SPATIAL REASONING ABOUT REMOTELY SENSED DATA FOR DRAINAGE NETWORK MAPPING

by

Shyuan Wang

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Computer Science and Application

APPROVED:

R.M. Haralick, Co-chairman

R.W. Ehrich, Co-chairman

J.B. Campbell, Jr.

F.G. Gray

L.G. Shapiro

August, 1983
Blacksburg, Virginia

SPATIAL REASONING ABOUT REMOTELY SENSED DATA FOR DRAINAGE NETWORK MAPPING

by

Shyuan Wang

(ABSTRACT)

In order to extract a drainage network from a LANDSAT scene of mountainous terrain, it is necessary to use an illumination model to separate the reflectance information from the topographic information in the LANDSAT data. From the reflectance information, visible stream segments can be detected. From the topographic information, ridges and valleys can be located and assigned relative elevations by an elevation growing model. Based on these, a complete elevation model can be estimated by interpolating between ridges and valleys and this estimate can be improved by making it consistent with the topographic information. Also the LANDSAT imagery can be reconstructed to evaluate the illumination model.

In order to label flow directions of visible stream segments, constraints at junctions based on orientations and lengths are defined and a consistent labeling technique is used.

ACKNOWLEDGEMENTS

The author wants to thank Dr. Haralick, Dr. Ehrich, Dr. Campbell, and Dr. Shapiro for their helpful discussions and suggestions for this dissertation. The help in programming from and Dr. Shapiro and the software/hardware support from Spatial Data Analysis Lab are greatly appreciated. The author wants also to thank NASA for helping to support the initial part of this research.

Table of Contents

1. Introduction

   1.1 The Problem

   1.2 Summary of Solution

   1.3 Previous Work

   1.4 Study Area

2. Illumination Model

   2.1 The Problem of Confounded Information

   2.2 The Model for LANDSAT Imagery

   2.3 Haze Correction

   2.4 Approach to the Problem of Confounded Information

   2.5 Ratio Images and Material Clusters

   2.6 Shadow Image

   2.7 Diffuse Light Image, Reflectance Image, and Topographic Modulation Image

   2.8 Clustering

3. Detection of Visible Rivers

4. Elevation Reconstruction

   4.1 Ridge-Valley Map

      4.1.1 Identification of Peak Junctions

   4.2 Relative Elevations of Ridges and Valleys

4.3 Interpolation between Ridges and Valleys

4.4 Use of Topographic Modulation Image to Improve the Result

4.5 Evaluation of Elevation Model and Assumption 2

4.6 Reconstruction of the LANDSAT Imagery

5. Flow Direction of Visible Rivers

5.1 Constraints at Junctions

5.2 Consistent Labeling

5.3 Spatial Reasoning Model and Flow Directions of Visible Rivers

6. Conclusion

. Bibliography

. Appendix

A. Computing the orientation of a border segment

B. Forward Checking Algorithm

C. GIPSY Runfiles

. Vita

Chapter I

INTRODUCTION

1.1 THE PROBLEM

Remote sensing practitioners are highly skilled in deriving information from complicated geographical structures represented on remotely sensed imagery despite the fact that they may have great difficulty explaining their reasoning processes. Because humans have great innate facilities for extracting information from visual shapes, forms, and textures, photointerpreters often do not devote much conscious thought to their analyses of the detailed relationships between the light and dark patterns that convey information about the content of a visual scene. Their reasoning process may be implemented through a series of implicit steps which are not immediately obvious. As a result, to mechanize this process it is necessary to devote some effort to constructing explicit definitions of many of the subtle (possibly vague) processes applied by a skilled photointerpreter as he analyses a scene.

The research problem addressed by this dissertation is that of deducing the drainage network and the flow directions of rivers in Landsat imagery. At an early stage it was realized that one cannot process LANDSAT imagery directly to deduce drainage network because different kinds of information are mixed together in this kind of imagery. It is necessary to first define an illumination model and extract each kind of information from the original LANDSAT imagery. In order to evaluate the illumination model, it is necessay to create an elevation model which assigns elevation values to all pixels. Thus three main problems are addressed in this dissertation: definition of an illumination model, understanding the elevation model, and deduction of drainage networks.

There are two reasons for the choice of this application topic. First, data are abundant, and they are available for familiar geographic regions. Second, despite the apparent triviality of the problem, the reasoning required to reconstruct the drainage network is in fact very complicated, requiring simultaneous analysis both at the detailed local level and at very general global levels. Furthermore, reasoning about drainage systems requires a great deal of knowledge of geography and physics that is not directly part of the spatial reasoning model. This knowledge, for example, includes such facts as "water runs downhill" and "in the morning, the sun is in the east".

1.2 SUMMARY OF SOLUTION

This dissertation describes a procedure by which the stream network and a relative elevation model can be inferred from a LANDSAT scene of mountainous or hilly terrain. The process has a number of distinct steps. First to appropriately prepare the imagery for processing it is necessary to destripe it and perform haze removal. Destriping can be done by the Horn and Woodham [1979] technique. Haze removal can be done by the Switzer, Kowalik and Lyon [1981] technique.

To a first order effect, after preprocessing, the intensity value at directly illuminated pixels is due to whether it is only diffusely or directly illuminated and if it is directly illuminated there are additional effects due to the angle at which the sun illuminates the ground patch corresponding to the pixel, the reflectance of the surface material on the ground patch, and diffuse light. For shadowed pixels, the cause of the intensity value is only due to diffuse light. To make sense of the spatial pattern first requires separation of these individual effects. For this purpose the model based on the Eliason, Soderblom and Chavez [1981] technique can be used to create four images from the one LANDSAT image. They are a shadow image, diffuse light image, reflectance image, and topographic modulation image which has information related to surface slope and sun illumination. Details are given in Chapter 2.

Chapter 3 discusses the identification of all the areas of water from the reflectance image using the Alfoldi and Munday [1978] procedure. Chapter 4 discusses how the shadow image can be used to identify the ridges and the valleys. Because the information of local slope and aspect angle are mixed in the topographic modulation image, it cannot be used directly to reconstruct elevations. A process called elevation growing is used initially to reconstruct relative elevation. This process first assigns local slopes to each pixel by the distance and orientation of the pixel to its nearest ridge and valley segments, and then one can integrate the assigned slopes either along valleys or across the hillsides from valley segments to ridge segments. Once ridges and valleys have been located and assigned relative elevations, a complete elevation model can be generated by interpolating values for non-ridge and non-valley pixels. After this, an iterative Bayesian technique is used to improve the reconstructed elevation model based on the topographic modulation image and the previous estimate of the elevation model.

The final element of the spatial reasoning is the assignment of stream or river flow direction for those water bodies which were directly identified by reflectance properties. The consistent labeling technique is used. This is discussed in Chapter 5.

1.3 PREVIOUS WORK

Since the launch of the first Earth Resources and Technology Satellite (ERTS, later renamed LANDSAT) in July 1972, much work in remote sensing has been done by using pattern analysis and picture processing techniques for image classification, restoration and enhancement. Few people have applied the scene analysis or artificial intelligence approach to describe the image in terms of the properties of objects or regions in the image and the relationships between them. Ehrich [1977] found global lineaments by partitioning the image into windows and applying long, straight, linear filters at different orientations in each window to extract local evidence. Dynamic programming [Montanari, 1971; Martelli, 1972] was then used to identify complete global lineaments. VanderBrug [1976] tested various detectors to obtain linear features in satellite imagery. This was only at the local level. Later VanderBrug [1977a] used relaxation to reduce noise in the output. Finally VanderBrug [1977b] defined a merit function that can be used to select pairs of segments to be merged so that local line detector responses can be linked together into a global representation of the curves. His work is closely related to the Shirai [1973] technique which employed sequential line following to find edges in scenes containing polyhedra. Li and Fu

[1976] used tree grammars to locate highways and rivers from LANDSAT images. The above investigations deal with the extraction of all the linear features from an image, but they do not deal with the interpretation of these linear features. In the following investigations, knowledge about the desired features are considered crucial.

Bajcsy and Tavakoli [1975] argued that an image filter is not meaningful unless one has a world model, a description of the world one is dealing with. They recognized objects matching this description and filtered them out. This strategy is used in sequence for the recognition of bridges, rivers, lakes, and islands from satellite pictures. Nagao and Matsuyama [1980] built an image understanding system that automatically located a variety of objects in an aerial photograph by using diverse knowledge of the world. It is one of the first image understanding systems that incorporated very sophiscated artificial intelligence techniques into the analysis of complex aerial photographs. Fischler, Tenenbaum and Wolf [1981] designed a low-resolution road tracking (LRRT) algorithm for aerial imagery. The approach was based on a new paradigm for combining local information from multiple sources, map knowledge, and generic knowledge about roads. The final interpretation of the scene was achieved by using either graph search or dynamic programming. Similarly, knowledge is important in this problem which requires analysis both at the local and global levels.

1.4 STUDY AREA

This research examines an area in southeastern West Virginia, shown in Figure 1.1. This region is a portion of the Appalachian Plateau's physiographic province, within the "unglaciated Allegheny plateau" described by Thornbury [1967]. In general, this region is a thoroughly dissected plateau-like surface. It receives 40 inches of precipitation each year and, as depicted on topographic maps, has a moderate drainage density. Drainage is through tributaries of the New (Kanawha) River, which flows west into the Ohio drainage system.

The overall drainage pattern within this region is that of a relatively large sinuous channel (the Gauley River) superimposed over the finer texture of a dentritic pattern formed by first, second, and third order streams. A number of the small first or second order streams flow directly into the large channel. Thus the overall pattern is composed of a mixture of many very small stream segments, many with very steep gradients, a prominent major channel with a relatively low gradient, and relatively few stream segments of intermediate length and gradient.

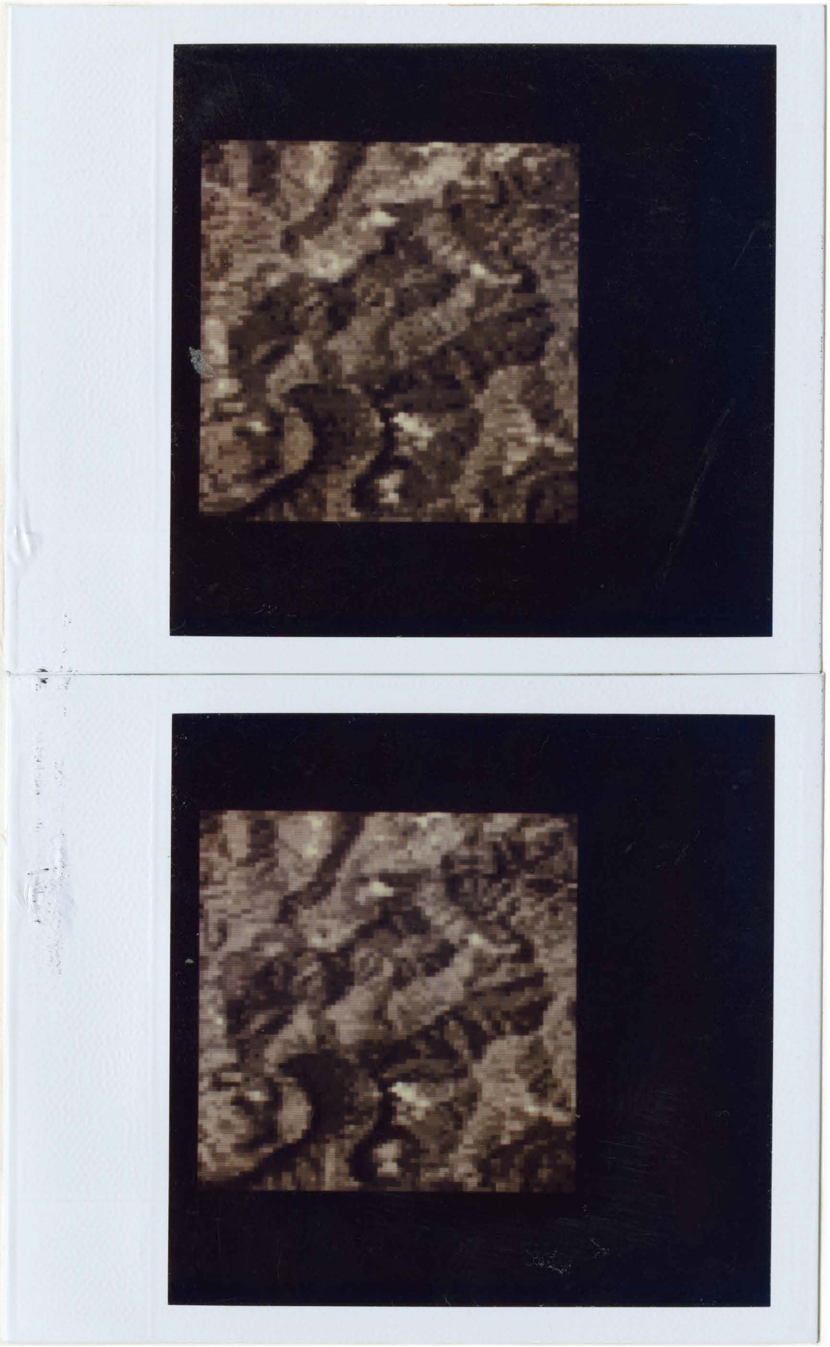


Figure 1.1 - Original LANDSAT scene.

