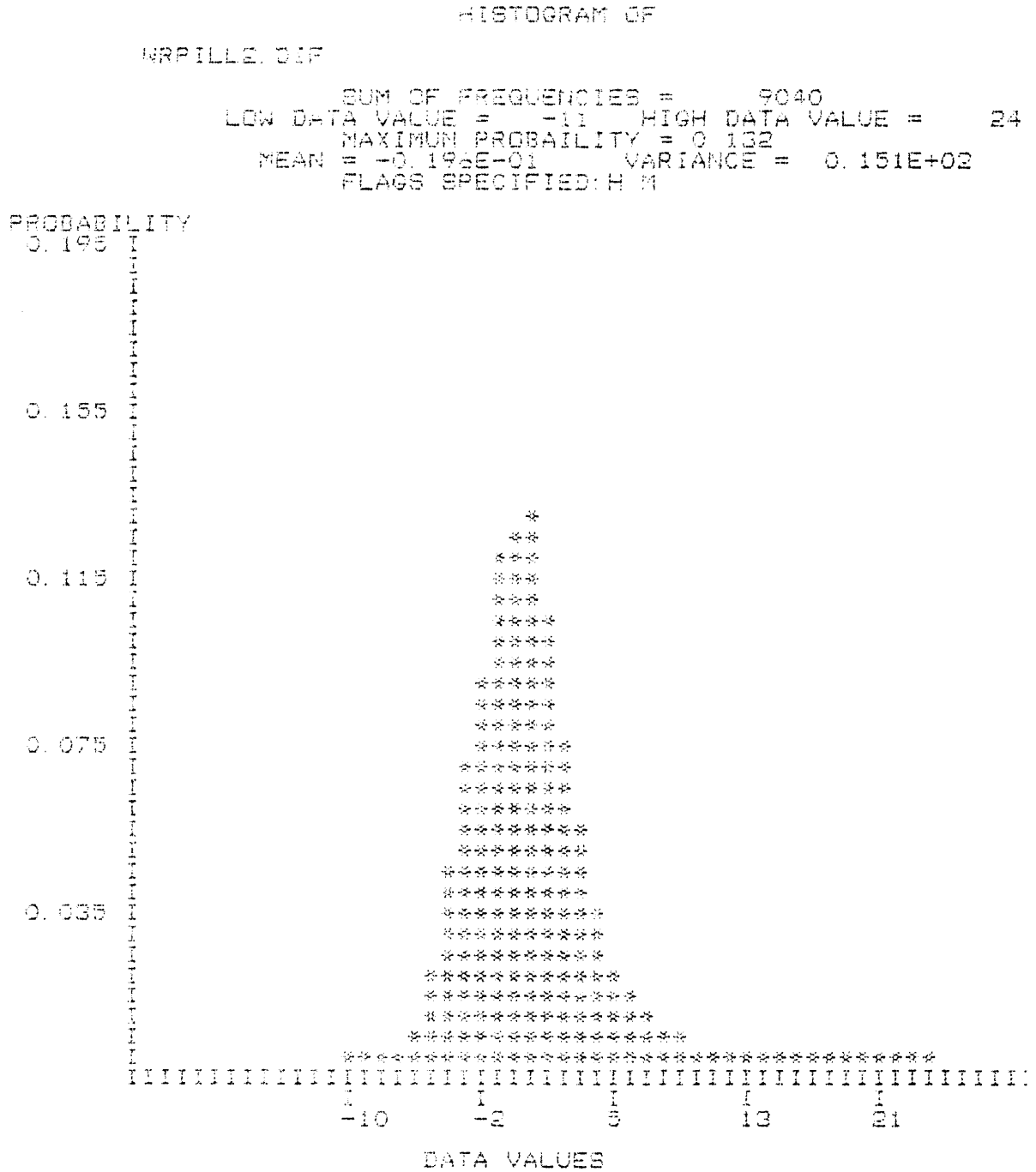


Figure 39:
The histogram including all pixels from the difference image 'WRPILL2.DIF'.

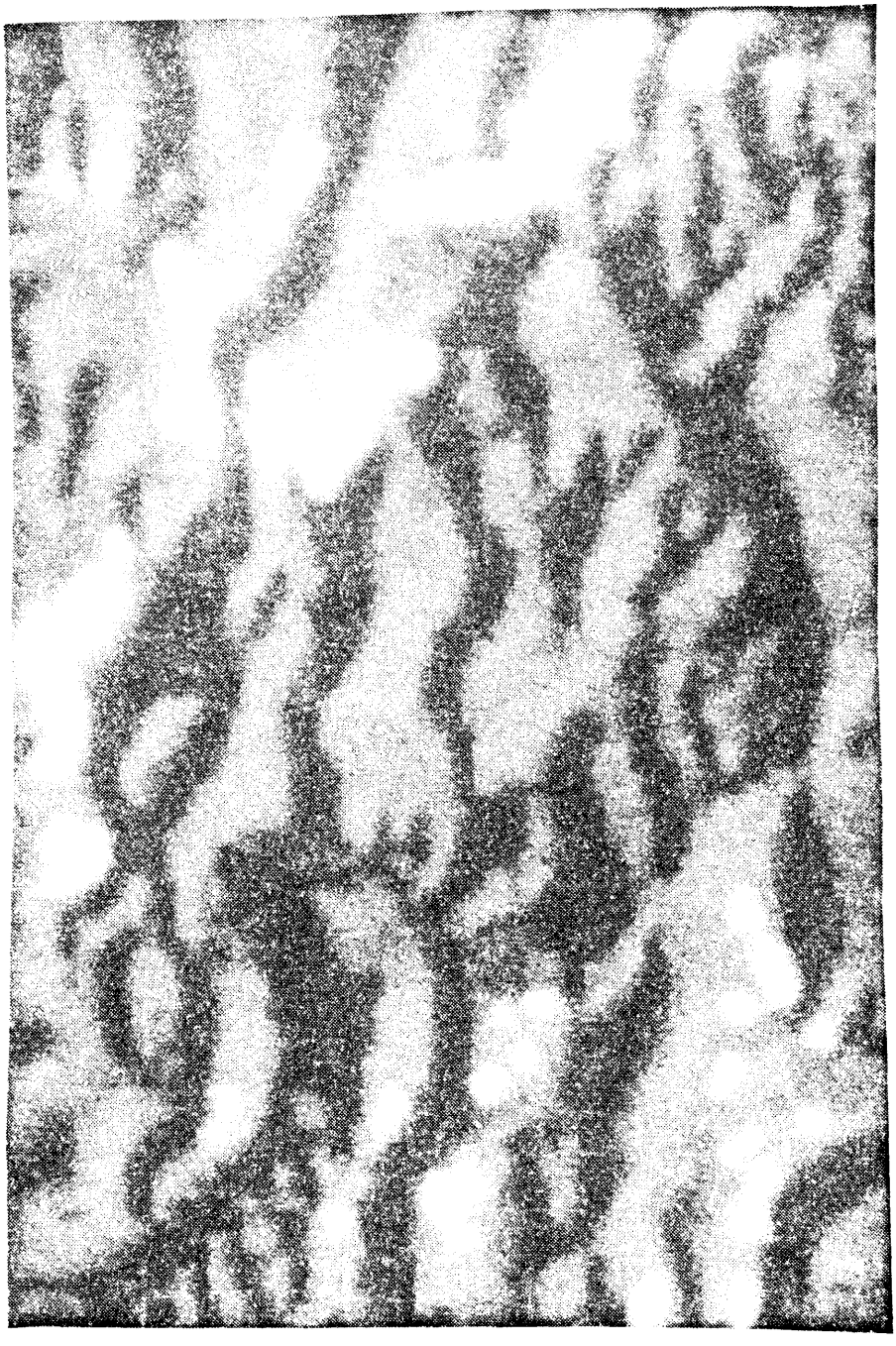


Figure 40:
The new illumination model image, including the reflectivity values.

Chapter VII

DISCUSSION AND CONCLUSIONS

The objective for the research described in this thesis was to develop an illumination model that can be used to help understand features observed in a LANDSAT scene of a mountainous area. The work that accomplished this objective included: creating a digital model to represent the topography of the area derived from the LANDSAT image itself; testing various implementations of this elevation model; considering the physics of illumination to develop a model that is relatively simple, but produces reasonable results; and, repeatedly comparing the illumination model results to the LANDSAT image to derive parameter values not known a priori that minimize differences between the model and the "real world" data. In this chapter, the accomplishments at each step will be discussed. Finally, suggestions for future work will be proposed.

The digital elevation (terrain) model used in this work was derived from the LANDSAT image, not digitized from a topographic map as is the most common technique recorded in the literature. The model image generated relies heavily on the ridge and valley pixels produced by the segmentation method implemented by S. Wang. The segmentation algorithm

would be expected to perform poorly for LANDSAT images created when the sun was near its zenith (high elevation angles), reducing the observed shadowing, or when the majority of the ridge lines (and associated valleys) were parallel to the solar beam. In the study area examined in this research, however, the major topographic features are almost normal to the illumination source and the elevation of that source does indeed result in sharp contrast between shadowed and directly illuminated areas. The ridges and valleys located seem to agree quite well with the topographic maps for this study area. A more detailed comparison with ground truth is presented in [Wang, et al., 1982].

Since the elevation values assigned to all ridge pixels are the same, and a single elevation value is assigned to all valley pixels, the model that results from assigning a value to each pixel determined by an interpolating surface that has the given, constant values at the ridge and valley pixels is a relative elevation model. That is, the pixel values in the elevation model image will not exactly represent the true elevation values that could be measured on the ground, since the values 0 and 100 (chosen as the "known" values of the valley and ridge pixels) were chosen arbitrarily. An advantage of the technique used here to

create the elevation model is that the model is precisely registered to the LANDSAT scene it represents.

The interpolation surface chosen to represent the topography of the area being studied had the constraints that the values Ø and 1ØØ corresponding to the valley and ridge pixels be preserved for those pixels and that the surface minimizes the quadratic variation. The results for this simple surface seemed to agree quite well with the topographic map in a qualitative sense. In other words, the ridges seemed to be located correctly as did the valleys. The elevation model generated by this method had "peaks" with values greater than 1ØØ. These peaks seem to correspond to the areas with greatest elevation on the topographic map. To correctly assess the true accuracy of even the qualitative properties of this model (whether the relationship between each pair of pixel is the same for both the model and ground truth), however, a digital terrain model derived from a topographic map or some other ground truth source would have to be registered and compared with the model image.

The conclusions that were drawn about the different iteration methods considered for implementing the elevation model were based on the tests described in Chapter 5. Though the Successive Over-Relaxation (SOR) method took

fewer iterations than the Gauss-Seidel method on the smaller problem, a major drawback was the fact that the optimal relaxation factor has to be calculated for each new linear system to which it is applied. The expense (in computer time) that would be saved by using the SOR method on large problems would be offset by expense of calculating the needed relaxation factor which itself involves solving a linear system. Though the tridiagonal iteration method took about the same number of iterations as the Gauss-Seidel method, the computer time required to do those iterations was much greater. For systems in which the A or Q matrix is symmetric and positive definite, as is the case for the interpolating surfaces defined by the masks in this report, the Gauss-Seidel method will always converge.

The illumination model defined and used in this research considers the three sources of radiance mentioned earlier. Direct illumination and the shadowing and shading it causes has the most observable effect, both in the real world and in the model image constructed in this work. The very dark pixels observed in the LANDSAT scene were found to be due to shadowing, except for the pixels of very low reflectivity values, but dark pixels surrounding the true shadowed pixels were found to be due diferential shading due to their orientation with respect to the solar beam. Compare Figures

30, pixels in shadow, and 26, the LANDSAT image. The diffuse lighting component of the illumination model proved to be important because without it shadowed areas would appear completely dark, but the darkest pixels in the LANDSAT image seem to be due to low albedo rather than shadowing.

Backlighting was modeled as a secondary and less intense source from the opposite direction of the primary source (the sun). The effect of backlighting on the model image is observed on pixels on slopes in shadow. With sky light as the only form of diffuse lighting, these pixels would have a uniform greytone value since skylighting is modeled as a constant value. The shadowed slopes on the particular LANDSAT image here were not very wide, however, due to the narrow valleys with steep sides in this area. The backlighting effects did not prove to be substantial for this study area, but for areas with larger slopes, the effects would certainly be more noticeable.

The parameter values used in the illumination model were given for the particular LANDSAT scene (azimuth angle and elevation angle of the sun, orientation of the image, and pixel size), estimated from a priori knowledge about the study area (the ground distance multiplier), and arrived at through experimentation (sun intensity and reflectivity

values). About 300 illumination model images were generated with different combinations of intensities for the primary and secondary sources used for backlighting. The combination was considered best that yielded the lowest root mean square difference between pixels in the LANDSAT image and the illumination model with that parameter value combination. The values for source intensity of 500 for the source of 119 degree azimuth and 100 for the secondary source at 189 degrees were chosen. That is, backlighting was about one-fifth the intensity of the direct ilumination.

The reflectivity values used in the tests described in Chapter 6 were also experimentally chosen. That is, values were tried, the results analyzed, and other values were used based on that analysis. By chosing reflectance values for each pixel in the model using this iterative trial and error process, an illumination model that approximates a particular LANDSAT scene even more closely can be derived.

Some additional research can be proposed which would build on the work described in this thesis. First, more methods for creating elevation models could be explored, but even more important, the new and current elevation models should be compared to a digital elevation model derived from ground truth data and registered to the LANDSAT scene. This would indicate which methods of deriving the elevation model

from the LANDSAT data alone are most correct and might aid in creating even better elevation models.

Given a digital elevation model from ground truth information, the illumination model itself could be tested to determine even better the parameter values that are required to accurately approximate the LANDSAT image. This statement is made because some of the differences that are observed between the current illumination model and the LANDSAT scene may be due to differences between the elevation model and the real world.

Another experiment that might have value is to try applying this model to other study areas. In areas with little shadowing or in LANDSAT scenes with considerable atmospheric scattering, the results may be less encouraging than for the area chosen here. For areas with ridges and valleys oriented parallel to the solar beam, there would certainly be more error using the techniques reported here to create an elevation model. The illumination model has been used to approximate the shading of simple surfaces such as spheres and hyperbolic paraboloids illuminated from different directions quite successfully. The effects of complex terrain combined with such objects as clouds and other factors that make an image less ideal than the LANDSAT scene studied here might produce interesting results.

REFERENCES

Anderson, Arthur T. and Schubert, Jane, "ERTS-1 Data Applied to Strip-Mining", Photogrammetric Engineering and Remote Sensing, vol. 42, n. 2, February 1979, pp. 211-219.

Anderson, Raymond R., "Production of a Map of Land-Use in Iowa Through Manual Interpretation of LANDSAT Imagery", Proceedings of the International Symposium on Remote Sensing of the Environment, 11th, Vol. 2, University of Michigan, Ann Arbor, Michigan, April 25-29, 1977, pp. 827-836.

Anuta, Paul E., "Digital Registration of Topographic Data and Satellite MSS Data for Augmented Spectral Analysis", Annual Meeting Proceedings of the American Society of Photogrammetry, 42nd, Washington, DC, February 22-28, 1976, pp. 180-187.

Billingsley, F.C., Gillespie, A.R., and Goetz, A.F.H., "Computer Image Processing", Application of ERTS Images and Image Processing to Regional Geologic Problems and Geologic Mapping in Northern Arizona, JPL Technical Report 32-1597, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, May 15, 1975, pp. 13-22.

Billingsley, F.C. and Goetz, A.F.H., "Computer Techniques Used for Enhancements of ERTS Images", Application of ERTS Images and Image Processing to Regional Geologic Problems and Geologic Mapping in Northern Arizona, JPL Technical Report 32-1597, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, May 15, 1975, pp. 1219-1228.

Brown, Fred R. and Karn, Fred S., "Air Pollution from the Ohio River and Monogahela River Valleys", ERTS-1, A NEW WINDOW ON OUR PLANET, Geological Survey Professional Paper 929, Richard S. Williams, Jr. and William D. Carter, editors, United States Government Printing Office, Washington, DC, 1976, pp. 261-265.

Campbell, Russell H., "Active Faults in the Los Angeles-Ventura Area of Southern California", ERTS-1, A NEW WINDOW ON OUR PLANET, Geological Survey Professional Paper 929, Richard S. Williams, Jr. and William D. Carter, editors, United States Government Printing Office, Washington, DC, 1976, pp. 113-116.

Dave, J.V., "Contrast Attenuation Factors for Remote Sensing", IBM Journal of Research and Development, Vol. 23, n. 2, March 1979, pp. 214-224.

Grimson, W.E.L., "An Implementation of a Computational Theory of Visual Surface Interpolation", Artifical Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02319, 1981.

Haralick, Robert M., "Analysis of Row Crop Reflectance", Unpublished paper at Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, May 1980.

Haralick, R.M. and Shanmugam, K.S., "Combined Spectral and Spatial Processing of ERTS Images", Symposium on Significant Results Obtained from Earth Resources Technology Satellite - 1, NASA SP-327, Washington, DC, pp. 1219-1228.

Horn, Berthold, K.P., "Understanding Image Intensities", Artificial Intelligence, Vol. 8, 1977, pp. 201-231.

Horn, Berthold K.P. and Bachman, Brett L., "Using Synthetic Images to Register Real Images with Surface Models", Communications of the ACM, Vol. 21, 1978, pp. 914-924.

Kimes, Daniel S. and Kirchner, Julie A., "Modeling the Effects of Various Radiant Transfers in Mountainous Terrain on Sensor Response" IEEE Transactions On Geoscience and Remote Sensing, Vol. GE-19, No. 2, April 1981, pp. 100-108.

Kim, Soon T. and Smith, David W., "Elimination of Evnironmental Effects from LANDSAT Radiance Data", Annual Meeting Proceedings of the American Society of Photogrammetry Photogrammetry, 45th, Technical Papers, Washington, DC, March 18-24, 1979, pp. 694-699.

Kriebel, Karl Theodor, "On the Variability of the Reflected Radiation Field Due to Differing Distributions of the Irradiation", Remote Sensing of the Environment, Vol. 4, Number 4, 1976, pp. 257-264.

Kitcho, C.A., "Optimum LANDSAT Sun Angles for Extreme Constrasts of Terrain", International Symposium on Remote Sensing of the Environment, 13th, Vol. 2, University of Michigan, Ann Arbor, Michigan, April 23-27, 1979, pp. 1213-1215.

Krinsley, Daniel B., "Selection of a Road Alinement Through the Great Kavir in Iran", ERTS-1, A New Window on Our Planet, Geological Survey Professional Paper 929, Richard S. Williams, Jr. and William D. Carter, editors, United States Government Printing Office, Washington, DC, 1976, pp. 296-299.

Krusemark, Scott, "A Transportable Image Processing System Called GIPSY", Masters Thesis, Virginia Polytechnic Instutite and State University, 1981.

Lucchitta, I., "The Shivwite Plateau", Application of ERTS Images and Image Processing, JPL Technical Report 32-1597, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, May 15, 1975, page 1.

Minnaert, M., "The Reciprocity Principle in Lunar Photometry", Astrophysics Journal, Vol. 93, pp. 43-410.

Mitchell, Richard J., "Surface Shadow Generation for the Preparation of Synthetic Imagery", Optical Engineering, Vol. 20, No. 3, pp. 446-449.

Phong, Bui Tuong, "Illumination for Computer Generated Pictures", Communication of the ACM, Vol. 18, No. 6, June 1975, pp. 311-317.

Post, Austin, Meier, Mark F., and Mayo, Lawrence R., "Measuring the Motion of the Lowell and Tweedsmuir Surging Glacier of British Columbia, Canada", ERTS-1, A New Window On Our Planet, Geological Survey Professional Paper 929, Richard S. Williams, Jr. and William D. Carter, editors, United States Government Printing Office, Washington, DC, 1976, pp. 113-116.

Potter, John F., "Correction of LANDSAT Data for the Effects of Haze, Sun Angle, and Background Reflectance", Annual Symposium on Machine Processing of Remotely Sensed Data, 4th, Purdue University, West Lafayette, Indiana, June 21-23, 1977, Published by IEEE, New York, NY, 1977, pp. 24-32.

Smith, J.A., Lin, Tzeu Lie, Ranson, K.J., "The Lambertain Assumption and LANDSAT Data", Photogrammetric Engineering and Remote Sensing, Vol. 46, No. 9, September 1980, pp. 1183-1189.

Temps, Ralph C. and Coulson, K.L., "Solar Radiation Incident Upon Slopes of Different Orientations", Solar Energy, Vol. 19, pp. 174-184. Van Nostrand's Scientific Encyclopedia, fifth edition, Published by Nan Nostrand Reinhold Company, New York, NY, 1976.

Venkataramanan, D., "Some Application of LANDSAT Imagery Interpretation for Petroleum Targetting in India", Proceedings of the International Symposium of Remote Sensing of the Environment, 13th, Vol. 2, University of Michigan, Ann Arbor, Michigan, April 23-27, 1979, pp. 911-924.

Viollier, M. and Baussart, N., "Enhancement of LANDSAT Imagery for the Monitoring of Coastal Waters, Waters, Application to the Southern Part of the North Sea", Proceedings of the International Symposium of Remote Sensing of the Environment, 13th, Vol. 2, University of Michigan, Ann Arbor, Michigan, April 23-27, 1979, pp. 10-93-1106.

Wang, S., Elliott, D., Campbell, J., Ehrich, R.W., and Haralick, R.M., "Spatial Reasoning in Remotely Sensed Data", To be published in IEEE Geoscience and Ellectronics, 1982.

Welch, R., Lo, R.C., and Pannell, C.W., "Mapping China's New Agricultural Lands", Photogrammetric Engineering and Remote Sensing, Vol. 45, No. 9, September 1979, pp. 1211-1228.

Appendix A

DOCUMENTATION FOR GIPSY COMMANDS USED

*ABSIMG Absolute value of an image.

VERSION: A.Ø1 DATE: Ø1-27-8Ø AUTHOR: OSCAR ZUNIGA

ACTION: This command operates on a standard image file with at least one numeric band and creates another standard image file in which every pixel is the absolute value of the corresponding pixel in the input image.

SOURCE: Disk, input file name.
DESTINATION: Disk, output file name.

FLAGS: None.

QUESTIONS: None.

COMMAND STRING EXAMPLE:

ABSIMG GIRLØ1.ABS < GIRLØ1.SIF

The output image GIRLØ1.ABS will be the absolute value of the input image GIRLØ1.SIF, pixel by pixel.

*ARITHM Add, subtract, divide, multiply - arithmatic operations.

VERSION: A.Ø1 DATE: Ø1-3Ø-81 AUTHOR: D.WHITEHEAD

ACTION: This command performs all arithmetic operations on either two images of similar size or on one image and a constant. The arithmetic operations implimented are: addition, subtraction, multiplication, and division. All operations are on a pixel by pixel basis.

SOURCE: Disk, input file names. (one or two)
DESTINATION: Disk, output file name.

FLAGS: (A) If the A flag is set, the addition operation will be performed.
(B) Process all the numeric bands.
(C) Copy over symbolic bands (if any).
(D) If the D flag is set, the division operation will be performed.
(I) If used then the true image min/max is not determined.
(M) If the M flag is set, the multiplication operation will be performed.
(S) If the S flag is set, the subtraction operation will be performed.
(T) Process all the symbolic bands too.

QUESTIONS: (1) Number of bands to operate on.
(2) Band numbers which will be operated on.
(3) Operation to be performed (if no flag is set).
(4) If the user wants to change the number of bits.
(5) The number of bits the user wants (if question 4 is answered y).

COMMAND STRING EXAMPLE:

```
ARITHM OUTPUT.SIF < CHECKER.SIF

SPECIFY ARITHMETIC OPERATION TO PERFORM (A/D/M/S)
     -- M
IMAGE CHECKER.SIF HAS 2 NUMERIC BANDS
HOW MANY BANDS DO YOU WISH TO SELECT ( >Ø
     ) -- 2
SELECTION 1 ( 1 - 2 ) -- 1
```

```
SELECTION 2 ( 1 - 2 ) -- 2
ENTER 2 CONSTANTS FOR ARITHMETIC OF IMAGES.
CONST( 1) = 3
CONST( 2) = 2
DEFAULT NUMBER OF BITS = 4 SCALING FACTOR
     = 1
DO YOU WISH TO DECREASE NUMBER OF BITS? (Y/N)
     -- N
```

ALGORITHM: For calculating the number of bits for the division operation :

NUMBER OF BITS=IFIX(LOG TO THE BASE 2 OF THE NUMBER) + 1 Where number is the max. value found in the image.

*BINSIF Convert binary sequential file into SIF file.