(Top: band 4; bottom: band 5)

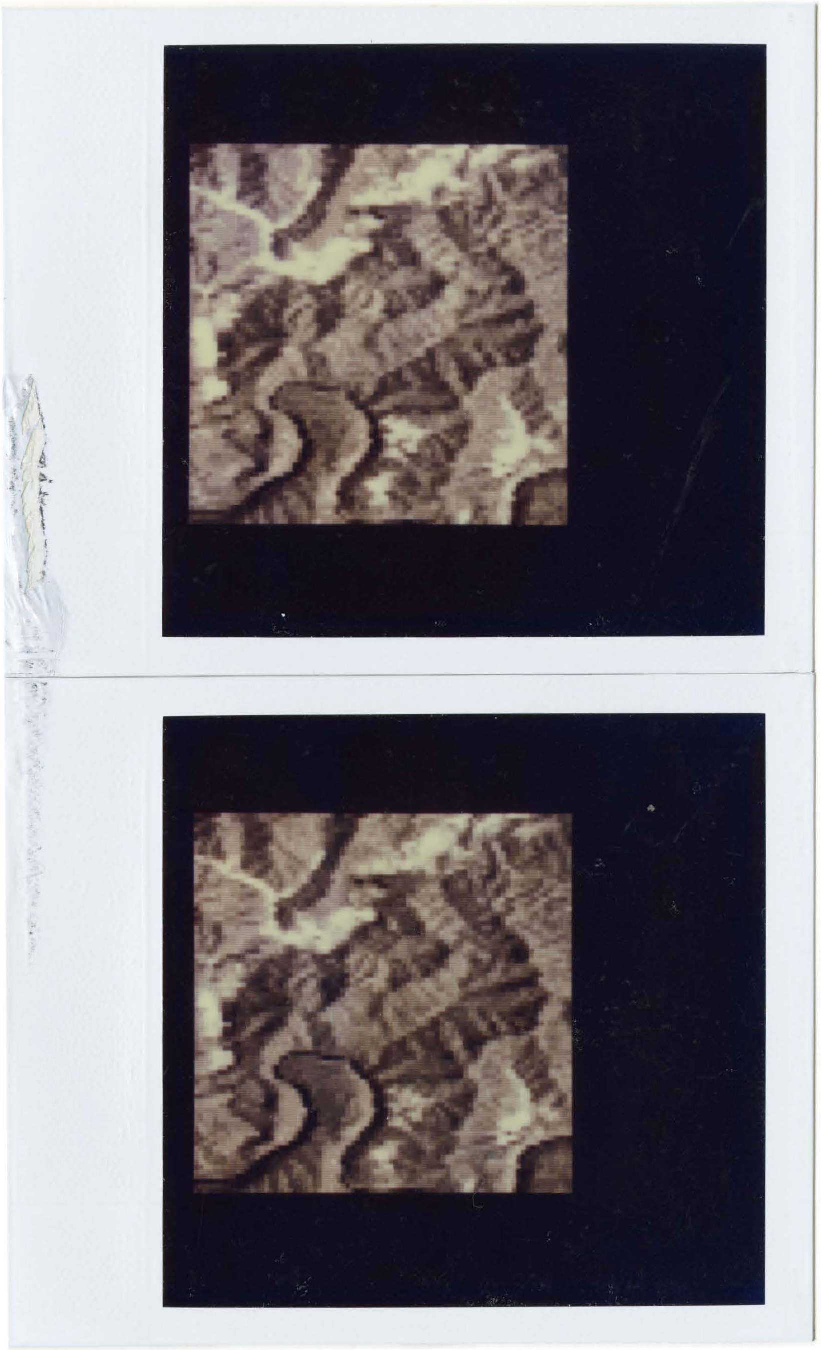


Figure 1.1 - Continued. (Top: band 6; bottom: band 7)

Throughout the area, flood plains (when present) are narrow and tend to closely follow the course of the stream channel. Valleys are narrow, with steep sides; the Gauley River, for example, follows a valley that is typically 150 meters deep but only 100 meters wide. Uplands often consist only of ridge crests; although plateau-like upland regions are present, they are not continuous or extensive. The area is forested with a dense cover of deciduous trees (Kuchler's " mixed mesophytic forest") [1964]. Cleared areas for agriculture (chiefly pasture) tend to follow the valleys of intermediate-sized streams. Settlements are small and dispersed, usually positioned in valleys.

This region appears on the Charleston, West Virginia/ Ohio USGS 1:250,000 quadrangle (NJ 17-5). Our investigations include areas in Nicholas County, W. VA and neighboring counties. This area was imaged by the LANDSAT-1 MSS on April 13, 1976 (scene id: 5360-14502; path 18, row 34). This date reflects important qualities of the scene. First, at this date the atmosphere was unusually clear--there is no evidence of atmospheric scattering or degradation of the data. Also, at this spring date most of the forested areas are without leaves, especially at higher elevations. Lower elevations have a cover of newly-emerged leaves and grasses. Within a few weeks leaves will have emerged in vegetation throughout the entire region, but at this time in April, there is a sharp spectral contrast between the vegetation cover of the higher elevations and that of some of the valleys.

Two other test imagery in this area are shown in Figure 1.2. In the following, only the imagery of Figure 1.1 will be used for illustration, but the final evaluation data will be calculated for all areas.

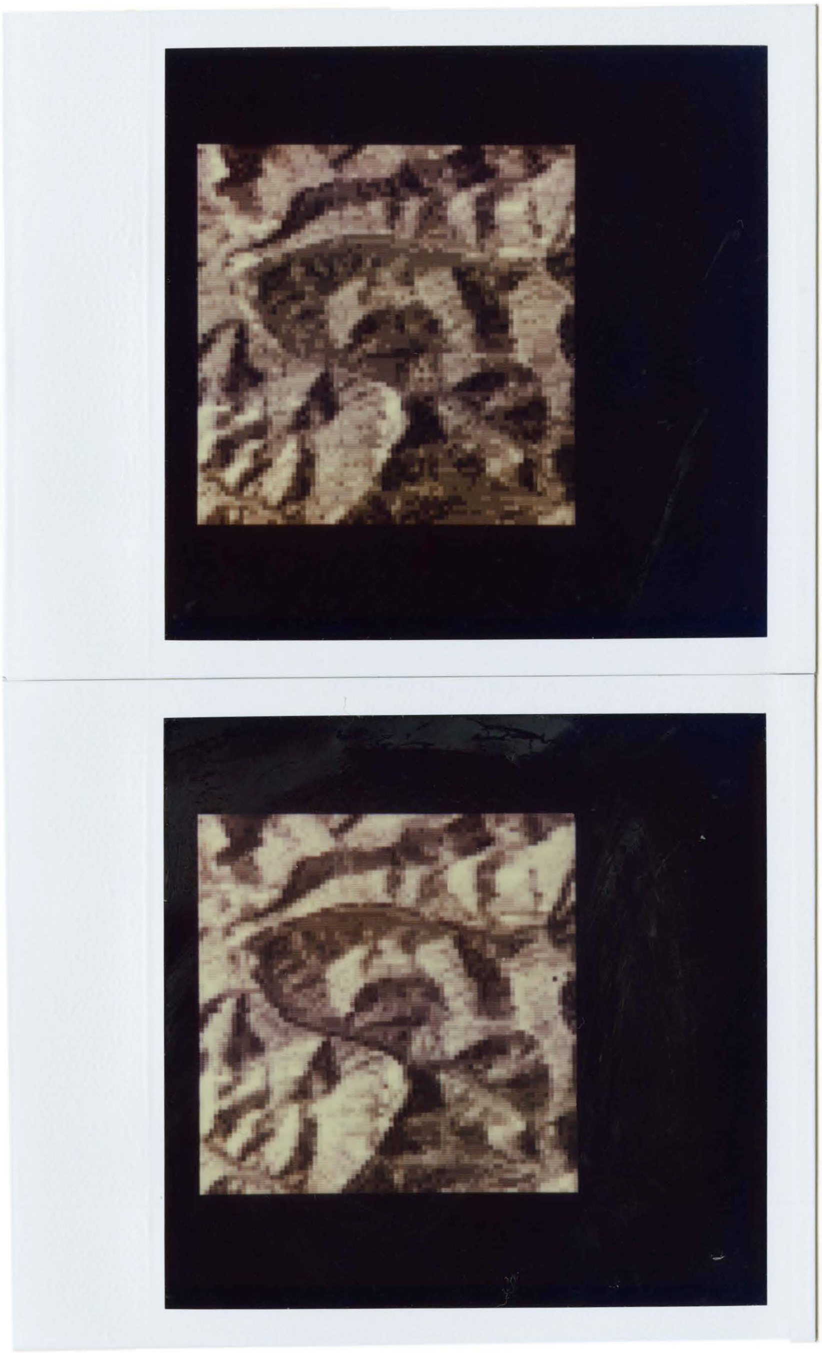


Figure 1.2.a - Test area 2.

(Top: band 4; bottom: band 5)

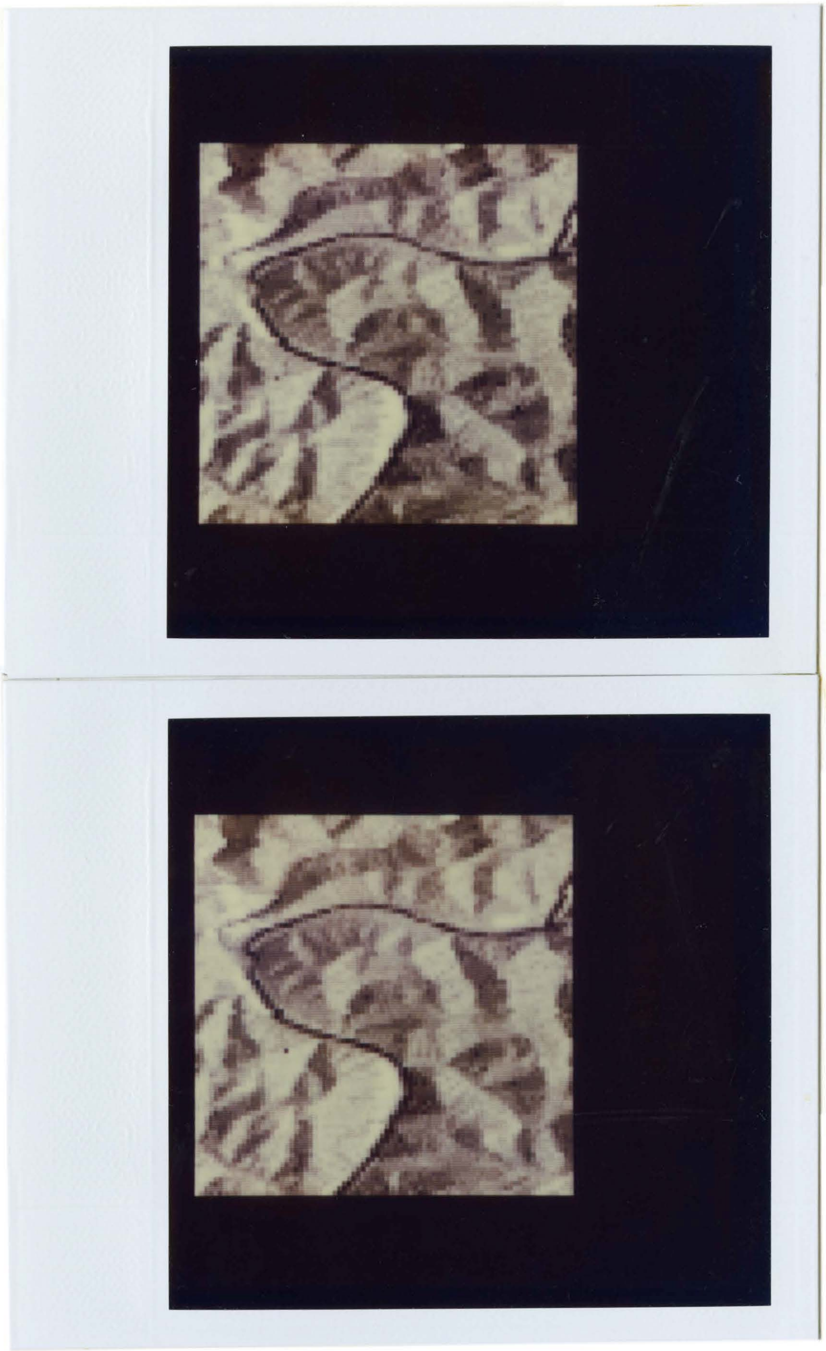


Figure 1.2.a - Continued. (Top: band 6; bottom: band 7)