VERSION: A.Ø1 DATE: Ø1-27-8Ø AUTHOR: JOHN WRIGHT

ACTION: This command takes a binary sequential file and converts it to a standard image file. The input file record length is assumed to be equal to the input file block size. Each record in the input sequential file becomes a row in the output SIF file.The input record is considered to be a contiguous string of bite and is compacted into K bit grey tone intensity values where K is a user specified parameter.

SOURCE: Disk, input file name.
DESTINATION: Disk, output file name.

FLAGS: None.

QUESTIONS:(1) The number of records and number of points across.
(2) The number of leading bits to skip.
(3) The number of trailing bits to skip.
(4) The number of bits per data point.

COMMAND STRING EXAMPLE:

```
BINSIF GIRLØ1.SIF < GIRLØ1.DATA

ENTER THE NUMBER OF RECORDS AND NUMBER
  OF POINTS ACROSS-- 1ØØ, 1ØØ
ENTER THE NUMBER OF LEADING BITS TO SKIP--
  Ø
ENTER THE NUMBER OF TRAILING BITS TO SKIP--
  Ø
ENTER THE NUMBER OF BITS PER DATA POINT--
  8

The binary sequential input file GIRLØ1.DATA
  is converted to the standard image format
  file GIRLØ1.SIF.
```

*BORCHN Create border segments from symbolic image

VERSION: A.Ø1 DATE: 11-Ø1-81 CONTACT: SHYUAN WANG

ACTION: For an input symbolic image with regions, this command creates the property file and chain code file for the border segments of the regions. The border segment is defined as one unit if it is continuously between two regions and only these two regions. If G flag is set, the border segment will be split into pieces corresponding to the north, south, east, and west borders of a region.

SOURCE: Disk, input file name
DESTINATION: Disk, output file name

FLAGS: (G) the border segment will be split into pieces corresponding to the north, south, east, and west borders of a region. This is for geography or remote sensing purpose.

QUESTIONS: Which band to be used if the # of bands is more than one.

COMMAND STRING EXAMPLE:

BORCHN SYMBOLIC.SIF > BORCHN.PRP, BORCHN.CC

Given the symbolic image SYMBOLIC.SIF, the command will create the border segment property file BORCHN.PRP, and chain code file BORCHN.CC.

COMMENTS: After the command is completed, use PRTPF to get a listing of the property file, and use PRTCC to get a listing of the chain code file. Command CCSIF will create the border segments symbolic image from the property file and chain code file so that you can look at them either from the printout or at the display. The property file contains: (1) I.D. # (2) - (3) beginning pixel row # (4) beginning pixel col. # (5) - (6) ending pixel row # (7) ending pixel col. # (8) left region (9) right region (1Ø)

pointer to chain code (11) length of chain code (12) - (13) - (14) - (15) - (16) border orientation code (17) border overlay data (18) link at beginning pixel (19) link at ending pixel. You will get (16) to (19) only when G flag is set. The exact beginning and ending positions of the border segment are at the upper left corners of the pixels specified by row # and column #.

*CHNSIF Create border segments image from chain code and prop. files

VERSION: A.Ø1 DATE: 12-29-81 CONTACT: SHYUAN WANG

ACTION: This command accepts the two output files from command BORSEG and creates a symbolic SIF image where each border segment is represented by continuous pixels which have the same I.D. # of that border segment as their values. If border is between upper and lower pixels, I.D. # is put into lower pixel. If border is between left and right pixels, I.D. # is put into right pixel. Normally, the output image is a single band image containing all the border segments from the inputs. However, if P flag is set, the output image will be a multi-band image for a user specified property, and each band contains only the border segments having a user specified value of that property. Please see the example below.

SOURCE: Disk; (1) input chain-code file,
(2) input property file.
DESTINATION: Disk; output file name.

FLAGS: (A) The min. and max. I.D. # of border segments in the header of property file will be used as the lowest and highest pixel values in the output image; otherwise, question will be given for these values.
(P) Multi-band image will be created for a specified property and some specified values.

QUESTIONS: 1. (P flag) Select one property.
2. (P flag) How many bands do you want for this property ?
3. (P flag) Specify property value for each band.
4. (No A flag) What are the lowest and highest I.D. # of border segments in the output ?

COMMAND STRING EXAMPLE:

```
CHNSIF BORDER.PRP, BORDER.CHN > BORDER.SIF
     (P)

ENTER PROPERTY # (D = 1, 1 - 19) -- 16
ENTER # OF BANDS (1 - 3) -- 2
FOR BAND 1 ENTER PROPERTY VALUE (D = 1, 1
     - 3) -- 1
FOR BAND 2 ENTER PROPERTY VALUE (D = 1, 1
     - 3) -- 2
ENTER LOW LABEL (D = 1, 1- - 250) -- 1
ENTER HIGH LABEL (D = 250, 1 - 250) -- 100
```

For border segments property file BORDER.PRP, and chain code file BORDER.CHN, the output image BORDER.SIF will be a 2-band SIF image. For border segment whose I.D. # is between 1 and 100, if its 16th property value is 1, the corresponding pixels will be put into band 1 of BORDER.SIF, and if its 16th property value is 2, the corresponding pixels will be put into band 2 of BORDER.SIF.

*CNSFCT Central neighborhood sloped facet

VERSION: A.Ø1 DATE: Ø6-2Ø-8Ø AUTHOR: OSCAR A ZUNIGA

ACTION: This command estimates the parameters of a linear gray tone surface for each pixels central neighborhood. The neighborhood is rectangular and its size is specified by the user. A linear least squares fitting is done inside this neighborhood and a multiband output image is created containing the fitted graytones, parameters and errors.

SOURCE: Disk, input file name
DESTINATION: Disk, output file name

FLAGS: (E) if set the actual errors are computed, otherwise
$\log_2 (1 + \text{error})$ is computed

QUESTIONS: (1) Neighborhood size
(2) number of bits for the output file

COMMAND STRING EXAMPLE:

```
CNSFCT GIRLØ1.CN3 < GIRLØ1.SIF

ENTER WINDOW SIZE FOR GIRLØ1.SIF
ROW BY COLUMN ( 1- 256 ), ( 1 - 256 ) --3,3
ENTER NUMBER OF BITS ( 12 - 17 < D=14 > ) --17
```

A 3 x 3 neighborhood is used to perform the filtering operation on the input SIF 'GIRLØ1.SIF'. The output SIF 'GIRLØ1.CN3' will consist of 4 bands: the output graytone band (also the independent term in the fitting parameters); the row slopes band, the column slopes band, and the log of the fitting errors band.

ALGORITHM: Let (NR, NC) be the size of the neighborhood used, ie, number of rows by number of columns. A linear fitting is performed inside this neighborhood according to the model:

$$X(R,C) = A*R + B*C + G + E(R,C)$$

$$Y(R,C) = A*R + B*C + G$$

Where X(R,C) is the input graytone at the (R,C) position inside the neighborhood, A is the horizontal slope, B is the vertical slope, G is the independent term, and E(R,C) is the fitting error. Y(R,C) is the result of the fit at the (R,C) position. The value of the fit at the center of the neighborhood, ie Y(Ø,Ø), along with the fitting parameters A and B, and the sum of the square errors are assigned to the output SIF. Scaling is done in the fitting parameters and in the square errors to keep maximum accuracy. The scaling factors are stored in the descriptor records and they represent the ratio of stored value to actual value.

*CONCT Labels maximally connected components of a symbolic image.

VERSION: A.Ø1 DATE: 12-17-8Ø AUTHOR: A.SINGH

ACTION: CONCT labels the maximally connected components of a symbolic image with distinct labels. If there is more than one symbolic band CONCT will act on all of the symbolic bands specified. The input file bands must be binary symbolic bands in line form and integer mode. The output image will be a multi-band image depending on the number of bands specified by the user. The output image is generated in line form and integer mode. All components having the user specified mark input label will be maximally connected and will be assigned a unique output label.

SOURCE: Disk, input file name.
DESTINATION: Disk, output file name.

FLAGS: (A) Connect all symbolic images.
(B) Prompts the user for the starting output label (default = 1).
(F) Must be set for four-connected components.
(M) Prompts the user for the mark input label to conect (default = 1).
(N) Output file is to be compressed if possible.

QUESTIONS: Bands to be processed if more than one 'A' flag is not set.

COMMAND STRING EXAMPLE:

```
CONCT INPUT.SIF > OUTPUT.SIF

IMAGE INPUT.SIF HAS 4 SYMBOLIC BANDS
HOW MANY BANDS DO YOU WISH TO SELECT (>Ø)--3
SELECTION 1 — 1
SELECTION 2 — 2
SELECTION 3 — 3
```

Connected regions on bands 1, 2, and 3 will be labeled. OUTPUT.SIF will be a 3 symbolic band image. No questons would have been asked if there was only one symbolic

image or if flag A had been set. Flag N causes the output file to be copied over using the smallest number of bits per pixel.

```
*CPYRPL Copies a file and replaces some pixel
        values if flags are set

 VERSION: A.Ø1 DATE: Ø7-2Ø-81 CONTACT: D.B. ELLIOTT

 ACTION: Copies the first input file into the
        output file and option- ally replaces
        the value of some pixels according to
        the flags that are set and the other
        input files given.

 SOURCE: 1 , 2 , or 3 disk input SIF file depending
        on the flags set.
 DESTINATION: 1 output SIF disk file.


 FLAGS:   (F) F flag means that pixels in first
        file will be replaced by those pixels
        in the second file which correspond
        to a '1' in the third file.
        (C) C flag allows user to change the
        JDENT array for the output file.
        (H) H flag means that pixels in the
        first file are replaced by pixels in
        the second file if the value of the
        pixel in the second file is >= large.
        if the R flag is set, the pixel is replaced
        if its own value >=large
        (L) L flag means that pixels in the
        first file are replaced by pixels in
        the second file if the value of the
        pixel in the second file is <= small.
        if the R flag is set, the pixel is replaced
        if its own value <=small
        (R) R flag means reverse the test in
        the L and H flags.

 QUESTIONS: None

 COMMAND STRING EXAMPLE:

 CPYRPL INPUT.SIF,VALUES.SIF,BINARY.SIF > OUTPUT.SIF
        (CF)
              :
              :
         PRINT OUT OF JDENT ARRAY
              :
    DO YOU WANT TO CHANGE ANY VALUES ( Y/N)?
    Y
    ENTER THE (JDENT LOCATION , VALUE):
```

15 2

The input file will be copied to the output file and each pixel which is marked in the corresponding position on BINARY.SIF by a '1', will take on the value of the coresponding pixel in the VALUES.SIF. Lastly, the the user has changed JDENT(15) to the value of 2.

COMMENTS: NOTE: The CPYRPL command always sets the JDENT array of the output file to reflect the true mininum and maximum of the output image. The user may change the JDENT values by using the C flag , but be warned that setting inconsistant values in the JDENT array may cause an error.

*DSPLY Displays an image in the standard image format on the light.

VERSION: A.Ø2 DATE: Ø2-Ø9-81 AUTHOR: P.MULGAONKAR

ACTION: This command displays the contents of a standard image file on the LIGHT display.

SOURCE: Disk, input file name
DESTINATION: LIGHT display.

FLAGS:(A) Asks the user for a portion of the image to be displayed. The default is the full image if it will fit on the LIGHT screen (which is 48Ø X 48Ø). If the image is larger than that, the user is prompted for the size of the image to be displayed by default . If the X flag is also set, the specified size corresponds to the screen size of the image when dsplayed. The sampling factor is set by the skipping ratio specified through the X flag.

(B) Forces the image to be black and white. The default for numeric images is black and white. The default for symbolic images is color.

(C) Clears the screen before displaying the image.

(E) Defines the image to be an edge image produced by MRKEGS. Also enables the overlaying of this image on an existing image already on the screen.

(I) Inverts the image. Default low numbers are darker than the high numbers. The I flag will cause the low numbers to be brighter.

(L) Location indicator flag. This flag treats the entire screen area as consisting of a number of pages each of which can fit one image of the size being currently displayed. The pages are conceptually organised as a rectangular array and the user is asked for the row and column in this page array on which the picture

is to be displayed.

(M) Maximum-minimum flag forces the command to use the actual maximum and minimum in the image for scaling purposes and not the default of the max and min values in the ID block.

(N) No color/black-and-white table is output. Any color table previously in effect is retained.

(O) Overlay flag. This flag divides the displayed color area into two parts - a black and white section and a color section. Numeric images are displayed in black and white with 240 gray levels. Color images are displayed in color with 10 color levels. Symbolic images may be put on top of black and white images (but not vice versa). the RFLAG changes this color set up by making numeric images color and symbolic images black and white.

(P) Position flag prompts the user for a row column position on the screen for the origin of the image. Unlike the L flag which forces the image to a fixed grid of pages, the user has full control over the exact placement of the image on the screen. This flag is incompatible with the L flag and yields precedence to the L flag. In the absense of the P or L flags, the image is centered on the screen.

(R) R flag reverses the color distribution of the O flag. See the OFLAG documentation.

(S) Prevents the scaling of the values in the image. If the image is numeric, the user is prompted for a starting value and 256 values from that point are displayed. The starting value maps to 0 and the one after it to 1 and so on. If however, the image is symbolic, a random wrap-around mapping of values is done to ensure that no two adjacent numbers get mapped to the same value

on the screen.

(X) Extract flag: If this flag is specified alone, the user is prompted for the number of pixels to skip for each pixel displayed. This sampling factor is applied equally to both the row and coloumn directions to maintain the aspect ratio. If the A flag is also set, the questions asked through the A flag, decide the finished size of the image and consequently the maximum possible compression ratio. The user is asked to specify the compression ratio. NOTE that a few lines or columns may not be displayed if the image size in the corresponding direction is not an integral multiple of 1+ the number of pixels to skip.

QUESTIONS: FLAG QUESTIONS AND EXPLANATION

(A) HOW MANY ROWS ? Enter the number of rows of the image to be displayed.
HOW MANY COLUMNS? Enter the number of columns.
ENTER STARTING ROW IN IMAGE.
ENTER STARTING COLUMN IN IMAGE.
Enter the starting row and column coordinates in the image.
(L) ENTER PAGE ROW. Enter the row of the page
ENTER PAGE COLUMN. and the column in which the image is to be displaye.
(P) ENTER THE ROW ORIGIN. Enter the row and column
ENTER THE COLUMN ORIGIN position on the screen for the image origin.
(S) ENTER THE START VALUE This question is asked for a numeric image with the no scale option and if the number of levels in the image exceed 256. The levels below this value are not displayed.

(X) ENTER THE NUMBER OF PIXELS TO SKIP For each pixel displayed, this many pixels/rows will be skipped. The user is also asked for the starting point in the image. (See A flag)

The output file name can be left out and defaults to LIGHT. This may change

if more display devices are added to the system in the future.

COMMAND STRING EXAMPLE:

```
DSPLY LIGHT < CHK.SIF (CB)
```

DSPLY: command name.
LIGHT: output display device name - must be previously defined. CHK.SIF:Input image file.
(CB) Flags: C flag will clear the screen before displaying the image. The B flag will force the image to be black and white.

COMMENTS: The device for the display (LIGHT) must be allocated and assigned to the logical name LIGHT before the command is invoked.

*DELETE Delete a specified SIF file.

VERSION: A.Ø1 DATE: Ø5-16-81 AUTHOR: Y. YASUOKA

ACTION: This command deletes one sif file specified by a user.

SOURCE: Disk, input file name
DESTINATION: None

FLAGS: None

QUESTIONS: None

COMMAND STRING EXAMPLE:

(1) DELETE CHK.SIF

The specified file CHK.SIF is deleted.

*DSTRPE Destripe the striped image from multi-sensor.

VERSION: A.Ø1 DATE 5-25-81 AUTHOR: JONGSOO LEE

ACTION: This command destripes the satellite images obtained with multiple image sensors which are not identical in the transfer function. The number of bands to be destriped are taken by the user.

SOURCE: Disk, input file name
DESTINATION: Disk, output file name

FLAG: None

QUESTIONS: (1) Number of bands to destripe
(2) Band numbers to destripe

Output of the destripped image bands is stored in sequence from the band number 1 to the number of bands destripped, in the order that the band is selected and destripped. This part is inserted newly by the UPDATE (8-19-81 BY J. Lee).

COMMAND STRING EXAMPLE:

DSTRPE B73A.DSP < B73A.TRA

ALGORITHM:

The method assumes that each sensor is exposed to scene radiances with approximately the same probability distribution. The sensor values can be modified so that each one is related in the same way to actual scene radiance. This modification can be done using the cumulative histograms of the whole image data and each sub-image from each sensor. Let H(x) be the occurrences of sensor outputs less than or equal to x out of a total of N values and let Hi(x') be the number of sensor outputs less than or equal to x', produced by sensor i, out of a total of Ni values. Here,

$$N = N_1 + N_2 + \ldots + N_n$$

where n is the total number of sensors.

Then, for a sensor value x' of sensor i, we assign x which satisfies

$$Ni * H(x) <= N * Hi(x') < Ni * H(x+1).$$

COMMENTS: The algorithm can be easily understood if we think the cumulative histogram as an inverse transfer function of each sensor.

*ELIMRG Eliminate regions on a symbolic image.

VERSION: A.Ø1 DATE: Ø3-28-81 AUTHOR: T.C.PONG

ACTION: This command creates a symbolic image in which pixels in a region having a specified property with values greater (or smaller) than a threshold input by the user are set to zero. The FILL command can then be used to fill the zeroed pixels with the label of a labeled pixel closest to it.

SOURCE: Disk; image file name and property file name

DESTINATION: Disk; symbolic image file name.

FLAGS: (A) Take the absolute value of the property values before comparing with the threshold.
(G) Set the pixels in a region to zero if its region property value is greater than the threshold. The default is pixels in a region are set to zero where the region property is less than the threshold.
(M) The user can select more than one properties. Set the pixels in a region to zero if its region property values are within the selected property ranges.
(R) The output symbolic image will be re-rabelled to positive sequential labels.

QUESTIONS: The user will be asked to enter the property number(s) to be looked at, and will also be asked to input a threshold.

COMMAND STRING EXAMPLE:

```
ELIMRG OUTPUT.FIL < IMAGE.SYM,PROPERTY.FIL

ENTER PROPERTY NUMBER (1 - 26): 4
ENTER THRESHOLD FOR PROPERTY 4: 1ØØ.
```

Regions in IMAGE.SYM with property four found in PROPERTY.FIL less than 1ØØ. are set to zero.

*EXDSP Examine displayed image.

This is a complete subsystem which allows the user to examine an image displayed on the LIGHT display device. The format of most of the subcommands is as follows:

SUBCOMMAND <parameters>

The subcommand allways has four characters-- truncations or abbreviations are not acceptable (in version 1 anyway.)

Parameters are (in most commands) optional and if they are left out, the system will report the value of the parameter controlled by the message.

If the parameter is a required parameter, the system will prompt for the value to be used.

What follows is a definition of the syntax of each of the subcommands followed by a brief explanation of what it does.

COMMANDS MODIFYING THE COLOR TABLES.

BLAK SET BLACK AND WHITE COLOR TABLE.
COLR SET COLOR TABLE TO 256 COLOR LEVELS.
USER INVOKE USER DEFINED COLOR SPACE.

The image is actually a 480x512 block of numbers. Each number has,associated with it, three parameters defining the amount of red, greenand blue to be displayed. This translation is stored in what is calledthe VLUT (Video LookUp Table). The Light display has some defaultVLUTs. BLAK sets the VLUT to be the default black and white (greytone)look up table. COLR defines the VLUT to be the default 256 color levelchart. EXDSP also provides for a user VLUT which is invoked by the commandUSER. Note that the VLUTs affect the entire picture uniformly.

COLOR BAR MODIFICATION COMMANDS

STRT <N> SET START POSITION OF COLOR BAR.

```
 UPER  <N>     SET UPPER END OF COLOR BAR.
 WDTH  <N>     SET WIDTH OF THE COLOR BAR.
 CBAR  <ON,OFF,R,G,B,W,C>  SET COLOR BAR TYPE.
(More information about R,G,B,W,C is given below.)
 INCR       INCREMENT START OF COLOR BAR.
 DECR       DECREMENT START OF COLOR BAR.
```

The user may want to examine portions of the image to determinethe actual numbers that make up portions of the display. The color baris one mechanism that permits the user to do just that. The color bardefines a block of numbers which take on a different color from the restof the image and the user can move this block around enabling him tosee the portions of the image which lie in this range. The backgroundimage (whose numbers lie outside this range) is displayed in black andwhite, and the portions in the range are displayed depending on thetype of color bar selected. The start of the bar is set by the STRTcommand and defaults to 1Ø. The width is defined by WDTH and also defaultsto 1Ø. The upper end then defaults to 19 but may be modified by UPER.Ofcourse the three are related by UPER = STRT + WDTH - 1.If any of these commands is entered without a numeric parameter, thecurrent value of the relevant parameter is reported.

The color bar is selected by the CBAR command. CBAR takes oneparameter which defines the colorbar operation to be performed.Only the first character of the parameter is examined. The parameters maybe: ON, OFf, Color, White, Blue, Red, Green. (minimum truncation is capitalised.)

ON turns on the color bar of the last specified type at the current values of start and width. (default type is color)
OFf turns off the color bar.
Color defines the color bar to be multicolored.
White defines a white color bar
Blue defines a blue color bar
Red defines a red color bar
Green defines a green color bar.

All the parameters (except OFF) also turn on the color bar. When thecolor bar is ON, any changes to Start, width or upper end of the bar,are immediately reflected on the display. If the bar is off, the changeis made but not displayed untill the next CBAR command turning on the bar.INCR and DECR move the color bar's starting position without alteringthe width and can be used to scan the picture.

Note that the Color bar can be used to threshold images. For exampleTo set every number above x to white and the rest to some arbitrary grey tones.use:

```
STRT x
UPER 255
CBAR W
```

ZOOM COMMANDS. (MAGNIFICATION)

ZOOM <IN,OUT> ZOOM IN OR OUT OF THE PICTURE.

The ZOOM command changes the magnification at which thepicture is displayed. ZOOM In doubles the magnification and ZOOM Outhalves it. Magnification is restricted to the range 1-8 and any attemptsto ZOOM beyond this range, are simply ignored.Entering ZOOM without any paramter, reportsnored.Enterithe current magnificationat which the image is being displayed.

COMMANDS WHICH MOVE THE PICTURE AROUND

```
RGHT      MOVE PICTURE TO THE RIGHT.
LEFT      MOVE PICTURE TO THE LEFT.
UPWD      MOVE PICTURE UPWARDS.
DOWN      MOVE PICTURE DOWNWARDS.
HOME      MOVE PICTURE TO HOME POSITION.
STEP <N>    SET STEP SIZE FOR PICTURE MOVEMENT.
```

The user can "move" the picture around and see different partsof it. When the magnification is 1 ofcourse the entire pictureis visible. However when the user has zoomed into the picture, only apart of the image is visible, and the picture has to be moved around tobe

able to examine different portions of it.

RGHT, LEFT, UPWD and DOWN move the picture by a predetermined amountin the specified direction relative to the viewer. Whenever the usermoves the picture to a position such that the window goes outside thebounds of the image, a warning message is displayed advising the user tozoom in.

The step size for the movement is determined by the STEP command.The default for the step is 60 pixels. which enables the user to scanthe entire picture in 8 steps per direction at full zoom. Entering STEPwithout parameters, reports the current step size.HOME moves the window origin back to the origin of the picture (Top right)

CHANGES TO THE USER VLUT (COLOR TABLE)

ENBL	ENABLE MODIFICATIONS TO USER COLOR TABLE.
DSBL	DISABLE MODIFICATIONS TO USER COLOR TABLE.(DEFAULT.
CHNG	CHANGE RGB VALUE IN SPECIFIED USER TABLE LOCATION.
SHOW	<N> SHOW RGB VALUE IN SPECIFIED USER TABLE LOCATION.
ZERO	SET ALL USER TABLE VALUES TO ZERO.
DEFN	DEFINE A DEFAULT USER COLOR TABLE.
BACK	BACK UP ONE CHANGE.

The User color table (as mentioned earlier) keeps the colors theuser defines for each numbers. The User color table is initialised tocontain special values for the RED, GREEN and BLUE components and thesevalues are recognized by the system and not displayed when the User tableis displayed. Any changes to the entries in the user table then overwritethese special flag entries and are then displayable. This enables a user tomake selected modifications to any existing color space (like the systemcolor space (invoked by the command COLR) and store just these changesin the user color space. To re display this "modified" color space, theuser can enter the commands

COLR followed by USER. Only the colors explicitlymodified by the user, will change, the rest remaining as they were.The ZERO command however puts all entries in the user space to BLACK.So only any changes after that are visible on the screen (all othersbeing invisible) and can therefore be used to selectively mask out certainregions and retain the rest.The user space is protected from accidental changes by an internal flagwhich can by set by the call to ENBL (enable changes) and reset byDSBL (disable changes). Any invocation of USER automatically permitsthe modification of the user space untill such a time that an explicitDSBL is given or a system default color space is invoked (by a commandlike BLAK or COLR).