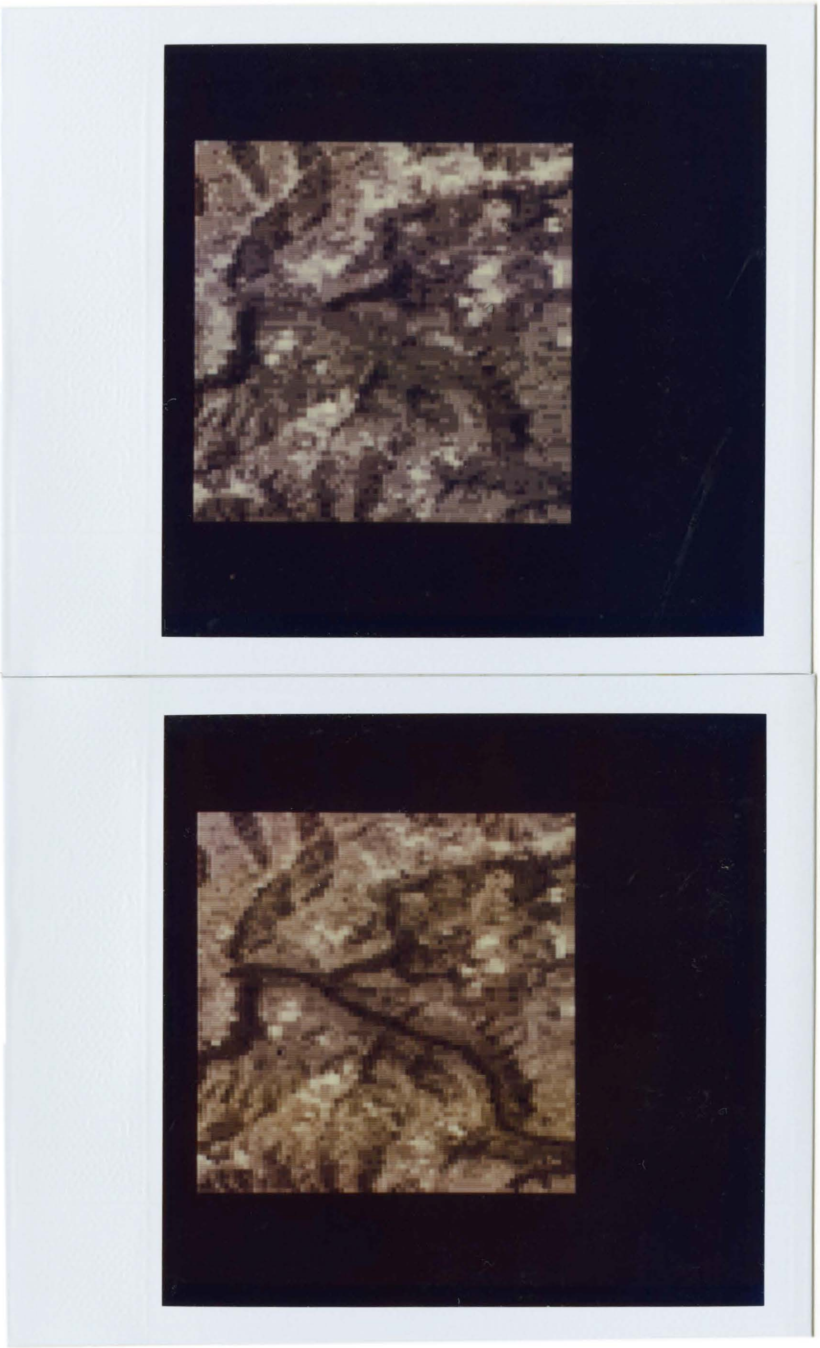


Figure 1.2.b - Test area 3.

(Top: band 4; bottom: band 5)

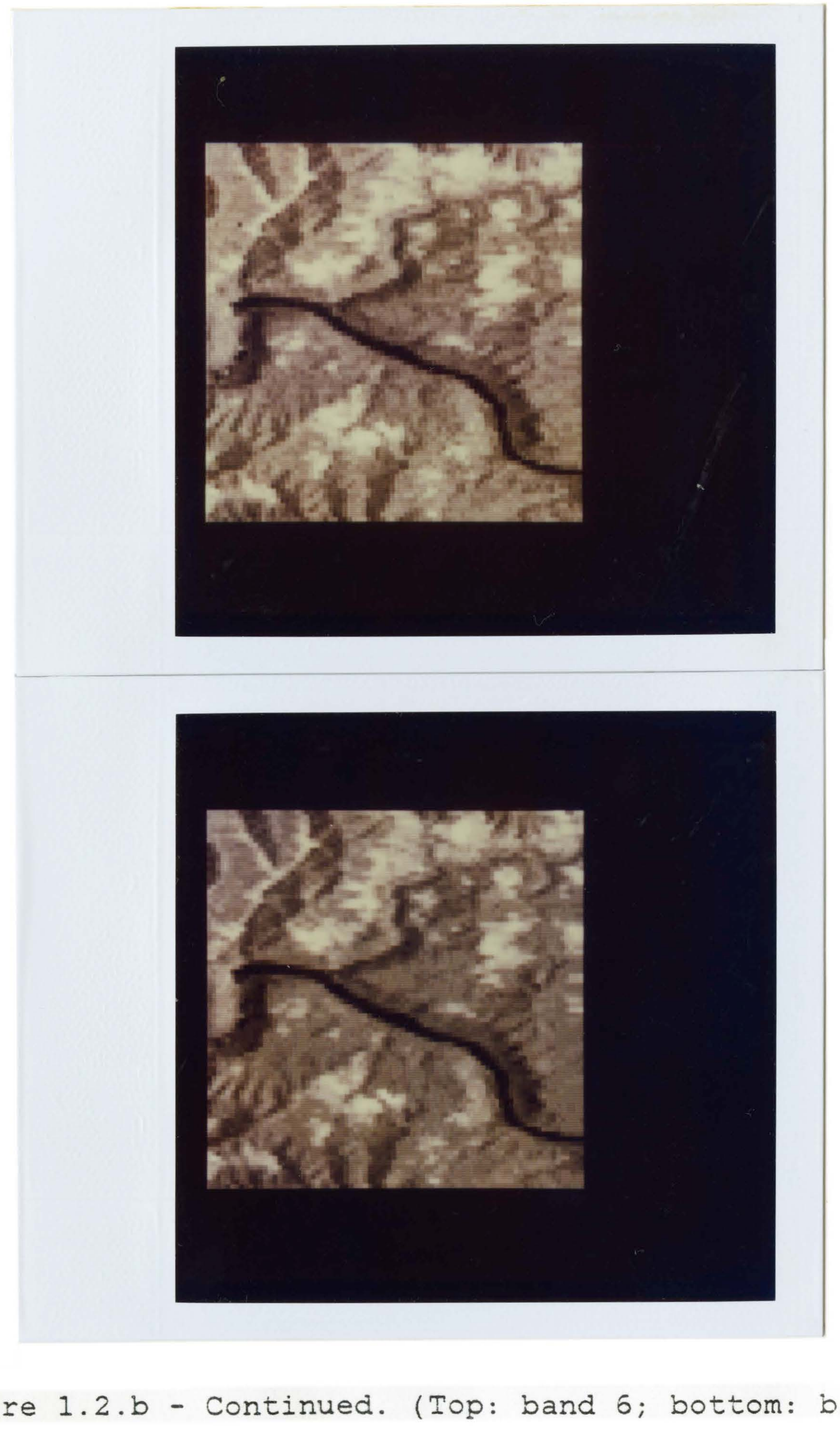


Figure 1.2.b - Continued. (Top: band 6; bottom: band 7)

Chapter II

ILLUMINATION MODEL

In order to develop really smart and comprehensive computer vision systems, it is necessary to utilize a reasonable model for photometry. In the early development of image understanding, researchers took the image brightnesses as data then processed them to find edges or regions without thinking about the physical meaning of the brightness changes. Recently, more people have tried to approach the problem based upon the physics of light.

Horn [1977] stressed that one cannot solve the image understanding problem by artificial intelligence alone. It is necessary to know the physical laws that underlie image formation. He used gradient space as a tool to analyze reflectance maps, edge-profiles, and lunar topography. Barrow and Tenenbaum [1978] suggested describing a scene in terms of intrinsic characteristics--such as reflectance, incident illumination, and orientation-- of the surface element. Because the intrinsic characterstics are confounded in the graytones of the image, they tried to recover them by exploiting assumptions and constraints from the physical nature of imaging and the world. Also they believed interpretation of boundaries played a key role in the recovery.

Two significant differences exist between the images used by AI researchers and the images used by remote sensing researchers. First, most AI researchers use single band images, but all remote sensing researchers use multi-band imagery. Second, the images used by AI researchers have more complicated illumination conditions, but for remote sensing imagery, there is always one illumination source--the sun--which can be considered to be a point source. These are also the two main reasons why the technique of extracting reflectance and topographic information [Eliason, Soderblom and Chavez, 1981] is so successful.

In the following, a modified model of Eliason, Soderblom, and Chavez will be described. This model differs from theirs mainly in including diffuse light and the extraction of a shadow image.

2.1 THE PROBLEM OF CONFOUNDED INFORMATION

Four kinds of information are mixed in LANDSAT imagery: surface reflectance, topography, diffuse light and haze. Components of any scene will possess varied reflectances over a given wavelength inteveral. For the scene considered here, differences in reflectance caused by varied surface materials are those due mainly to differences between open water, leaf-on vegetation (mainly unforested regions at lower elevation) and leaf-off vegetation (mainly forested regions at higher elevations). Assuming the ground surface is flat, the leaf-on vegetated areas have high reflectance for some spectral regions and appear as bright areas (especially in bands 6 and 7) to the LANDSAT sensor. The leaf-off vegetated areas possess intermediate, although generally low, brightness at MSS wavelengths. On the contrary, water areas have low reflectance and appear as dark areas to the LANDSAT sensor.

If topography is then considered, there is a pattern of directly illuminated and shadowed slopes due to the varied heights and orientations of the slopes. However, graytones for image pixels corresponding to shadowed locations are not zero because of the diffuse light coming indirectly from the sun. Finally, when light is reflected from the ground back to the sensor, there is additive haze due to atmospheric scattering. Thus, the brightness observed in the final scene are composed of several components that are not immediately separable. The following sections describe a method for isolating the brightness caused by each of these components.

2.2 THE MODEL FOR LANDSAT IMAGERY

The basic image model for a Lambertian surface illuminated by a point source is

$$G(x,y) = r(x,y)\ I \cos\theta(x,y) \qquad (2.1)$$

where G is the graytone in the image,

x,y are pixel coordinates,

r is surface reflectance,

I is the illumination flux from the sun, and

θ is the angle between sun incidence direction and surface normal (Figure 2.1).

Adding band number, diffuse light, and haze into this model, one has the general model for LANDSAT imagery as:

I. For directly illuminated pixels

$$G(x,y,b) = r(x,y,b)\ I(b) \cos\theta(x,y) + r(x,y,b)\ D(b) + H(b)$$

II. For shadowed pixels

$$G(x,y,b) = r(x,y,b)\ D(b) + H(b) \qquad (2.2)$$

where b is the spectral band number,

D is diffuse light, and

H is the haze due to atmospheric scattering.

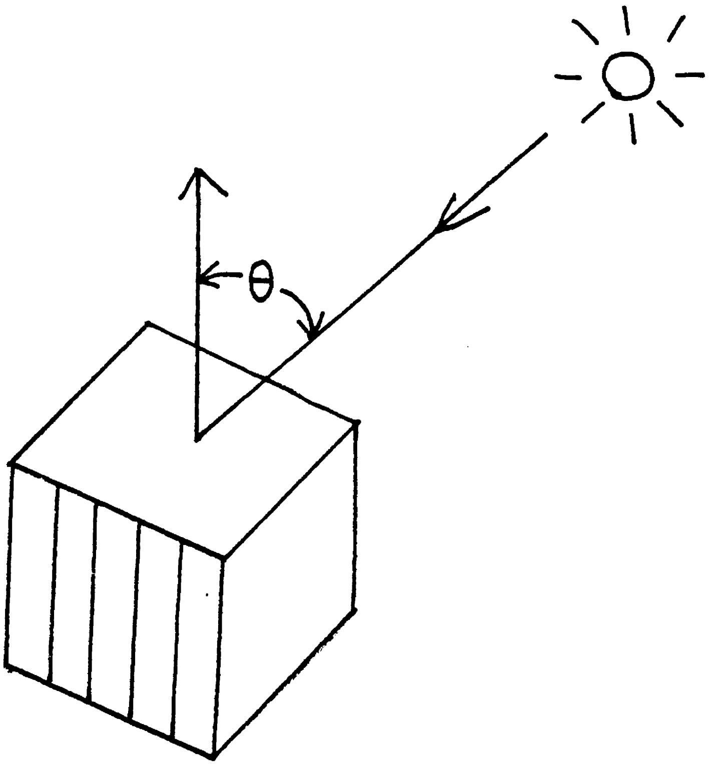


Figure 2.1 - θ is the angle between sun incidence direction and surface normal.

Because haze is an additive constant independent of pixel locations, a technique for haze removal will be described in the next section. After this, the problem of confounded information can be handled.

2.3 HAZE CORRECTION

Because haze is inversely proportional to the fourth power of wavelength, it is smallest for band 7 (.8 to 1.1 µm). Chavez [1975] used the correlation between bands paired individually with band 7 to estimate haze. Switzer, Kowalik, and Lyon [1981] used the correlation between all four bands of data simultaneously instead of in pairs. Their illumination model is

$$G(x,y,b) = R_a(b)\cos\theta(x,y) + H(b) + e(x,y,b) \qquad (2.3)$$

where R_a is proportional to the average surface reflectance, e is the residual unexplained variability including diffuse light, and θ, H are defined as in Equation 2.1.

From the known G(x,y,b) values, the method of least squares is used to choose unknown values of R_a, cosθ, and H which minimize $\sum_x \sum_y \sum_b e^2(x,y,b)$.

Differentiating the sum of e^2 with respect to cosθ(x,y), $R_a(b)$, and H(b) leads to

$$\sum_b G(x,y,b)R_a(b)-\cos\theta(x,y)\sum_b R_a^2(b)-\sum_b R_a(b)H(b) = 0 \qquad (2.3.1)$$

$$\sum_{x,y} G(x,y,b)\cos\theta(x,y)-R_a(b)\sum_{x,y}\cos^2\theta(x,y)$$

$$-H(b)\sum_{x,y}\cos\theta(x,y)=0 \qquad (2.3.2)$$

$$\sum_{x,y} G(x,y,b)-R_a(b)\sum_{x,y}\cos\theta(x,y)-nH(b)=0 \qquad (2.3.3)$$

where n is the total number of pixels.