Since this is meant to be a primarily interactive kind of a procedurethe user is expected to make several iterations before he arrives at someacceptable color chart. To help in this, there is a builtin automaticsave area. This saves the values of the RGB components in the user spacewhenever a CHNG command is issued. Only those values which are beingmodified, are saved. These values may be retrieved by a command BACKwhich backs up the changes one at a time. Any old state may then be retrievedin the reverse order of the changes made. The size of the save area is finite and therefore the number ofold entries that can be retrived, limited (although the limit is largerthan will usually be needed).The SHOW command shows (at any time) the values of the specifieduser location.

EXTERNALLY CREATED USER COLOR TABLES.

LOAD <FILENAME> LOAD USER COLOR TABLE FROM FILE.
STOR <FILENAME> STORE USER COLOR TABLE ON FILE.

These commands enable the user to store user color tables on diskfiles and then retrieve them for later use. The files

may be created by usingthe system editors but have to have the following format:

```
Record 1: Upto 7Ø characters of identifying information
Record 2:
   :
   :    Color number, red, green, blue (color
          number between Ø and
   :    255, red green and blue between Ø and
          999. separated by
   :    blanks or commas.
Record n  -do-
```

All the 256 colors need not be defined. Only those mentionedare changed, the rest remaining unchanged when the color space is loaded.On giving the command STOR, the system asks the user for one line ofcomments which are then stored as the first line of the output file.These comments are then displayed whenever the file is LOADed.

GENERAL UTILITY COMMANDS.

INIT	REINITIALISE DISPLAY PARAMETERS.
INFO	SHOW CURRENT DISPLAY STATUS.
SVOC	SHORT VOCABULARY OF COMMANDS.
RAMP	DISPLAY A VERTICAL RAMP.
DONE	LEAVE EXDSP COMMAND MODE.

These commands are self explanatory and will not be discussedin any great detail.

*EXSIF Examine standard image file.

VERSION A.Ø1 DATE: 1Ø-15-79 AUTHOR: SCOTT KRUSEMARK

ACTION: This is an SIF editor which allows examination and modification of any pixel within an SIF file. This editor has its own command language which is designed to facilitate access to all parts of an SIF file. This allows users to check results of their programs during debugging and to get printouts of the file when necessary. The editor is capable of handling two files at once, making comparisons between input and output files easy. These files are designated as file A and B.

SOURCE: None
DESTINATION: None

FLAGS: None

QUESTIONS: The following commands exist under the EXSIF subsystem:
A — Switch to file A
B -- Switch to file B
BLK — Print parts of image
BRIEF -- Short or long form
CLOS -- Close a file
COL — Set column limits for BLK command
DONE — Exit form EXSIF back GIPSY command level
FIND — Locate a value in image
FORM -- Set format for BLK command
IMG — Set image number for all operations
INFO -- Provide information on current file
MID — Modify identification record of SIF file
MOD — Modify image data
OPEN -- Open file
OUT -- Set output device (printer or terminal)
PROT — Set/clear protection
REPL — Replace all occurrences of value with another
ROW -- Set row limits for BLK command
SVOC -- Short vocabulary listing of EXSIF commands

TOP -- Move image pointers to top of image (REPL,MOD)

By appending the letter (A or B) to the end of the command the switch is make automatically to that file before the subcommand is executed.

COMMAND STRING EXAMPLE:

EXSIF

This invokes the subsystem EXSIF. For more information on EXSIF and an example of its use, please see external documentation.

*INTERP Perform a mask operation with optional boundary points

VERSION: A.Ø2 DATE: Ø8-18-81 CONTACT: D.B. ELLIOTT

ACTION: This command allows the user to define an MxN mask operator an apply it to the input SIF file. If the B-flag is set, the user must include a second input file (a binary file) which indicates which pixels from the first input file are to be treated as boundary points (i.e. there values will not be changed by the mask operator, though the values they have will be used in operations on other (non-boundary) pixels. The mask operator can be applied iteratively for any number of iterations (user specified) or until the maximum difference over all pixels between any two iterations falls below a user-specified threshold. After the user indicates the number of iterations or the threshold and the task has been completed, a report is printed and the user is permitted to request that more iterations be performed, or he may quit

SOURCE: Disk; Input file name
DESTINATION: Disk, (REPORT IS PRINTED TO TERMINAL)

FLAGS: (B) Allows user to specify boundary points whose values will not be changed by the operation.
(R) Allows the user to enter a relaxation factor to implement SOR (successive overrelaxation) iteration.

QUESTIONS: 1. Size of the mask (operator grid).
2. Actual values of the mask (operator grid).
3. Is the grid that you just entered correct?
4. Enter relaxation factor (ONLY IF Rflag IS SET).
5. Enter the number of iterations and the cutoff value for the maximum difference between iterations.
6. Do you want to go again?

COMMAND STRING EXAMPLE:

```
INTERP TEST.SIF, BINARY.SIF > TESTED.NEW
(B)


Enter the size of the grid (rows , cols ).
  3 3
Enter row 1 of the grid:
  Ø 1 Ø
Enter row 2 of the grid:
  1 -4 1
Enter row 3 of the grid:
  Ø 1 Ø
The grid you specified is:
     Ø 1 Ø
     1 -4 1
     Ø 1 Ø
 Is this correct? (Y/N)
  Y
Enter number of iterations and maximum difference:
  2Ø , 5.


CONVERGENCE WAS OBTAINED BEFORE ITERATIONS
     COMPLETED.

Iterations requested 2Ø max. diff requested
     5.
Iterations performed 18 max. diff. calculated
     4.9987
45Ø2 points were calculated, average difference
     = .123455E+Ø1
Do you want to go again? (Y/N)
  N

 The operator shown above (Laplacian = Ø
     equation) was applied to all pixels
     in the image TEST.SIF except those coresponding
     to a '1' in the BINARY.SIF for 18 iterations.
     Because the pixel which changed the
     most between the 17th and 18th iterations
     had a difference between the two iterations
     of 5 (the cutoff value specified by
     the user), the program stopped and printed
     a report.
```

COMMENTS:

Using the digital Laplacian mask (as in the above example) with this command is the same as Gauss-Siedel iteration. The use of the Rflag enables the user to do SOR iteration which will produce better (faster) results IF the correct relaxation factor is used for that problem.

*INTRPD Interpolates elevations between ridges and valleys

VERSION: A.Ø2 DATE: 11MAY82 CONTACT: D.B.Elliott

ACTION: The input files tell the distance to the nearest ridge and valley for every pixel. Also, either the elevation of the closest ridge and valley pixel is given for every pixel, or a constant elevation is given for all ridge and valley pixels The output image gives an elevation value for every pixel. This is calculated by doing a linear interpolation between the elevations of the nearest ridge and valley pixels. If the Cflag is set, a cubic polynomial interpolation is done, with the constraints that the slope at all the ridge and valley pixels is zero and that the elevation values at the ridge and valley pixels is given as in the linear case.

SOURCE: DISK, 2 input files
DESTINATION: DISK, 1 output file

FLAGS: (V) The user includes 2 bands on each input file. The first band for each file is the distance to the nearest valley pixel (file 1) and ridge pixel (file 2). The second band on each input file is the elevation of the nearest valley (file 1) and ridge (file 2) pixel.

(C) A cubic polynomial interpolation with the constraint that the ridge and valley pixels have slope Ø is done instead of the linear interpolation.

QUESTIONS:(1) The elevation value for the valley pixels and
(2) the elevation value of the ridge pixels is asked if Vflag is not set.

COMMAND STRING EXAMPLE:

```
INTRPD VALLEY.DST , RIDGE.DST > ELEVATION.SIF
  Enter valley elevation value: Ø
  Enter ridge elevation value :1ØØ
```

An elevation map with ridge pixels having

an elevation of 100 and valley pixels having an elevation value of 0 and all other pixels having elevations calculated by interpolating between the nearest ridge and valley pixels is output in ELEVATION.SIF.

*LNTRA Linear spatial transformation -- rotate, skew and scale.

VERSION: A.Ø1 DATE: Ø1-15-8Ø AUTHOR: SCOTT KRUSEMARK , GREG BANGE

ACTION: This command performs a spatial transformation on image data. It allows two modes where the user enters the 4 coefficient transformation matrix or the user enters the angles to rotate, and skew, and the scale factors.

SOURCE: Disk, input file name.
DESTINATION: Disk, output file name.

FLAGS: (A) Process all bands.
(C) Set vertical and horizontal scale factors to 1.Ø.
(F) Ask user for fill value (around edges).
(K) Set skew angle to Ø.Ø radians. (no skew).
(M) User wants to specify matrix (see algorithm below).
(P) User wants to square up the image before rotation.
(R) Set rotate angle to Ø.Ø radians.

QUESTIONS: (1) User is asked if the resolution cells are square
(2) User is asked to enter a skew angle
(3) user is asked to enter a rotate angle
(4) User is asked for horizontal and vertical scale factors

COMMAND STRING EXAMPLE:

```
LNTRA OUTFILE.SIF < INFILE.SIF

IS THE RESOLUTION CELL SQUARE (Y/N) -- Y
ENTER SKEW ANGLE ( IN RADIANS ) -- .2
ENTER ROTATE ANGLE ( IN RADIANS ) -- 1.2
ENTER HORIZONTAL AND VERTICAL SCALE FACTORS
     -- 1.5,1.5
```

This would rotate the image clockwise 1.2 radians with a skew (before rotate) of .2 radians. The result would be an image larger by 1.5 times the original size.

ALGORITHM: The data is transformed by the equation:

$$\begin{pmatrix} r' \\ c' \end{pmatrix} = \begin{pmatrix} e & f \\ g & h \end{pmatrix} \begin{pmatrix} r \\ c \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}$$

Where e,f,g,h are the entries of the linear transform matrix and r and c are the output row and column positions. r' and c' are the input row and column positions. The values a and b are internally generated to offset the image to remain in the positive quadrent.

*MHIST Image histograms.

VERSION: A.01 DATE: 01-15-80 AUTHOR: A.SINGH

ACTION: This command generates histograms of an integer SIF file. There are several formats of output that are available (1) bar graph (horizontal) (2) histogram (vertical) (3) output tables of counts and probabilities. The operations can be done in linear or log scales, and selected band can be supplied by user.

SOURCE: Disk, input file name.
DESTINATION: Terminal or printer.

FLAGS: (A) All bands of input file used.
(B) Bar graph output.
(C) Expanded mode for printer.
(D) No form feed in bar graph and table probabilities.
(E) Get first and last values to graph from user.
(H) Histogram output (default).
(I) Symbolic bands as well.
(K) Calculate grey level where between group variance over within group variance is maximized. This flag should not be used at the same time as the M-flag.
(L) Use log scale for histogram (linear-default)
(M) Compute true min/max from file.
(N) All user to enter name for each histogram.
(O) Omit some pixels from histogram.
(R) Non-zero values counted only.
(S) No shift in bar graph.
(T) Output tables of counts and probabilities.

QUESTIONS: (1) Bands to use.
(2) If E flag is set, low and high values to use as limits are asked.
(3) If N flag is set, title for band 1 is asked.
(4) If O flag, enter the two test values: slope test, Laplacian test.

COMMAND STRING EXAMPLE:

MHIST TT < IN SIF (MABHLT)

No questions

This example outputs all possible outputs.
It uses all bands that are on the input
file 'IN SIF'. The counts are based
on the log of the pixel values. The
image is looked at to get the true min/max
for graphing. The outputs are: (1) bar
graph, (2) histogram, and (3) the actual
counts.

```
*PQUANT Probability quantised image.

 VERSION: A.Ø1 DATE: Ø5-3Ø-81 AUTHOR: K. FlANNERY

 ACTION: PQUANT performs an equal, normal, or
         chi-square probability quantization
         on all numeric images within a SIF multi-image
         file. In addition, the user may input
         a table defining the quantized graytone
         value for every current graytone value.
         The user specifies one of the four distributions
         (the default is Uniform) and the input
         image is quantized to the specified
         number of levels. For normal and chi-square
         distributions an optimal least-squared
         error fit is used.

 SOURCE: Disk, input file name
 DESTINATION: Disk, output file name

 FLAGS: (A) Quantizes all numeric bands of the
          image
        (B) Performs a Normal probability quantization
        (C) Performs a Chi-square probability
          quantization
        (N) Causes Ident(15) of the input SIF
          to be used as the minimum gray level,
          and Ident(16) to be used as the maximum
          gray level
        (R) A sequential file is read in (INPUT
          FILE2) which defines the complete mapping
          between current graytone values to the
          new "quantized" values. The format of
          the sequential file is:
       RECORD 1 -- INTEGER which gives number
          of mapping table entries to follow
       RECORD 2 -- value1 quantized value1 (both
          INTEGERs)
          .
          .
          .
       RECORD N+1 -- valueN quantized valueN (both
          INTEGERs) The values in the left column
          of the table are the image graytone
          values starting with the minimum and
          incrementing by 1 to the maximum (If
          some values in this range are not represented
          in the image, they still must be in
          the table. Thus if the minimum of the
          image is 1Ø and the maximum is 2Ø, the
```

left column of the table must have 1Ø,11,12,13,14-
,15,16,17,18,19,2Ø.) --- The sequentiall,12,13,14-
file may be TT (terminal).
(S) Causes symbolic bands to be included
in the output image
(Q) Allows the user to specify the levels
to be used in the quantized image. If
not set, the levels range from Ø to
the number of quantized levels minus
1

QUESTIONS: (1) Number of quantized levels (integer,
>Ø)
(2) Relative mean, or "shift" relative
to the number of quantized levels /
2 (for Normal) (real)
(3) Standard deviation factor, a positive
real number used to multiply (number
of quantized levels)/6 (for Normal)
(4) Degrees of freedom (integer, >Ø)
(for Chi-Square)
(5) Number of equal intervals to include
(for the Q flag)

COMMAND STRING EXAMPLE:

```
PQUANT TEST1.SIF > TEST2.SIF (AB)

ENTER NUMBER OF QUANTIZED LEVELS FOR OUTPUT
     IMAGE(>Ø)-- 1Ø
ENTER RELATIVE MEAN(Ø IS CENTER)-- -2.1
ENTER SIGMA FACTOR(1 IS STANDARD)-- Ø.5

The input file TEST1.SIF will be quantized
     to fit a normal density function with
     mean 5 - 2.1 and standard deviation
     .5( 1Ø/6 ) on the interval ( Ø,1Ø )
     using discrete values Ø - 9, and the
     resulting image copied to TEST2.SIF
     .
```

*QVINTP Performs a masking operation to minimize quadratic variation

VERSION: A.Ø2 DATE: 26APR82 CONTACT: D.B. ELLIOTT

ACTION: This command takes an input SIF image and iteratively applies mask operators that minimize the quadratic variation of the image. If the Bflag is set, the user can include a binary image file as the second input file that marks pixels as known points whose values are not to be changed. The user is prompted for the stoping criteria just as in the INTERP command.

SOURCE: DISK
DESTINATION: DISK, (REPORT IS PRINTED TO THE TERMINAL)

FLAGS: (B) : Allows user to specify boundary (known) points whose values are not to change.

QUESTIONS: 1. Enter number of iterations to be performed and the cutoff value for the maximum difference between iterations.
2. Do you want to go again?

COMMAND STRING EXAMPLE:

```
 QVINTP  TEST.SIF , BINARY.SIF > TESTED.NEW (B)
       Enter number of iterations: 2Ø
       Enter maximum difference : 5.

  CONVERGENCE WAS OBTAINED BEFORE ITERATIONS
        COMPLETED

  Iterations requested 2Ø max. diff requested
        5.
  Iterations performed 18 max. diff calculated
        4.9987
  45Ø2 points were calculated with average difference
        = .123456E+Ø1

  Do you want to go again (Y/N) ?
   N
```

The operator masks were applied to all pixels not marked with a 1 in the BINARY.SIF file for 18 iterations. The pixel whose

value changed the most between the 17th and 18th iteration had a difference of 4.9987 between those iterations which is less than the 5 specified by the user, thus the iterations stopped and the report was printed.

*REGRES Does a Least Squares Fit between 2 files
and outputs a table

VERSION: A.Ø2 DATE: Ø7-17-81 CONTACT: D.B. ELLIOTT

ACTION: This command takes a pixel pair (input file 1 , input file 2) for all pixel positions and performs a Least Squares Fit Order 1 (Order 2 if Q-flag is set). Then, a table mapping the greytone values of the first input file to the greytone values of the second input file is output as a sequential file.

SOURCE: Two disk files (Three if the O-flag is set)
DESTINATION: One sequential file

FLAGS: (Q) Makes the Least Squares Fit fit a quadratic curve through the pionts rather than the default order one fit (a straight line).
(M) Does a regression where the domain maps to the mean of the ranges. Therefore, the resulting function may not be linear or quadratic . It may be of any order.
(O) Allows the user to include a third input file and the pixel pair that is marked with a 1 in the third input file is omitted from the Least Squares computation.

QUESTIONS: None

COMMAND STRING EXAMPLE:

REGRES B73A.DSP , SUNLIT.SIF > TABLE.DAT
(Q)

A table which gives a "mapped" value for each integer in the interval (min,max) on input file 1 is produced. The function that produces the "mapped" value is a Least Squares Fit of Order 2 through the pixel pairs by taking each pixel in file 1 with the pixel in the same position in file 2. The taole is formatted as follows:
first record -- an integer indicating number of records to follow
other records-- first file greytone

"mapped" greytone (both integers separated
by a space and can be read with GETI
)

NOTE: The sequential output file may be the
terminal (TT) or printer (PRINT)

*RELAX4 Computes the optimal relaxation factor for SOR iteration

VERSION: a/Ø1 DATE: 2ØAPR82 CONTACT: D.B. ELLIOTT

ACTION: Given the binary file that defines which pixels are "known" and not variables, the program finds the spectral radius (maximum eigenvalue) of the Jacobi iteration matrix that results from applying the digital Laplacian mask to every variable pixel. The optimal relaxation factor is printed out using the spectral radius calculated. The optimal

SOURCE: DISK, binary file
DESTINATION: DISK, TERMINAL, or PRINTER (sequential file).

FLAGS: NONE

QUESTIONS: NONE

COMMAND STRING EXAMPLE:
    RELAX4 TEST.BIN > TT

MAX Eigenvalue is real = .975Ø04
Optimal relaxation factor for SOR iteration
= 1.636Ø8

ALGORITHM: The EISPACK package is used to calculate the eigenvalues.

*RMS Root mean square difference between two images.

VERSION: A.Ø1 DATE: Ø1-15-8Ø AUTHOR: D.JOHNSON

ACTION: This command takes two images and compares them. If the two files are integer, the number of unequal entries is output. On all modes of files the rms error is output. This is computed by subtracting the two images, squaring the difference, summing the result and taking the square root. This command is very useful in finding how similar two images are.

SOURCE: Disk, input file name.
DESTINATION: Disk, output file name.

FLAGS: None.

QUESTIONS: None.

COMMAND STRING EXAMPLE:

RMS TT < IN1.SIF, IN2.SIF

This yeilds:

IMAGES IN1.SIF AND IN2.SIF

BAND PAIR - 1 - ERROR = 4.1Ø85E-Ø2
2 POINTS DIFFER OUT OF 25ØØ

The output from above indicates that the two files are exactly the same with exeption of two points.

*SBIMG Extract a subimage.

VERSION: A.01 DATE: 01-15-80 AUTHOR: K.NEIKIRK

ACTION: This command subdivides a multiband SIF file.

SOURCE: Disk, input file name.
DESTINATION: Disk, output file name.

FLAGS: (B) Flag indicates that user wants the same size image as the input image(i.e. he just wants to copy the input image)
(A) Flag indicates that user subdivision is to be taken from all bands.

QUESTIONS: (1) The first and last rows of the input image that will make up the output image
(2) The first and last columns of the input image that will make up the output image

COMMAND STRING EXAMPLE:

```
SUBIMG CHECK.SIF  < OUT.SIF

ENTER FIRST AND LAST ROWS(1 - 50 ) -- 1 20
ENTER FIRST AND LAST COLUMNS(1 - 50 ) -- 1 20
```

This will cause the output file OUT.SIF
to contain the rows 1-20 and columns 1-20 of
the input file CHECK.SIF which is a 50 by 50 image.

*SHADE Computes greytone values for the lit/shadowed image of SHADOW

VERSION: A.Ø1 DATE: Ø6-25-81 CONTACT: D.B.ELLIOTT

ACTION: Computes the greytone values due to sun azimuth , elevation, sun light intensity, and 3-d orientation of each pixel for the input image produced by the SHADOW command.

SOURCE: Disk, Input file names (3 files described below),
DESTINATION: Disk, Output file name

FLAGS: (R) This flag allows the user to include a third input file to be used as a reference map for the pixels.

QUESTIONS: 1. Corrected Sun azimuth
2. Sun elevation.
3. Diffuse lighting ratio
4. Direct sun light intensity
5. Ground distance multiplier.

COMMAND STRING EXAMPLE:

SHADE INPUT.SUN , INPUT.ANG , INPUT.RFL > OUTPUT.INT (R)

ENTER CORRECTED SUN AZIMUTH: 1Ø7.
ENTER SUN ELEVATION ANGLE : 45.
ENTER DIFFUSION RATIO (DIFFUSSION / SUN INTENSITY) : .1Ø
ENTER SUN INTENSITY : 1ØØ
ENTER GROUND DISTANCE MULTIPLIER: 1.

The INPUT.SUN file (produced by the SHADOW command) is used to show which pixels are lit or shaded , the INPUT.ANG file (produced by the CNSFCT command) is used to get the 3-d orientation of each pixel, the INPUT.RFL file is a file of per- centages (Ø - 1ØØ) representing the percentage of light reflected for each pixel in the image. If the R-flag had been omitted, this file would not appear in the command line and all pixels would have an assumed reflectance of 1ØØ per-cent. OUTPUT.INT is the output

image that would result from the sun shining on the original image (image before SHADOW command was used) with the parameters specified.

ALGORITHM: The first input file is used to determine which pixels are lit or shaded. The shaded pixels are all assigned the diffuse lighting. The lit pixels are assigned greytone values determined by the reflectance of the pixel and its orientation with respect to the sun.

NOTE: (1) The sun azimuth , sun elevation , and ground distance multiplier must agree with those parameters entered for the SHADOW command.

(2) The corrected sun azimuth asked for in SHADE is the Sun azimuth - the correction for top of image in SHADOW.

*SHADOW Marks pixels lit/shaded for a given sun elev. & azimuth angles.

VERSION: A.Ø2 DATE: Ø6-17-81 AUTHOR: D.B.ELLIOTT

ACTION: Uses the input file and parameters supplied by the user to produce an image with pixels either lit or shadowed as it would appear if lit by the sun in the specified position. This command just produces lit (1) or shadowed (Ø). The command SHADE computes intermediate greytones for the lit pixels.

SOURCE: Disk,input file name.
DESTINATION: Disk, output file name

FLAGS: None

QUESTIONS:(1) Orientation of the top of the image (degrees from north):
(2) Azimuth of sun (degrees from north):
(3) Ground distance multiplier:
(4) Elevation of sun: (degrees up from horizontal)

The orientation is used if the top of the input image is not toward the north but the sun azimuths are known relative to north. If the sun azimuth is known relative to the top of the image, or if the image is pointing due north, enter Ø. All azimuths are measured as degrees clockwise from north. The elevation is measured in degrees up from horizontal (counterclockwise). The ground distance multiplier is used to make the units in the row , column plane equal to the units in the vertical direction (greytone heights).

COMMAND STRING EXAMPLE:

```
SHADOW INPUT.SIF > OUTPUT.SUN

  ENTER ORIENTATION OF THE TOP OF THE IMAGE:
     12.
  ENTER AZIMUTH OF THE SUN: 119.
  ENTER GROUND DISTANCE MULTIPLIER: 12.
  ENTER ELEVATION OF SUN : 45.
```

A file OUTPUT.SUN will be produced which
has 1's for lit pixels and Ø's for shaded
pixels.

*SURFAC Creates a surface specified by polynomial or trig. formula

VERSION: a.Ø1 DATE: 13 APR 82 CONTACT: D.B.ELLIOTT or GLENN FOWLER

ACTION: User inputs either coefficients of a polynomial in r and c or the cycle and power parameters for a Sine function. The specified function is used to generate greytone values for the output image.

SOURCE: Terminal
DESTINATION: DISK

FLAGS: T flag indicates that user wants a surface generated by a function of the form Z = SIN(K*(R*R+C*C)**(P/2)), where K is the cycle factor and P is the power factor.

S flag indicates that the SQRT of the values given by the polynomial for each pixel are assigned as greytone values to the output file. (Only used when Tflag is not set).