From Equations 2.3.2 and 2.3.3

$$R_a(b)=\frac{\sum_{x,y}[G(x,y,b)(\cos\theta(x,y)-c')]}{\sum_{x,y}(\cos\theta(x,y)-c')^2} \qquad (2.3.4)$$

where c' is the average of $\cos\theta$ for all pixels.

From Equation 2.3.3

$$H(b)=G'(b)-R_a(b)c' \qquad (2.3.5)$$

where $G'(b)=(1/n)\sum_{x,y} G(x,y,b)$.

From Equations 2.3.1 and 2.3.5

$$\cos\theta(x,y)-c'=\frac{\sum_{b} R_a(b)(G(x,y,b)-G'(b))}{\sum_{b} R_a^2(b)} \qquad (2.3.6)$$

From Equations 2.3.4 and 2.3.6

$$R_a(b)=q\sum_{k} R_a(k)S_{bk}$$

where $q=\sum_{k} R_a^2(k)/[\sum_{jk}\sum R_a(j)R_a(k)S_{jk}]$

and $S_{jk}=(1/n)\sum_{x,y} (G(x,y,j)-G'(j))(G(x,y,k)-G'(k))$

For all four bands, the above equations can be put in a matrix form

$$[r_j] = q\ [S_{jk}]\ [r_k]$$

where $[r_j] = [R_a(4),R_a(5),R_a(6),R_a(7)]$ is a vector of size 4 and $[S_{jk}]$ is the covariance matrix of size 4 by 4. Thus the eigenvector of the covariance matrix $[S_{jk}]$ gives $[R_a(4),R_a(5),R_a(6),R_a(7)]$. If H(7) is assumed to be zero, then from Equation 2.3.5,

$c' = G'(7)/R_a(7)$

Thus $H(b) = G'(b) - G'(7)\ R_a(b)\ /\ R_a(7)$ (2.4)

for b = 4, 5, 6.

After haze is calculated, one has the haze corrected image G':

I. For directly illuminated pixels

$$
\begin{aligned}
& G'(x,y,b) \\
& = G(x,y,b) - H(b) \\
& = r(x,y,b)\ I(b)\ \cos\theta(x,y) + r(x,y,b)\ D(b)
\end{aligned}
$$

II. For shadowed pixels

$$G'(x,y,b) = r(x,y,b)\ D(b) \qquad (2.5)$$

The haze-corrected imagery of Figure 1.1 is shown in Figure 2.2.

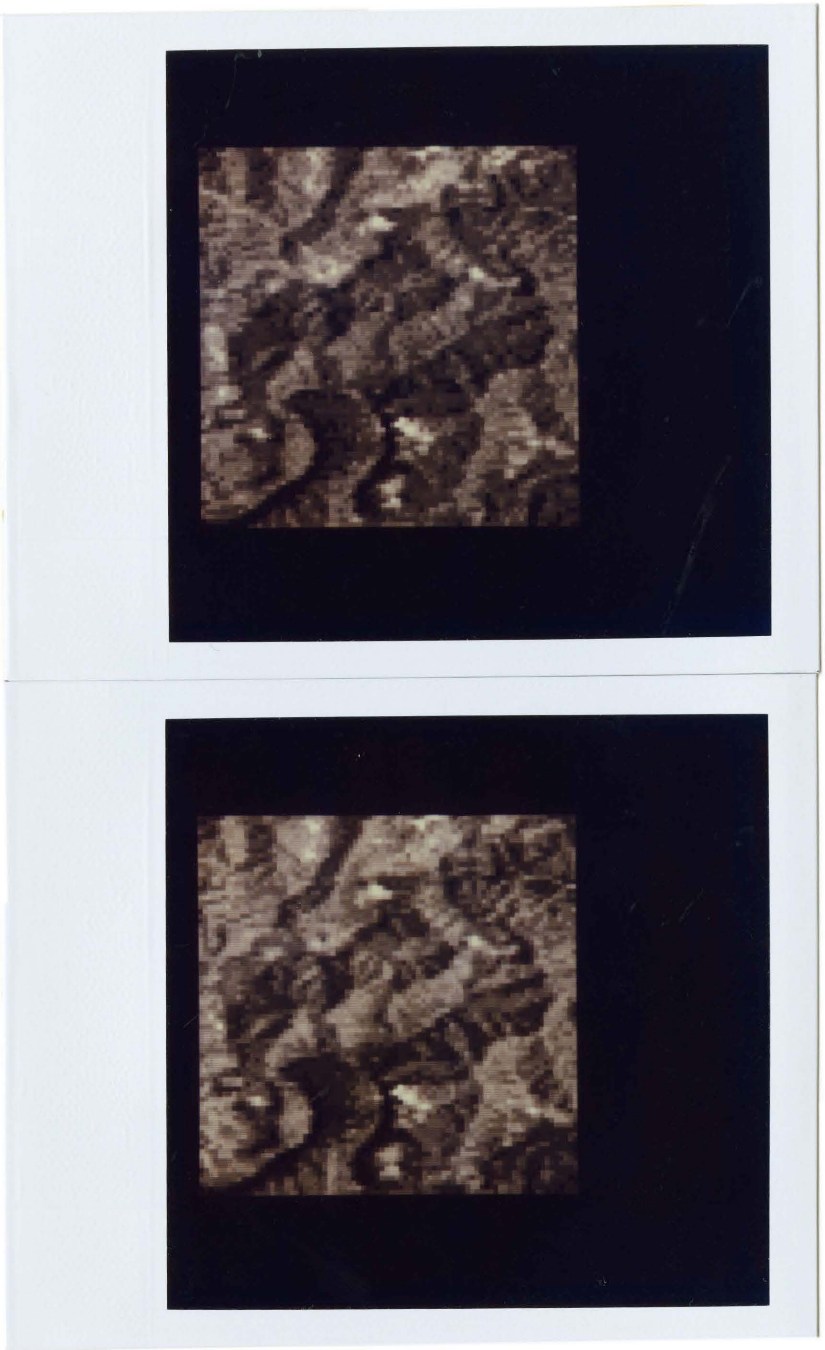


Figure 2.2 - Dehazed imagery.

(Top: band 4; bottom: band 5)

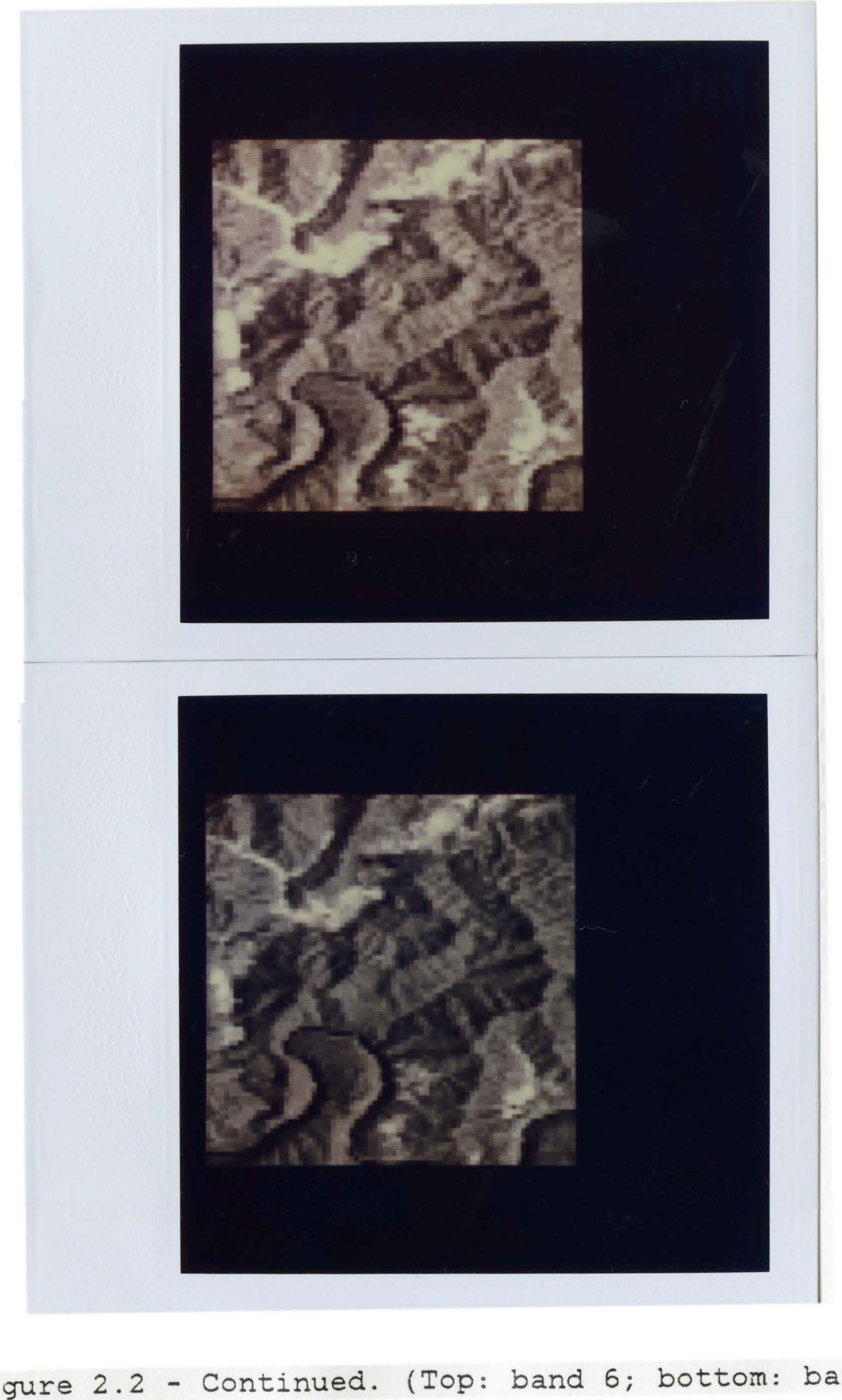


Figure 2.2 - Continued. (Top: band 6; bottom: band 7)

2.4 APPROACH TO THE PROBLEM OF CONFOUNDED INFORMATION

After haze is removed, it can be seen that unconfounding amounts to extracting a diffuse light image D_f which contains the information of r(x,y,b)D(b), a reflectance image R which contains the information of r(x,y,b)I(b), and a topographic modulation image T_p which contains the information of cosθ(x,y) such that

I. For directly illuminated pixels

$$G'(x,y,b) = R(x,y,b)\ T_p(x,y) + D_f(x,y,b)$$

II. For shadowed pixels

$$G'(x,y,b) = D_f(x,y,b) \qquad (2.6)$$

The first problem to be solved to accomplish this unconfounding is the determination of which pixels are directly illuminated from which pixels are in shadow. Once this is accomplished the unraveling can begin. For example, for the diffuse image, pixels which are in shadow take their value as the dehazed image value. Pixels which are directly illuminated take their value as the average dehazed image value taken over all shadowed pixels which are likely to be from the same material as they are. Once D_f and R'(estimated R) images are derived, T_p'(estimated T_p) image can be calculated as

$$T_p'(x,y,b) = \frac{G'(x,y,b) - D_f\ (x,y,b)}{R'(x,y,b)} \qquad (2.7)$$

The principal component image of this 4-band image can be used as $T_p(x,y)$. Finally,

$$R(x,y,b) = \frac{G'(x,y,b) - D_f(x,y,b)}{T_p(x,y)} \qquad (2.8)$$

To separate the shadow pixels from the directly illuminated pixels, one seeks to transform the images in a way in which the only effect is reflectance, then within groups of pixels with similar reflectance, the bright appearing ones can be separated from the dark appearing ones on the dehazed image. This two step technique is more accurate than a simple thresholding technique [Wang et. al., 1983].

One way to transform the image so that the only remaining effect is reflectance is to take ratios of one band to another. This is discussed in the next section.

2.5 RATIO IMAGES AND MATERIAL CLUSTERS

The ratio image has been widely used by remote sensing researchers to subdue surface topographic effects [Vincent, 1973; Raines, et al., 1978]. The crucial observation is that T_p is independent of band number. That is, even if the reflectance can be very different in different spectral regions, the topography will always be the same. Thus, from multi-band imagery, it is possible to first remove this component of topographic information. From Equation 2.5, the ratio image of two bands with band number b_1 and b_2 for directly illuminated pixels is

$$\frac{G'(x,y,b_1)}{G'(x,y,b_2)} = \frac{r(x,y,b_1)[I(b_1)\cos\theta(x,y) + D(b_1)]}{r(x,y,b_2)[I(b_2)\cos\theta(x,y) + D(b_2)]}$$

If one assumes illumination and diffuse light in bands b_1 and b_2 are related by

$$I(b_1) = a\, I(b_2),$$

$$D(b_1) = a\, D(b_2), \text{ then}$$

$$\frac{G'(x,y,b_1)}{G'(x,y,b_2)} = \frac{r(x,y,b_1)\ a\ [I(b_2)\cos\theta(x,y) + D(b_2)]}{r(x,y,b_2)\ \ [I(b_2)\cos\theta(x,y) + D(b_2)]}$$

$$= a \frac{r(x,y,b_1)}{r(x,y,b_2)} \qquad (2.9.1)$$

Similarly, the ratio image for shadowed pixels is

$$\frac{G'(x,y,b_1)}{G'(x,y,b_2)} = a \frac{r(x,y,b_1)}{r(x,y,b_2)} \qquad (2.9.2)$$

Thus, whether shadowed or directly illuminated, the ratio image is independent of cosθ. Three independent ratio images taken from the 4-band imagery in Figure 1.1 are shown in Figure 2.3. It can be clearly seen that the effects of shadows have been removed.

Since these three ratio images depend on material reflectance only, pixels of the same material reflectance will be grouped together if one performs clustering on them. The result is called a material cluster image M_c which is defined as

M_c: X x Y --> {1, 2, ..., N_c}

where X is the set of row coordinates,

Y is the set of column coordinates, and

N_c is the total number of clusters.

Each material cluster of cluster number cl can be defined as a set of pixels

$$C(cl) = \{(x,y) \mid M_c(x,y) = cl\}, \ 1<=cl<=N_c.$$

The material cluster image for Figure 1.1 is shown in Figure 2.4. The clustering method used to get this result will be described in Section 2.8. Once the material clusters are defined on the basis of the ratio images, one can find directly illuminated and shadowed pixels and define a binary shadow image. To do this, one collects together all dehazed 4-band pixel value belonging to a single material cluster and subcluster these 4-tuples into dark and bright subcluster classes. This procedure is the subject of next section.

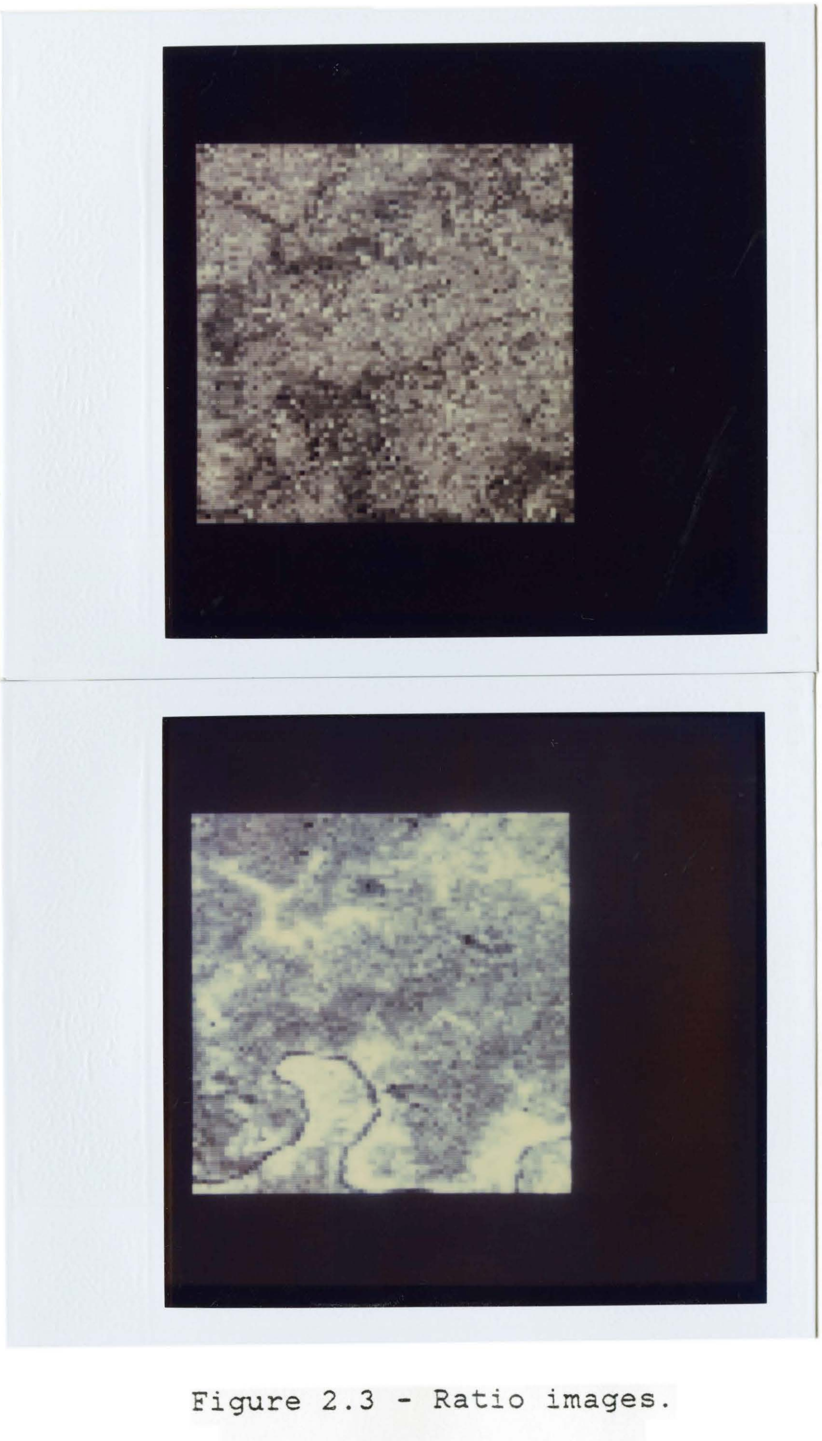


Figure 2.3 - Ratio images.

(Top: 5/4; bottom: 6/5)

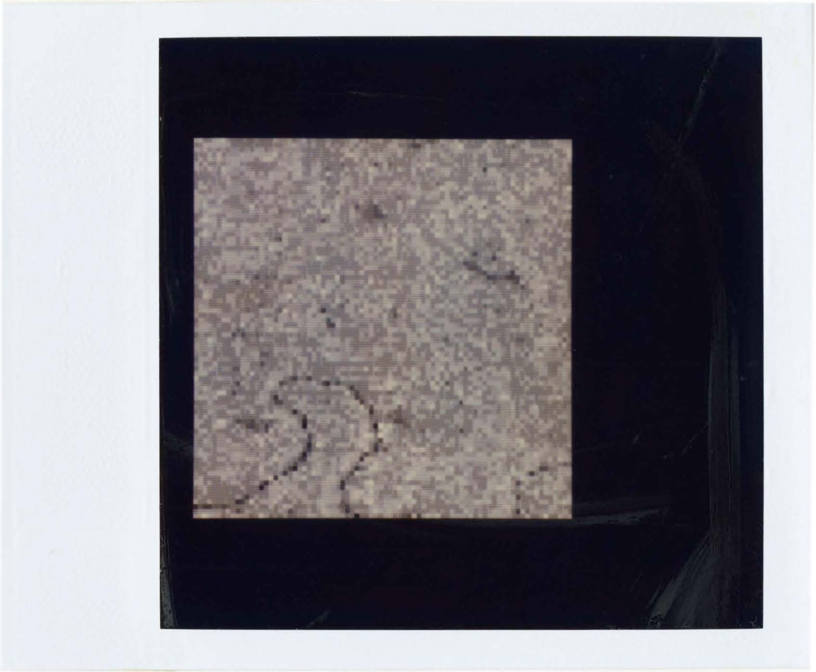


Figure 2.3 - Continued. (7/6)

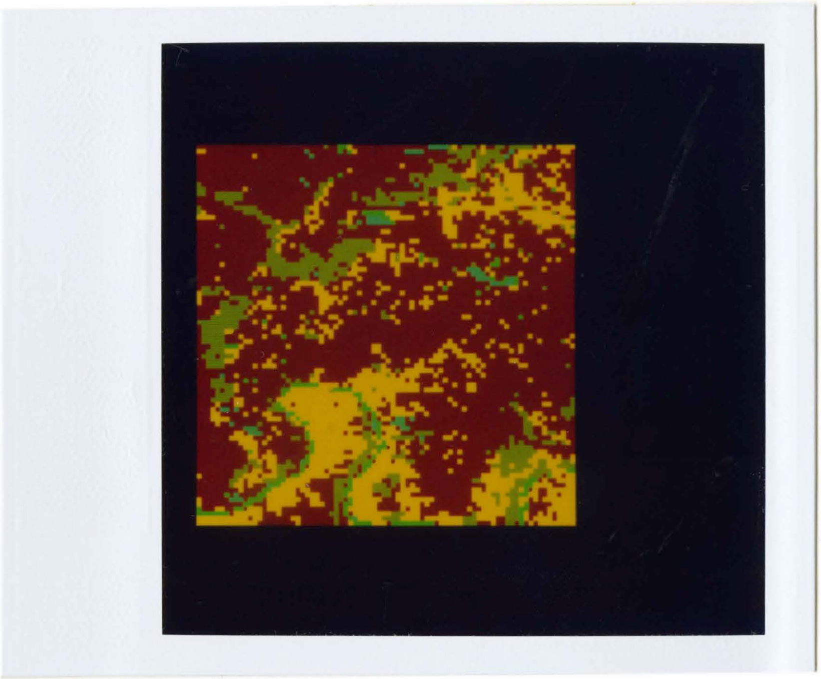


Figure 2.4 - Material cluster image.

2.6 SHADOW IMAGE

If one overlays the material cluster image M_c over any band of the dehazed image, one can see, within each material cluster, some pixels are bright and the others are dark. These differences are due to topographic variations; the bright pixels are directly illuminated pixels, and the dark pixels are in shadow. To separate the shadow pixels from the directly illuminated pixels, for each material cluster cl, one performs a subclustering on the dehazed pixel values in the set

$$\{G'(x,y,b) \mid M_c(x,y) = cl\}$$

which is the set of all dehazed values for pixels whose material cluster index is cl. This subclustering on cluster cl separates the directly illuminated pixels $C_0(cl)$ from the indirectly illuminated pixels $C_1(cl)$:

$$C_0(cl) = \{(x,y) \mid (x,y) \text{ is directly illuminated on the basis of the subclustering}\}$$

$$C_1(cl) = \{(x,y) \mid (x,y) \text{ is indirectly illuminated on the basis of the subclustering}\}$$

A shadow image S_w can be defined as

$$S_w : X \times Y \dashrightarrow \{0, 1\},$$

$$S_w(x,y) = \begin{cases} 0 \text{ if } (x,y) \varepsilon\ C_0(M_c(x,y)) \\ 1 \text{ if } (x,y) \varepsilon\ C_1(M_c(x,y)) \end{cases} \qquad (2.10)$$

The shadow image for Figure 1.1 is shown in Figure 2.5. The correspondence between this and topographic map is quite good. Now that directly illuminated and shadowed pixels have been identified, it is possible to use the dehazed image to get the diffuse image D_f, the reflectance image R, and the topographic modulation image T_p.

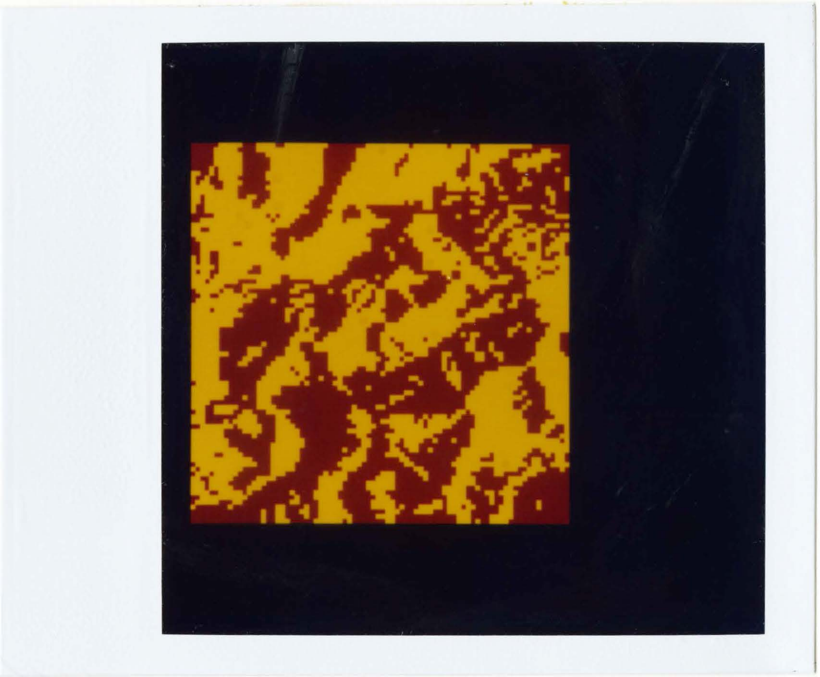


Figure 2.5 - Binary shadow image.

2.7 DIFFUSE LIGHT IMAGE, REFLECTANCE IMAGE, AND TOPOGRAPHIC MODULATION IMAGE

From Equation 2.6, shadowed pixels contain only the information of diffuse light, but directly illuminated pixels contain the information of both diffuse light and direct sun illumination. Also by the method of clustering, each pixel in the image belongs to a material cluster, and each material cluster has a bright and a dark subcluster. For pixels in the dark subcluster, one can simply use their dehazed values as their values in the diffuse image D_f. On the other hand, for pixels in the bright subcluster, one can define their value in the diffuse image to be the average value of all the pixels from the dark subcluster associated with the material cluster to which these pixels belong.

With this definition, the diffuse light image D_f is

I. For directly illuminated pixels

$$D_f(x,y,b) = \sum_{\substack{(u,v) \in C_1(cl) \\ cl=M_c(x,y)}} \frac{G'(u,v,b)}{\#C_1(cl)}$$

where, for a set S, #S means the size of this set.