Z flag indicates that all negative values generated will be replaced by Ø in the output file. (Only used when Tflag is not set: automatically invoked when Sflag is set).

QUESTIONS;

When Tflag is set:

Number of rows in output
Number of columns in output
Cycle factor The cycle factor determines the number of cycles on the output image given a power factor of 1.
Power factor The power factor allows the sine function argument to be raised to a power. A power factor > 1 produces a contracting ring pattern and a power factor < 1 produces an expanding ring pattern.
Maximum greytone value in output If a greytone value is computed larger than this maximum, its value will be replaced by maximum.

When Tflag is not set:

Number of rows in output

```
        Number of columns in output
        Row coordinate of origin
        Column coordinate of origin
        Minimum greytone value possible in output
         If a greytone value is computed smaller
         than this minimum, its value will be
         replaced by minimum.
        Maximum greytone value possible in output
        Scaling factor for row and column distance
         This value is multiplied times the number
         of rows/columns from the origin to give
         the "distance" in the row or column
         direction from the origin.
        Order of the polynomial
        Coefficients of the polynomial The coefficients
         of the polynomaial are prompted for
         one at a time.

COMMAND STRING EXAMPLE #1:
      SURFAC SPHERE.OUT (S)
   Number of rows--100
   Number of cols--100
   Row(origin)-----50
   Col(origin)-----50
   Minimum---------0
   MAximum---------39.
   Scaling factor--1.
   Order of poly.--2
     R**0 C**0 -- 1521. (39**2)
     R**1 C**0 -- 0.
     R**0 C**1 -- 0.
     R**2 C**0 -- -1.
     R**1 C**1 -- 0.
     R**0 C**2 -- -1.

    The output will have the greytone values
         representing a sphere (actually top
         half of a sphere) of radius 39 , with
         center at the pixel 50,50. The total
         output image is 100,100.


COMMAND STRING EXAMPLE #2:
    SURFAC TRIG.OUT (T)
        Number of rows -- 100
        Number of col -- 100
        Cycle factor -- 1.0
        Power factor -- 1.0
        Max greytone -- 512
```

The output pattern is similar to the ripple effect of dropping a pebble into a pond of water. The image contains 2 complete cycles with no contraction or expansion.

*TRIDAG Uses a Tridiagonalization method as a digital Laplacian operator.

VERSION: a.Ø1 DATE: 12APR82 CONTACT: D.B.Elliott

ACTION: This routine produces the same result as INTERP would with a Digital Laplacian operator, except the operations are done using the LINPACK linear system solving routines instead of applying a mask operator to each pixel. The S-flag computes the spectral radius of the iteration matrix for the specific problem being solved (giving the user the rate of convergence of the system). This routine requires that a binary file which points out the "known points" be given (just like the B-flag in INTERP). The Rflag is used to allow a relaxation factor. The questions asked are similar to those in INTERP.

SOURCE: Input file one, initial guess image containing correct values for the "known points" ---- DISK

Input file two, binary file with ls marking pixels in input file one that have "known" values an are not to be changed----DISK

DESTINATION: Output file one, result image---- DISK
Output file two (used if Sflag is set), sequential file.

FLAGS: S-flag Prints the spectral radius for this system on second output file (any sequental file including TT & PRINT)

R-flag Allows the use of a relaxation factor.

QUESTIONS: Relaxation factor-- (only if Rflag is set)
Number of iterations to do--
Maximum cutoff value --

COMMAND STRING EXAMPLE:
TRIDAG TEST.SIF,TEST.BIN > TEST.OUT,TT

```
Max. eigenvalue is real = .9675

     Number of iterations desired-- 20
     Maximum difference cutoff -- 1.0



   Iterations requested: 20
     Max. difference requested: 1.0
   Iterations performed: 15
     Max. difference computed: .998

     4591 points were calculated.
     Average difference = .75834

The system will be run one iteration after
     another until EITHER
 1. The number of iteration you requested
     are done , OR
 2. the Maximum change over all pixels is
     less than the Max. diff you specified
     between 2 iterations.
The user is asked if he wishes to proceed
     or stop.
NOTE: the spectral radius was printed since
     Sflag was set.

ALGORITHM: The problem Ax = b , where A is defined
     by the 3 x 3 digital Laplacian operator
     being applied to every pixel in the
     input image (except those points marked
     known by having a 1 in the corresponding
     position in the binary (second) input
     image ) with Neumann boundary conditions.
     x is the initial guess, input image
     one. b is the zero vector except for
     the affects of the "known points". The
     A, x, and b matrices/vectors are constructed
     in the program from the input images.
     Then the A matrix is re-written as A=
     N-P, where N is the Tridiagonal part
     of A and -P is the rest of A (zeros
     in place of the tridiagonal elements).
     The problem now is written: Nx=Px
     b, giving an iteration equation: x(N+1)
     = N-1Px(N) + N-1b, N-1 is N inverse,
     and x(I) is the image at iteration I.
     NOTE that A only involves the points
     that are not "known", the "known" values
     are represented in the initial image
```

x(Ø) and the b vector. The spectral radius of N-1P gives an estimation of the convergence properties of the system.

COMMENTS: This routine requires a lot of storage, especially for the S-flag.

*WARP Image warping

VERSION: A.Ø1 DATE: 11-29-81 CONTACT: O A ZUNIGA

ACTION: This command warps an SIF image using
a polynomial transformation of the row
and column indexes.

SOURCE: Disk, input file name
DESTINATION: Disk, output file name

FLAGS: (I) If set, then an output pixel graytone
is obtained by a bilinear interpolation
of the 4 nearest input pixels, otherwise
the ouput pixel is assigned the graytone
of the nearest input pixel.
(S) If set, then the user is asked for
the size of the output image, otherwise
the output image will be the same size
as the input image.

QUESTIONS: (1) Number of rows for output image
(If S flag set).
(2) Number of columns for output image
(if S flag set).
(3) Number of rows of output block
(4) Number of columns of output block
(5) The dimension of the row transformation
(degree of the row polynomial)
(6) The dimension of the column transformation
(degree of the column polynomial).
(7) The row polynomial coefficients
(8) The column polynomial coefficients.

COMMAND STRING EXAMPLE:

```
    WARP CHECK1.WRP < CHECK1.SIF

 ENTER OUTPUT BLOCK NUMBER OF ROWS ( D = 16,
        1 - 2Ø ) -- 1
 ENTER OUTPUT BLOCK NUMBER OF COLUMNS ( D = 16,
        1 - 2Ø ) -- 2Ø
 ENTER ROW TRANSFORM DIMENSION ( D = 1, 1 - 1Ø
        ) -- 1
 ENTER COL TRANSFORM DIMENSION ( D = 1, 1 - 1Ø
        ) -- 1
 ENTER ROW TRANSFORM POLYNOMIAL
 ENTER ROW**Ø*COL**Ø TERM ( D = Ø.ØØ ) -- 21.ØØ
 ENTER ROW**1*COL**Ø TERM ( D = Ø.ØØ ) -- -1.ØØ
 ENTER ROW**Ø*COL**1 TERM ( D = Ø.ØØ ) -- Ø.ØØ
```

```
ENTER COLUMN TRANSFORM POLYNOMIAL
ENTER ROW**Ø*COL**Ø TERM ( D = Ø.ØØ ) -- 21.ØØ
ENTER ROW**1*COL**Ø TERM ( D = Ø.ØØ ) -- Ø.ØØ
ENTER ROW**Ø*COL**1 TERM ( D = Ø.ØØ ) -- -1.ØØ
```

The command will print the following message:

```
INPUT BLOCK ACTUAL NUMBER OF ROWS = 1
INPUT BLOCK ACTUAL NUMBER OF COLS = 2Ø
INPUT BLOCK REQUIRED NUMBER OF ROWS >= 1
INPUT BLOCK REQUIRED NUMBER OF COLS >= 2Ø
```

A 18Ø degrees rotation is done on the 2Ø X 2Ø image CHECK1.SIF which is in line format. The command prints out the actual block size of the input image and the required block size. If the block size is appropriate the command proceeds, otherwise it is necessary to re-block the input image.

ALGORITHM: Let (r,c) be the row and column index of a pixel in the output image and let (r',c') be the row and column index of the corresponding transformed pixel in the input image. Let M and N be the degree of the row and column transformation polynomials. Then (r,c) and (r',c') are related by:

$$r' = f(r,c)$$
$$c' = g(r,c)$$

where f and g are polynomials of degree M and N respec- tively entered by the user. Storage of the M previously computed row indexes and the N previously computed column indexes allow the computation of the current row and column indexes to be done using M and N operations respectively. Memory management is done in such a way that only 4 adja- cent input blocks are necessary to generate one output block.