II. For shadowed pixels

$$D_f(x,y,b) = G'(x,y,b)$$

If there were no variations in reflectance for pixels from the same material, one has

<u>ASSUMPTION 1</u>: $r(x,y,b)$ is a constant $r'(cl,b)$ for all pixels (x,y) in $C(cl)$, where $cl=M_c(x,y)$.

Using this assumption, for directly illuminated pixels,

$$D_f(x,y,b) = r'(cl,b)D(b) \sum_{(u,v)\,\varepsilon\, C_1(cl)} \frac{1}{\#C_1(cl)}$$

$$= r'(cl,b)\, D(b) \qquad (2.11)$$

From Equation 2.5, 2.11 and Assumption 1, $G'-D_f$ is

I. For directly illuminated pixels

$$G'(x,y,b) - D_f(x,y,b)$$

$$= r(x,y,b)I(b)\cos\theta(x,y) + r(x,y,b)D(b)$$

$$- r'(cl,b)\, D(b) \qquad (cl=M_c(x,y))$$

$$= r(x,y,b)I(b)\cos\theta(x,y)$$

II. For shadowed pixels,

$$G'(x,y,b) - D_f(x,y,b) = 0 \qquad (2.12)$$

An initial or raw estimated reflectance image R' can be calculated by assigning each pixel's value to be the average $G'-D_f$ value of all the pixels from the bright subcluster associated with the material cluster to which the pixel belongs.

$$R'(x,y,b) = \sum_{\substack{(u,v) \in C_0(cl) \\ cl = M_c(x,y)}} \frac{G'(u,v,b) - D_f(u,v,b)}{\#C_0(cl)}$$

$$= \sum_{(u,v) \in C_0(cl)} \frac{r(u,v,b)I(b)\cos\theta(u,v)}{\#C_0(cl)}$$

$$= r'(cl,b)I(b) \sum_{(u,v) \in C_0(cl)} \frac{\cos\theta(u,v)}{\#C_0(cl)}$$

Define $X_c(cl) = \sum_{(u,v) \in C_0(cl)} \frac{\cos\theta(u,v)}{\#C_0(cl)}$

$X_c(cl)$ is the spatial average of $\cos\theta$ for pixels in the bright subcluster.

Then

$$R'(x,y,b) = r'(cl,b)\ I(b)\ X_c(cl) \qquad (2.13)$$

The raw estimated reflectance images can be used to produce raw estimates of the topographic modulation image. By Equations 2.7, 2.12, and 2.13, the raw estimated topographic modulation image T_p' for band b is

I. For directly illuminated pixels

$$T_p'(x,y,b) = \frac{G'(x,y,b) - D_f(x,y,b)}{R'(x,y,b)}$$

$$= \frac{r(x,y,b)\ I(b)\ \cos\theta(x,y)}{r'(cl,b)\ I(b)\ X_c(cl)} \qquad (cl=M_c(x,y))$$

$$= \frac{\cos\theta(x,y)}{X_c(cl)}$$

II. For shadowed pixels

$$T_p'(x,y,b) = 0 \qquad (2.14)$$

The D_f, R', and T_p' images for Figure 1.1 are shown in Figures 2.6, 2.7, and 2.8.

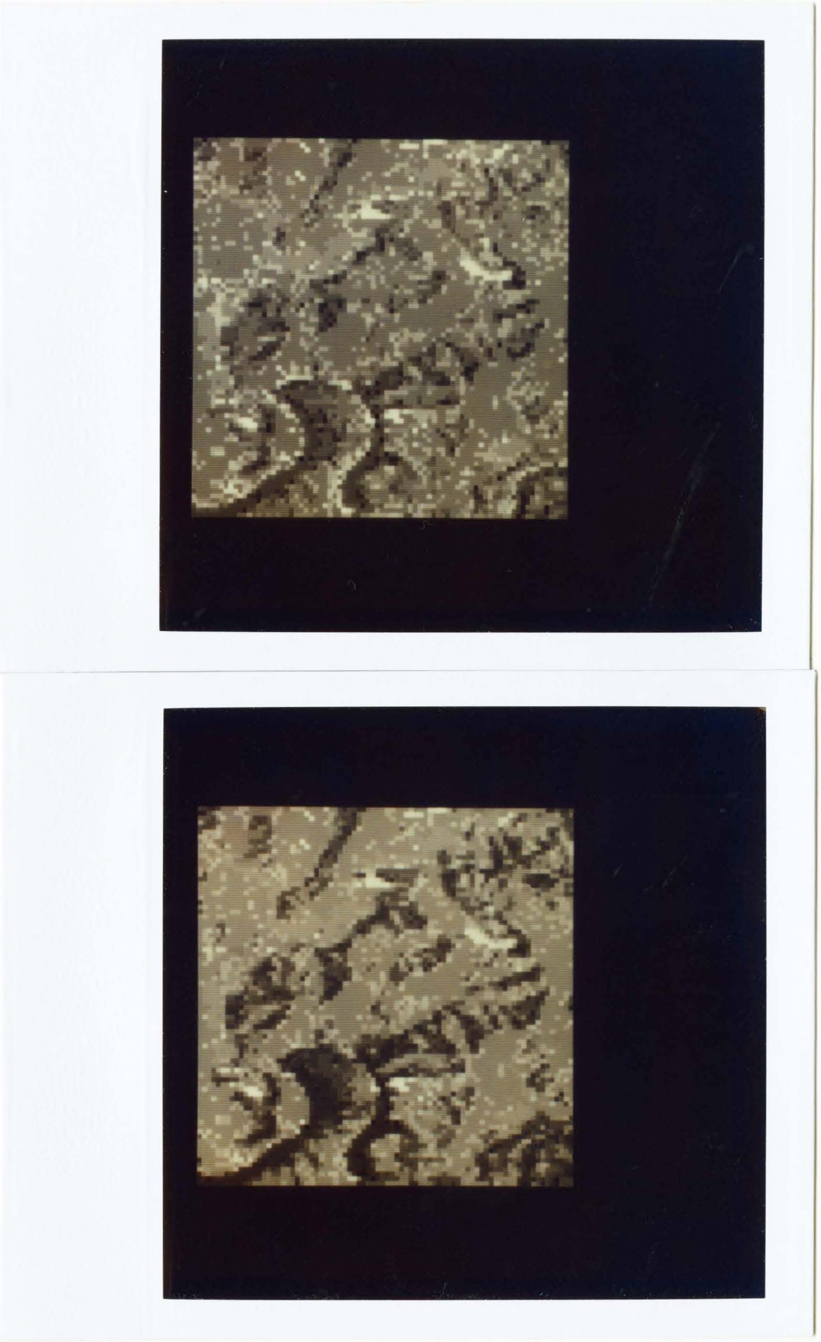


Figure 2.6 - Diffuse light image.

(Top: band 4; bottom: band 5)

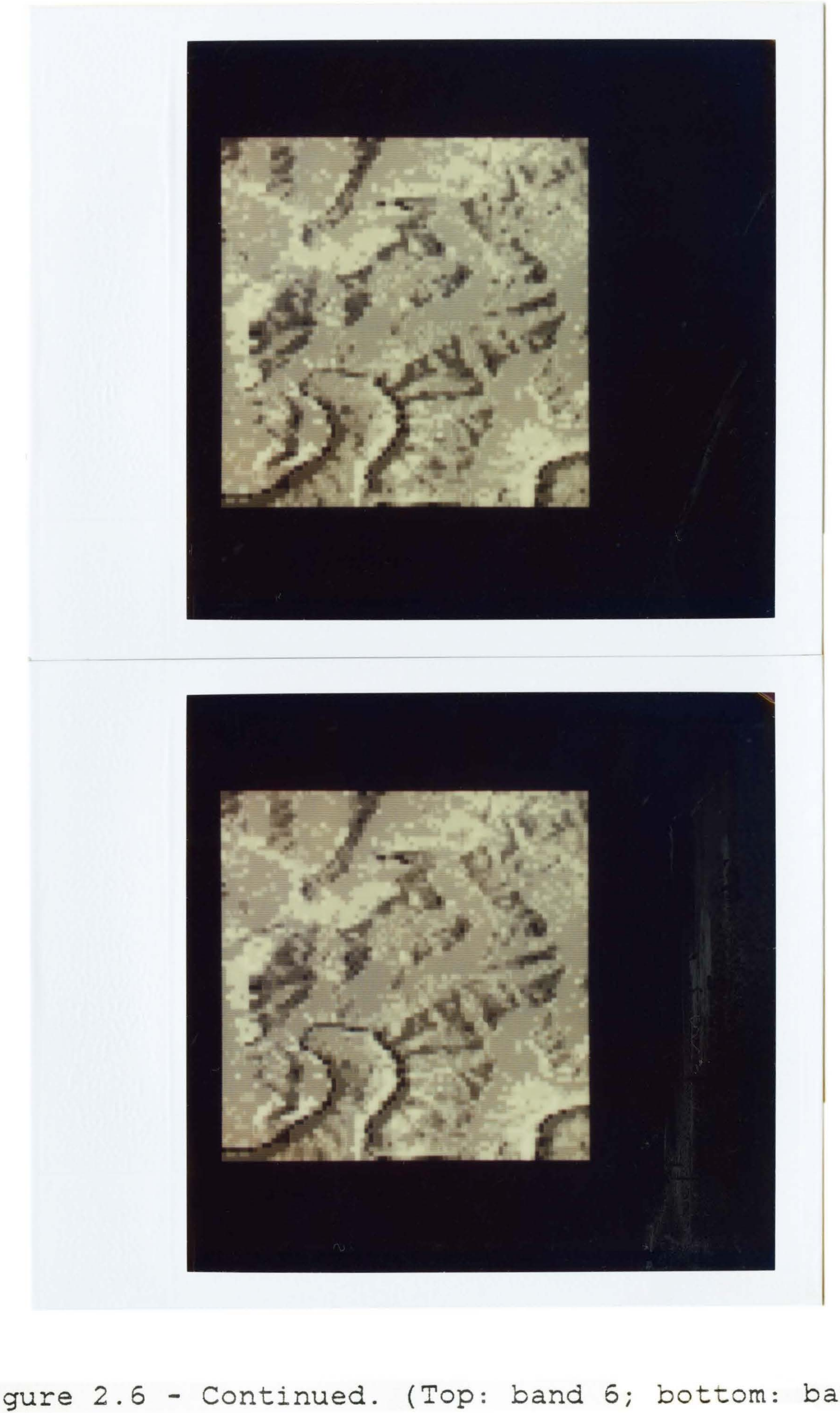


Figure 2.6 - Continued. (Top: band 6; bottom: band 7)

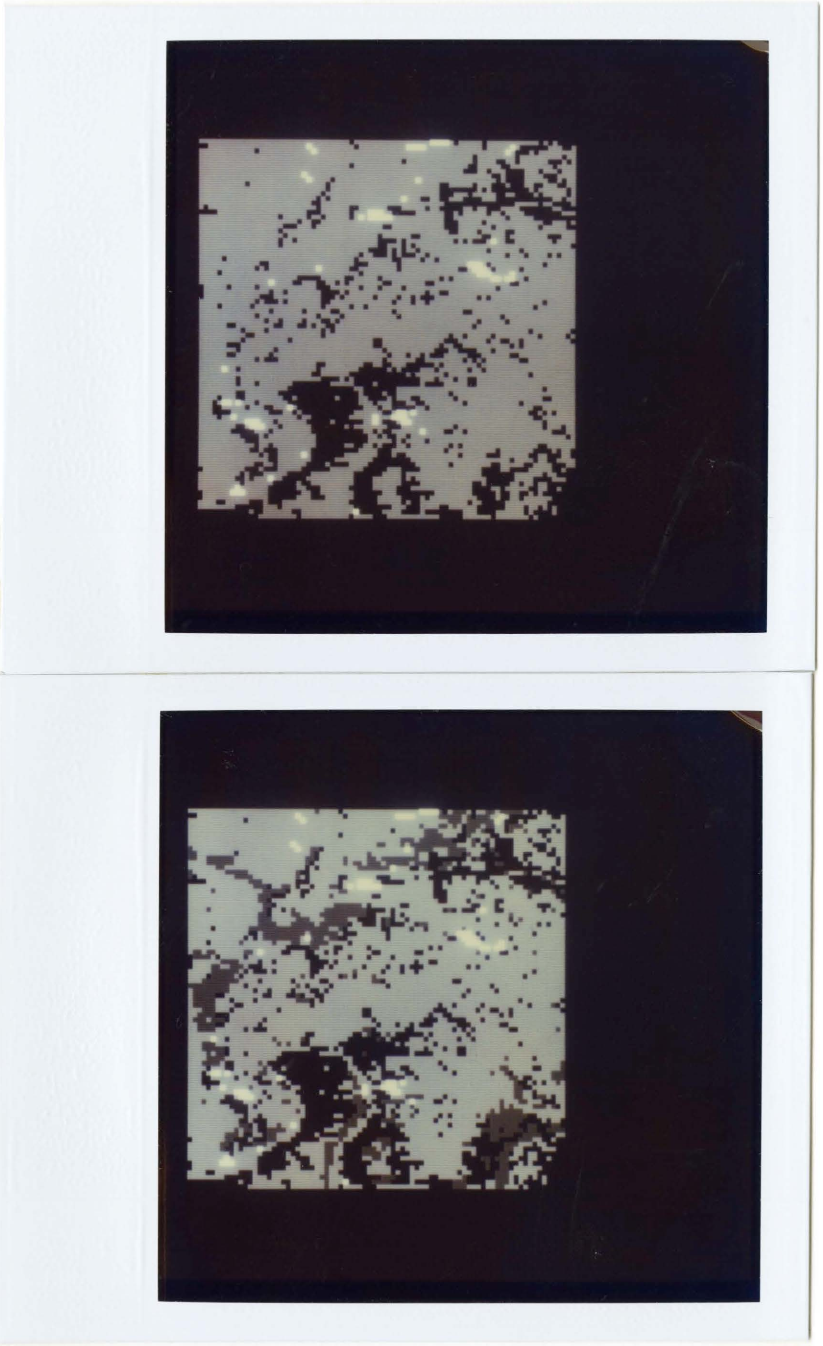


Figure 2.7 - Estimated reflectance image.

(Top: band 4; bottom: band 5)

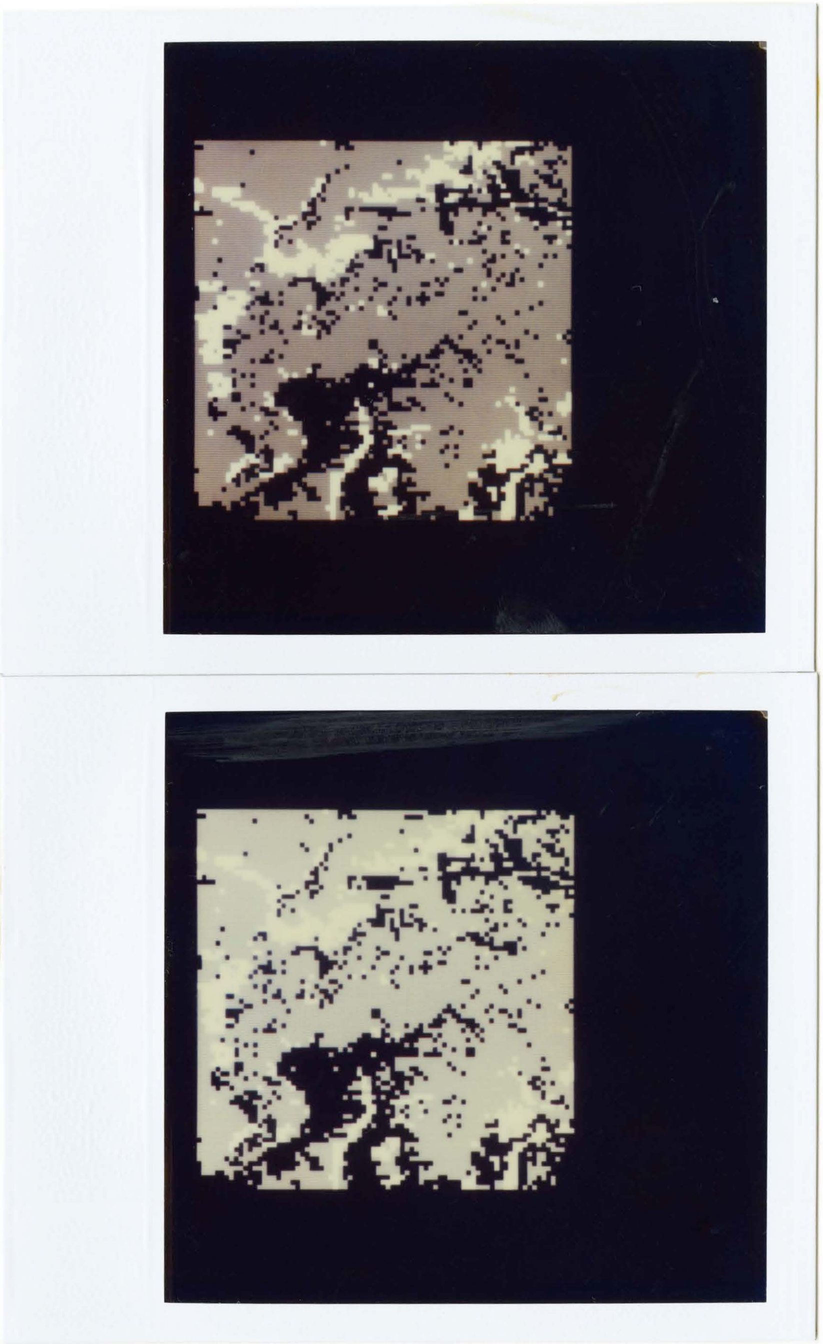


Figure 2.7 - Continued. (Top: band 6; bottom: band 7)

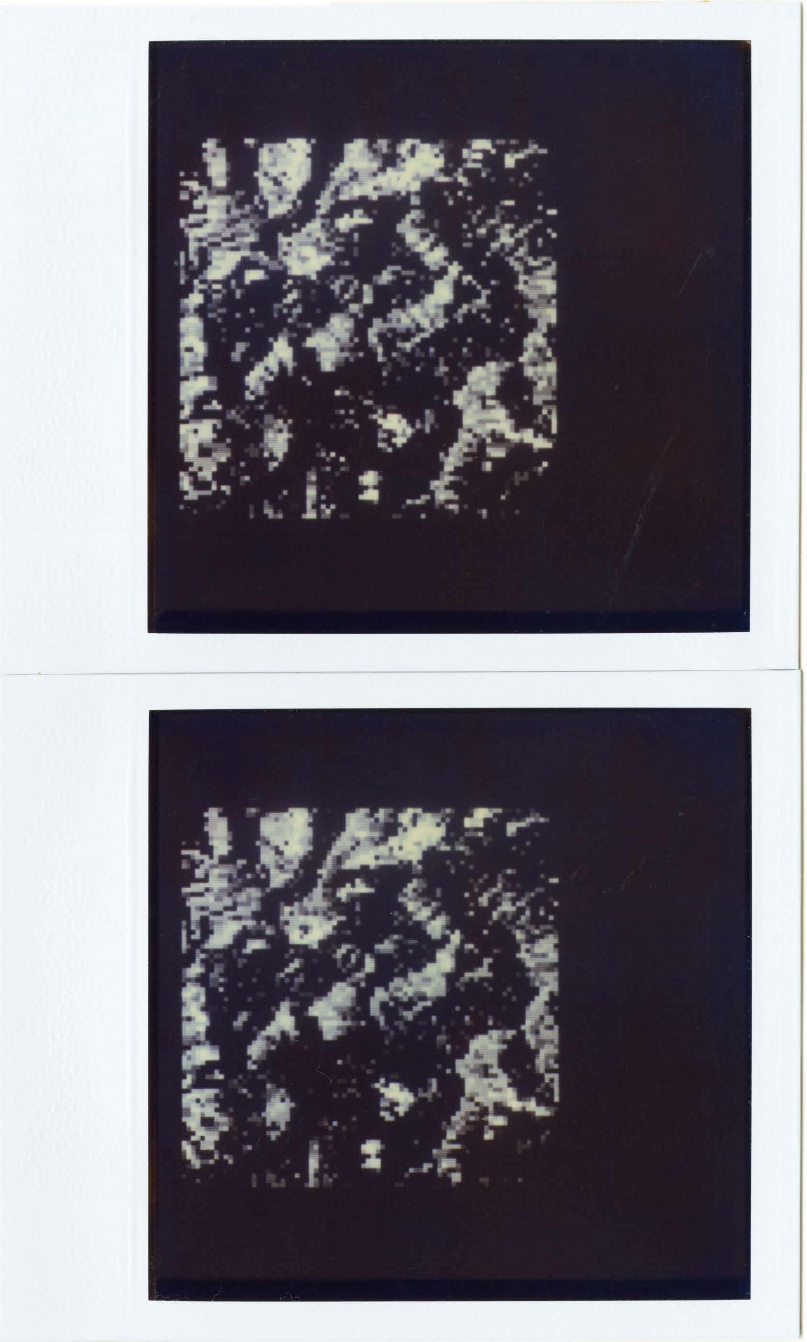


Figure 2.8 - Estimated topographic modulation image.

(Top: band 4; bottom: band 5)

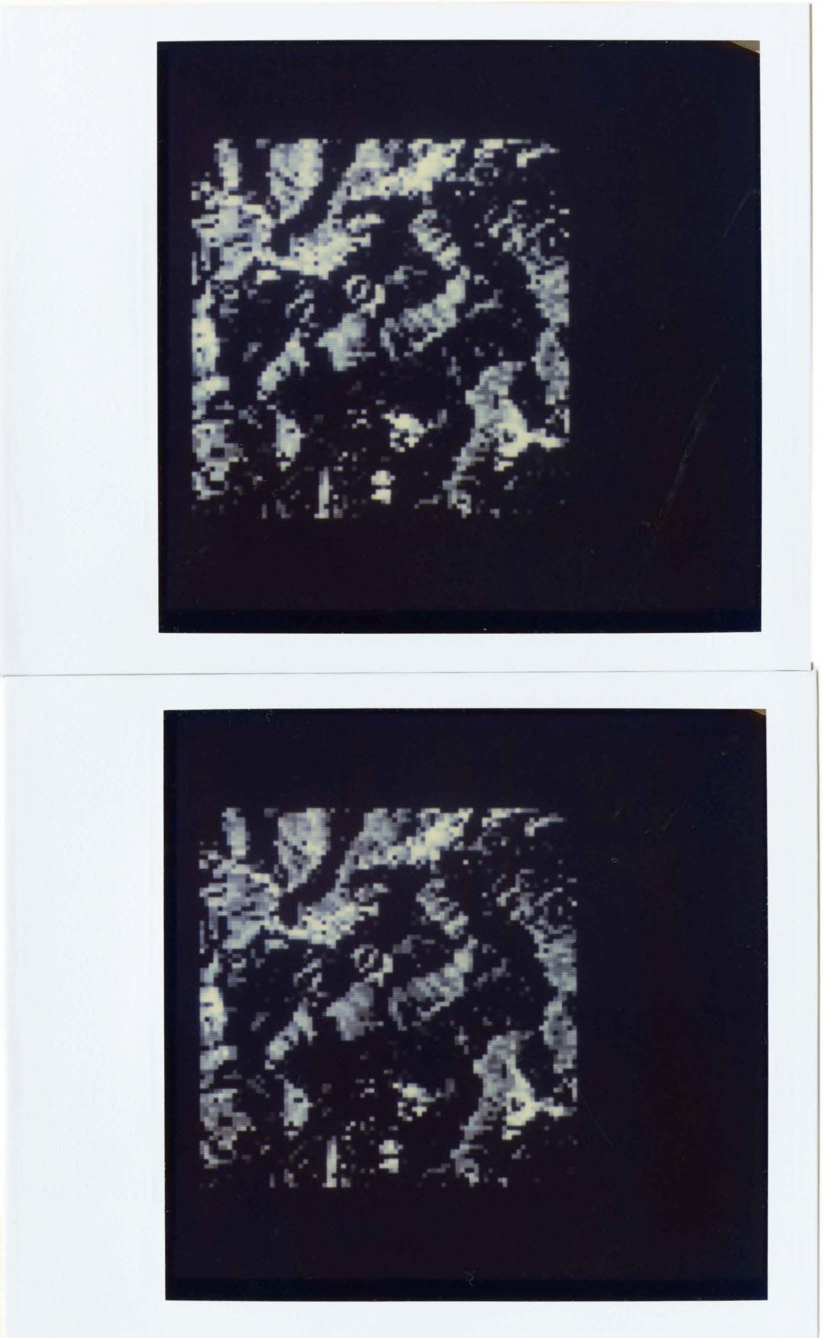


Figure 2.8 - Continued. (Top: band 6; bottom: band 7)

The following assumption is made to convert the raw topographic images to a final estimated topographic modulation image.

ASSUMPTION 2: $X_c(cl)$ is the the same for all material clusters, $1<=cl<=N_c$.

The evaluation of this assumption will be done in Section 4.5 when an elevation model is reconstructed.

With this assumption, a principal component image can be computed as the final single-band estimated T_p image. First, a covariance matrix of the 4-band T_p' image is computed. Eigenvalues and eigenvector are then computed from this covariance matrix. If the eigenvector corresponding to the largest eigenvalue is $[x_4,x_5,x_6,x_7]$, then

I. For directly illuminated pixels,

$$T_p(x,y) = \sum_{i=4}^{7} x_i \, T_p'(x,y,i)$$

$$= k \cos\theta(x,y) \text{ for a constant } k \qquad (2.15)$$

II. For shadowed pixels,

$$T_p\ (x,y) = 0$$

Having the topographic modulation image, the final estimated reflectance image is easily computed using Equation 2.8,

I. For directly illuminated pixels,

$$R(x,y,b) = \frac{G'(x,y,b) - D_f(x,y,b)}{T_p(x,y)}$$

$$= \frac{r(x,y,b)\ I(b)\ \cos\theta(x,y)}{k\ \cos\theta(x,y)}$$

$$= \frac{r(x,y,b)\ I(b)}{k} \qquad (2.16)$$

II. For shadowed pixels,

$$R(x,y,b) = \sum_{\substack{(u,v)\ \varepsilon\ C_0(cl) \\ cl = M_c\ (x,y)}} \frac{R(u,v,b)}{\#C_0(cl)}$$

$$= \frac{I(b)}{k} \sum_{(u,v)\ \varepsilon\ C_0(cl)} \frac{r(u,v,b)}{\#C_0(cl)}$$

The T_p image is shown in Figure 2.9, and R image is shown in Figure 2.10.