Appendix B

RUNFILE SHOWING HOW COMMANDS ARE USED

```
$!
$!    ***** PREPROCESSING *****
$!
$!  READ IN THE LANDSAT DATA FROM TAPE
$!
$!   CONVERT TO SIF FORMAT
$!
$ BINSIF LANDSAT.TAP > B7.SIF
 512 512
 Ø
 Ø
 8
$!
$!
$!  SUB IMAGE TO GET AREA DESIRED
$!
$SBIMG B7.SIF > B73.SIF
 238 474
 1   237
$!
$!   CORRECT FOR SKEWING
$!
$ LNTRA B73.SIF > B73.TRA
 Y
 -.Ø573377
 1
 1
$!
$!  SUB IMAGE AGAIN
$!
$SBIMG B73.TRA > B73A.TRA
 61 14Ø
 26 1Ø5
$!
$! DE-STRIPE
$!
$DSTRPE B73A.TRA > B73A.DSP
$!
$!  WARP TO GET PIXELS THAT REPRESENT SQUARE AREAS
$!
$WARP B73A.DSP > B73A.WRP
 1
 8Ø
 1
```

```
 Ø. .7Ø8861 Ø.
 Ø. Ø. 1.
$!
$!    **** FIND RIDGES AND VALLEYS ****
$!
$!
$!     THRESHOLD THE IMAGE TO GET THE BINARY IMAGE
          OF DARK PIXELS
$!
$SLICE B73A.WRP > B73A.B
 Ø 12
$!
$CONCT B73A.B > B73AB.CON
$!
$!     ENLARGE THE DARK REGIONS BY INCLUDING SURROUNDING
          NOISY PIXELS
$!
$GFILL B73A.WRP,B73AB.CON > B.FIL (D)
 13 13
$!
$PRPMBI B.FIL,B73A.WRP > B.PRP(L)
 1 1ØØ
$!
$ELIMRG B.FIL,B.PRP > B.ELI
 1
 6
$!
$DELETE B.FIL
$DELETE B73A.B
$DELETE B73AB.CON
$DELETE B.PRP
$!
$!     THRESHOLD THE IMAGE TO GET THE BINARY IMAGE
          OF VERY BRIGHT PIXELS
$!
$SLICE B73A.WRP > B73A.W
 24 5Ø
$!
$CONCT B73A.W > B73AW.CON
$!
$PRPMBI B73AW.CON,B73A.WRP > W.PRP(L)
 1 1ØØ
$!
$ELIMRG B73AW.CON,W.PRP > W.ELI
 1
 3Ø
$!
$DELETE B73A.W
$DELETE B73AW.CON
$DELETE W.PRP
```

```
$!
$!      CREATE A SYMBOLIC IMAGE OF ALL ONES
$!
$SLICE B73A.WRP > B73A.Z1
 Ø 5Ø
$!
$!      CREATE BORDER SEGMENTS P-FILE AND CHAIN CODE
          FILE
$!
$BORCHN BGW.REL > BGW.LP1, BGW.PX1 (G)
$!
$!      CREATE THE SYMBOLIC IMAGE OF BORDER SEGMENT
          FILE
$!
$CHNSIF BGW.LP1, BGW.PX1 > BGW.BO (A)
$!
$!      CALCULATE ORIENTATION INFOMATION AT 2 ENDS
          OF BOEDER SEGMENTS
$!
$BORORI BGW.LP1, BGW.PX1 > BGW.LP2 (A)
$!
$!      CREATE JUNCTION P-FILE
$!
$BORJCT BGW.REL, BGW.BO, BGW,LP2 > BGW.JP1, BGW.LP3
          (A)
$!
$PRPMBI BGW.REL, BGW.WRP > BGW.PRP
$!
$!      PUT REGION BRIGHTNESS INFOMATION INTO P-FILE
$!
$LBREG BGW.LP3, BGW.PRP > BGW.LP4
 13 24
$!
$!      GET RIDGE-VALLEY INFORMATION
$!
$RDGVL BGW.JP1, BGW.LP4 > BGW.JP2, BGW.LP5 (A)
$!
$!      CREATE 2-BANDS IMAGE CONTAINING RIDGE-VALLEY
          LOCATIONS
$!
$CHNSIF BGW.LP5, BGW.PX1 > BGW.VR (PA)
 22 2 1 2
$!
$!      CREATE BORDER SEGMENTS P-FILE AND CHAIN CODE
          FILE
$!
$BORCHN BGW.REL > BGW.LP1, BGW.PX1 (G)
$!
$!      CREATE THE SYMBOLIC IMAGE OF BORDER SEGMENT
          FILE
```

```
$!
$CHNSIF BGW.LP1, BGW.PX1 > BGW.BO (A)
$!
$!      CALCULATE ORIENTATION INFOMATION AT 2 ENDS
           OF BOEDER SEGMENTS
$!
$BORORI BGW.LP1, BGW.PX1 > BGW.LP2 (A)
$!
$!      CREATE JUNCTION P-FILE
$!
$BORJCT BGW.REL, BGW.BO, BGW,LP2 > BGW.JP1, BGW.LP3
          (A)
$!
$PRPMBI BGW.REL, BGW.WRP > BGW.PRP
$!
$!      PUT REGION BRIGHTNESS INFOMATION INTO P-FILE
$!
$LBREG BGW.LP3, BGW.PRP > BGW.LP4
 13 24
$!
$!      GET RIDGE-VALLEY INFORMATION
$!
$RDGVL BGW.JP1, BGW.LP4 > BGW.JP2, BGW.LP5 (A)
$!
$!      CREATE 2-BANDS IMAGE CONTAINING RIDGE-VALLEY
           LOCATIONS
$!
$CHNSIF BGW.LP5, BGW.PX1 > BGW.VR (PA)
 22 2 1 2
$!
$!    SPLIT RIDGE-VALLEY FILE INTO INPUTS FOR ELEVATION
           MODEL
$!
$!
$!   THIS RUN FILE SPLITS A 2-BAND RIDGE-VALLEY
           FILE
$!   BAND 1 - VALLEY , BAND 2 - RIDGE , INTO 2
           IMAGE FILES
$! NAMED WRPBGW.RV AND WRPBGW.RG. THE INPUT IS
           FILE IS CALLED WRPBGW.VR.
$! THE TWO BANDS ARE NOT BINARY, BUT SYMBOLIC
           AND EVERY LINE
$! SEGMENT IS REPRESENTED AS A DIFFERENT REGION.
$!
$! MAKE AN IMAGE OF ALL 1s TO BE USED LATER.
$!
$ MKCHK ONE.SIF
 113 80
 113 80
 1 100
```

```
 Ø.
$!
$! OPEN THE ONE.SIF FILE AND CHANGE IT FROM NUMERIC
         TO A SYMBOLIC IMAGE
$!
$EXSIF
 OPEN ONE.SIF
 MID 18 1
 Y
DONE
$!
$! 'AND' THE BAND 1 OF THE 2-BAND IMAGE TO THE
         ONE.SIF TO GET VALLEY BINARY
$!  IMAGE
$!
$BOOL WRPBGW.VR , ONE.SIF > WRPBGW.RV
 1
 1
 1
$!
$! 'AND' THE BAND 2 OF THE 2-BAND IMAGE TO ONE.SIF
         TO GET THE RIDGE BINARY
$!  IMAGE.
$!
$BOOL WRPBGW.VR , ONE.SIF > WRPBGW.RG
 1
 2
 1
$DELETE ONE.SIF
$!
$!
$!   **** CREATE ELEVATION MODEL ****
$!
$!
$!   THIS RUNFILE CREATES A SMOOTHED ELEVATION
         MODEL IMAGE.
$!   THE REQUIRED INPUTS ARE: WRPBGW.RV , WRPBGW.RG.
         THE RESULT
$!   IMAGES ARE:Ø50100.SIF, RDGVAL.BIN, ELEVQV.OUT,
         AND ELEVQV.SIF
$!   THE INPUT FILES ARE SINGLE BAND IMAGES.
$!
$BOOL NEWRNV.TMP < WRPBGW.RV,WRPBGW.RG (X)
 9
$EXSIF
 OPEN NEWRNV.TMP
 MID 18 Ø
 Y
 DONE
$ARITHM Ø50100.SIF < NEWRNV.TMP
```

```
 A
 100
 N
$EXSIF
 OPEN 050100.SIF
 TOP
PROT OFF
 REPL 100 50
 TOP
 REPL 101 0
 TOP
 REPL 102 100
 DONE
$BOOL RDGVAL.BIN < WRPBGW.RV, WRPBGW.RG
 5
$QVINTP ELEVQV.OUT < 050100.SIF, RDGVAL.BIN (B)
 100
 1.5
 N
$!
$! TRANSLATE SO ALL VALUES >= 0
$!
$ARITHM ELEVQV.OUT > ELEVQV.SIF (A)
 28
 N
$!
$DELETE NEWRNV.TMP
$!
$!
$!
$!    **** CREATE ILLUMINATION MODEL IMAGE ****
$!
$!
$ SHADOW ELEVQV.SIF > QVSUN1.BIN
 12.
 119.
 30.621118
 45.
$!
$!   COMPUTE 3-D ORIENTATION OF PIXELS
$!
$ CNSFCT ELEVQV.SIF > ELEVQV.SFT
 3 3
$!
$!   CREATE LIGHT INTENSITY IMAGE FOR PRIMARY
       SOURCE
$!
$ SHADE QVSUN1.BIN , ELEVQV.SFT > QVSHD1.SIF
 107.
 45.
```

```
 .14
 500
 30.621118
$!
$!   CREATE SHADOW IMAGE FOR BACKLIGHTING
$!
$ SHADOW ELEVQV.SIF > QVSUN2.BIN
 12.
 199.
 30.621118
 45.
$!
$!   COMPUTE LIGHT INTENSITIES FOR BACKLIGHTING
$!
$ SHADE QVSUN2.BIN , ELEVQV.SFT > QVSHD2.SIF
 187.
 45.
 .14
 100
 30.621118
$!
$!   ADD PRIMARY AND BACKLIGHTING TOGETHER
$!
$ ARITHM QVSHD1.SIF , QVSHD2.SIF > QVILLUM1.SIF
          (A)
 N
$!
$!   **** DO REGRESSION FOR COMPARISON ****
$!
$ REGRES QVILLUM1.SIF , B73A.WRP > QVWRP1.REG
          (M)
$!
$ PQUANT QVILLUM1.SIF, QVWRP1.REG > QVILLUM1.REG
          (R)
$!
$!    COMPUTE DIFFERENCE BETWEEN LANDSAT AND ILLUMINATION
         MODEL
$!
$ ARITHM B73A.WRP , QVILLUM1.REG > WRPILL1.DIF
          (S)
 N
$!
$!   PRINT HISTOGRAM ON TERMINAL
$!
$MHIST TT < WRPILL1.DIF (M)
$!
$!   COMPUTE ABSOLUTE VALUE OF THE DIFFERENCE
          IMAGE FOR FIGURE
$!
$ABSIMG WRPILL1.DIF > WRPILL1.ABS
```

```
$!
$!   DO MORE EXPERIMENTS
$!   TEST REGRESSION WITHOUT HIGH AND LOW
$!
$!
$!   DO REGRESSION , BUT OMIT HIGH AND LOW VALUES
          FROM LANDSAT
$!
$!   THRESHOLD LANDSAT IMAGE
$!
$ TRSLD B73A.WRP > B73ALW.BIN
 8
$!
$ TRSLD B73A.WRP > B73AHI.BIN (B)
 25
$!
$ BOOL B73AHI.BIN , B73ALW.BIN > B73AHILW.BIN
 5
$!
$ REGRES QVILLUM1.SIF , B73A.WRP , B73AHILW.BIN
          > QVWRP1.RG2 (MO)
$!
$ PQUANT QVILLUM1.SIF, QVWRP1.RG2 > QVILLUM1.RG2
          (R)
$!
$!   HISTOGRAM THE NON-ERROR PIXELS
$!
$ MKCHK 1000.SIF
 113 80
 113 80
 1000
 1001
 0
$!
$ ARITHM B73A.WRP,QVILLUM1.RG2 > WRPILL1.DF2 (S)
 N
$!
$!
$ CPYRPL WRPILL1.DIF, 1000.SIF, B73AHILW.BIN >
          WRPILL1.HT1 (FC)
 Y 5 11
 N
$!
$ CPYRPL WRPILL1.DF2, 1000.SIF, B73AHILW.BIN >
          WRPILL1.HT2 (FC)
 Y 5 11
 N
$!
$MHIST TT < WRPILL1.HT1 (E)
 20
```

```
$!
$MHIST TT < WRPILL1.HT2 (E)
 20
$!
$DELETE 1000.SIF
$!
$!   **** DO SECOND ILLUMINATION MODEL
$!        USING REFLECTANCE ****
$!
$!
$!
$!    THRESHOLD THE DIFFERENCE IMAGE AND MAKE
        REFLECTANCE MAP
$!
$ TRSLD WRPILL1.DIF > ILL1LW.BIN
 -8
$!
$ TRSLD WRPILL1.DIF > ILL1HI.BIN (B)
  9
$!
$ BOOL NEWRNV.TMP < ILL1LW.BIN,ILL1HI.BIN (X)
 9
$!
$ EXSIF
 OPEN NEWRNV.TMP
 MID 18 0
 Y
 DONE
$!
$ARITHM REFLCT1.RFT < NEWRNV.TMP
 A
 100
 N
$!
$EXSIF
 OPEN REFLCT1.RFT
 TOP
PROT OFF
 REPL 100 80
 TOP
 REPL 101 60
 TOP
 REPL 102 100
 DONE
$!
$DELETE NEWRNV.TMP
$!
$!   CREATE LIGHT INTENSITY IMAGE FOR PRIMARY
        SOURCE
$!
```

```
$ SHADE QVSUN1.BIN , ELEVQV.SFT , REFLCT1.RFT
          > QVSHD12.SIF (R)
 107.
 45.
 .14
 500
 30.621118
$!
$!   COMPUTE LIGHT INTENSITIES FOR BACKLIGHTING
$!
$ SHADE QVSUN2.BIN , ELEVQV.SFT , REFLCT1.RFT
          > QVSHD22.SIF (R)
 187.
 45.
 .14
 100
 30.621118
$!
$!   ADD PRIMARY AND BACKLIGHTING TOGETHER
$!
$ ARITHM QVSHD12.SIF , QVSHD22.SIF > QVILLUM2.SIF
          (A)
 N
$!
$!   DO REGRESSION FOR COMPARISON
$!
$ REGRES QVILLUM2.SIF , B73A.WRP > QVWRP2.REG
          (M)
$ PQUANT QVILLUM2.SIF, QVWRP2.REG > QVILLUM2.REG
          (R)
$!
$!    COMPUTE DIFFERENCE BETWEEN LANDSAT AND ILLUMINATION
          MODEL
$!
$ ARITHM B73A.WRP , QVILLUM2.REG > WRPILL2.DIF
          (S)
 N
$!
$!   PRINT HISTOGRAM TO TERMINAL
$!
$MHIST TT < WRPILL2.DIF (M)
$!
$!   COMPUTE ABSOLUTE VALUE OF DIFFERENCE IMAGE.
$!
$ ABSIMG WRPILL2.DIF > WRPILL2.ABS
$!
$EXIT
```

The vita has been removed from the scanned document

THE APPLICATION OF AN ILLUMINATION MODEL TO A MOUNTAINOUS LANDSAT SCENE

David B. Elliott

(ABSTRACT)

The multispectral scanners (MSS) of the LANDSAT satellites collect solar radiation of different wavelengths reflected from the earth's surface. While the different greytone values of a given band of a LANDSAT image of simple terrain are due almost entirely to the various reflectivity values of the features of the earth's surface (such as vegetation, mineral deposits, or bodies of water), the same is not true of areas of complex topography. In mountainous areas, the mixture of light and dark regions in a LANDSAT image may be due to shadow effects as well as the reflectivity values for those wavelengths of light of the various surface features.

In this research, an illumination model is developed to help understand features observed in a LANDSAT scene of a mountainous area. The illumination model is defined and its implementation in the GIPSY (General Image Processing SYstem) system is discussed. The application of the model to a particular LANDSAT scene is described including the development of an elevation model from the LANDSAT data. Finally, the illumination model image is compared with the LANDSAT scene and the the results are discussed.