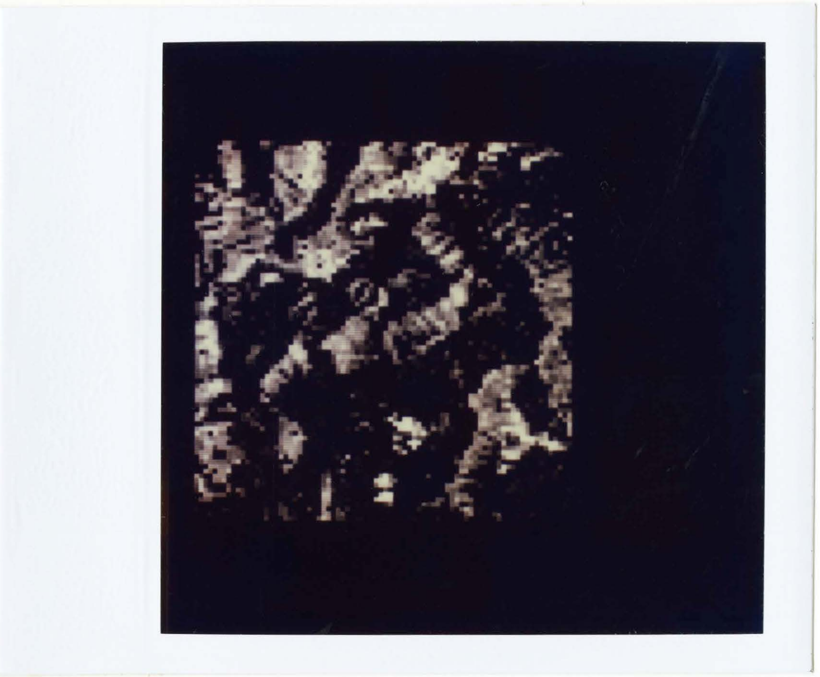


Figure 2.9 - Topographic modulation image.

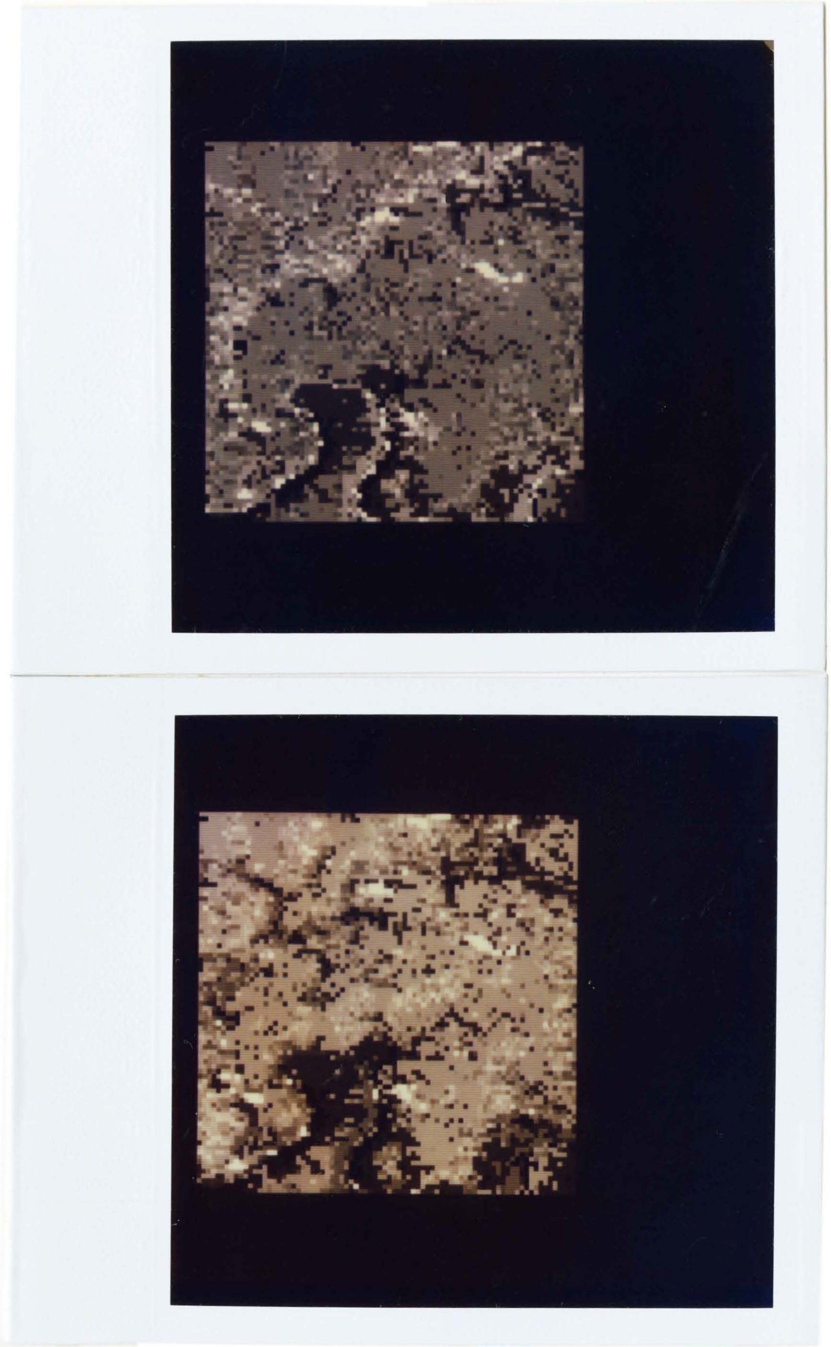


Figure 2.10 - Reflectance image.

(Top: band 4; bottom: band 5)

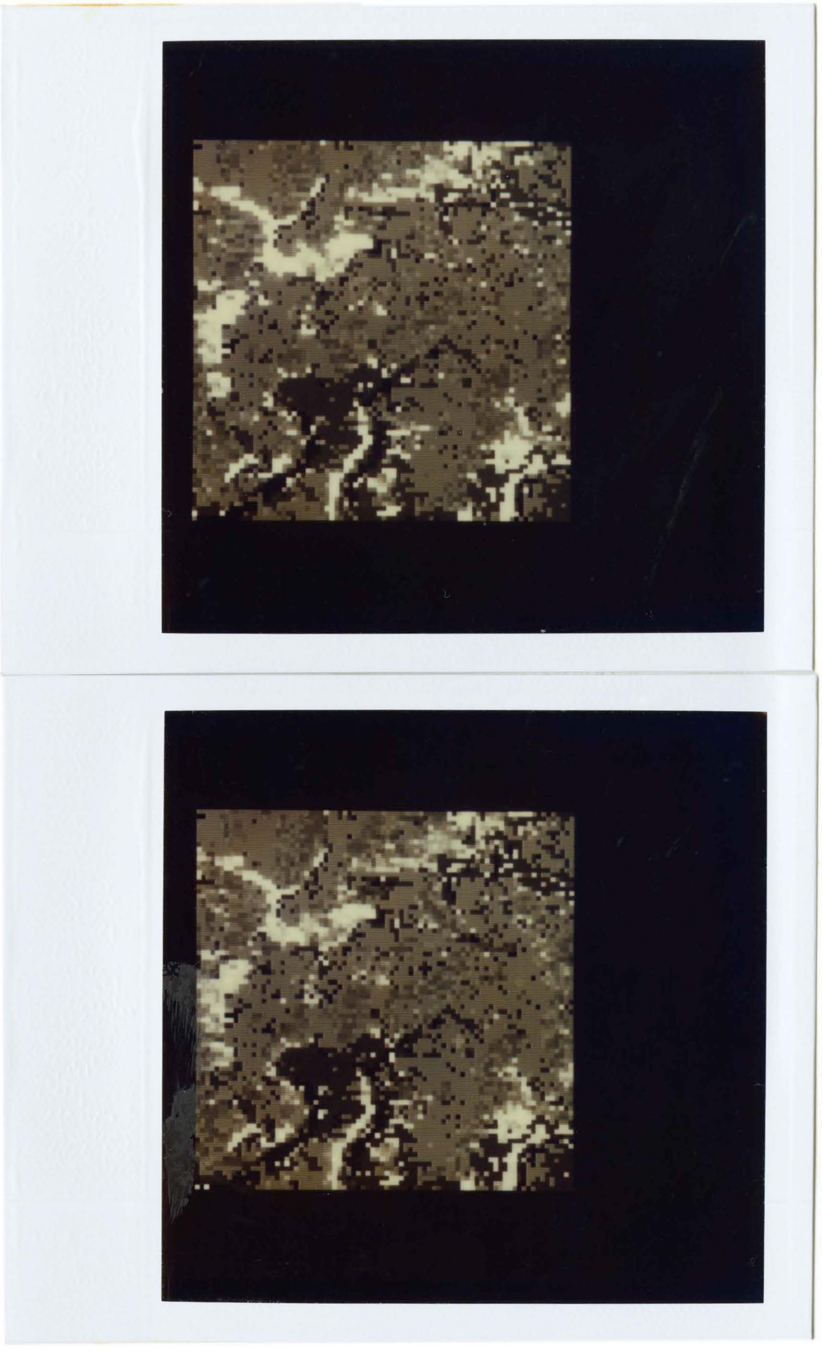


Figure 2.10 - Continued. (Top: band 6; bottom: band 7)

Instead of using a principal component technique to get the T_p image in Equation 2.15, a Bayesian method can be used to estimate this. From Equation 2.14 and Assumption 2, $X_c'\ T_p'(x,y,b)$, $b = 4$ to 7, are four samples of the true $\cos\theta(x,y)$ value, where X_c' is a constant for the spatial average of $\cos\theta$.

Let $\cos\theta'(x,y)=(1/4)\sum_{b=4}^{7}[X_c'T_p'(x,y,b)]$ be the sample mean

which can be used in the Bayes equation

$$P(\cos\theta(x,y)|\cos\theta'(x,y))$$

$$=\frac{P(\cos\theta'(x,y)|\cos\theta(x,y))\ P(\cos\theta(x,y))}{\int P(\cos\theta'(x,y)|\cos\theta(x,y))\ P(\cos\theta(x,y))} \qquad (2.17)$$

$P(\cos\theta'(x,y)|\cos\theta(x,y))$ can be written as

$$k_1 e^{k_2\cos^2(\theta(x,y)-\theta'(x,y))} \qquad \text{for constants } k_1,\ k_2.$$

The distributions of the first order partial derivatives f_x, f_y for the area of Figure 4.13 which contains the true elevation data for test area 1 is examined and found to be Gaussian-like. Together with Equation 4.3, it is assumed that the density function $P(\cos\theta(x,y))$ has normal distribution. Using Table 4.3, a estimate mean of 0.7366 and a estimate variance of 0.3 are used for this normal distribution. Given these, the value of cosθ(x,y) which maximize the value of Equation 2.17 can be computed as the final estimated cosθ value at the pixel location (x,y). This is computed by using the MINPACK routine LMDIF which implements the Levenberg-Marquardt algorithm. The resulting T_p image is shown in Figure 2.11.

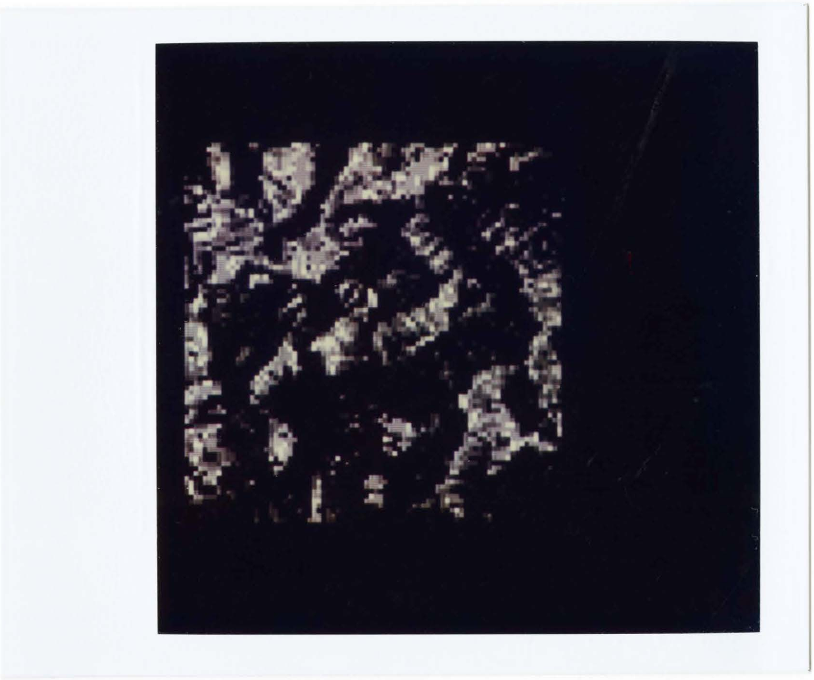


Figure 2.11 - Topographic modulation image by Bayesian method.

2.8 CLUSTERING

In the noisy ratio images of Figure 2.3, there are three major clusters: water area, leaf-on vegetated area, and leaf-off vegetated area. The size of water area is much smaller than that of the other two. For this reason, the mode approach [Goldberg and Shlien, 1978] which uses a fixed threshold to get cluster centers does not work. AMOEBA [Bryant, 1979] works better, but fails to obtain unbroken river segments. It was found that ISODATA modified in such a way that class sizes are also taken into consideration works best, despite its simplicity.

The algorithm of basic ISODATA is given in the textbook by Duda and Hart [1973]:

1. Choose some initial values for the means u_1, u_2, ..., u_c, where c is the total number of classes.
2. Classify the samples by assigning them to the class of the closest mean.
3. Recompute the means as the average of the samples in their class.
4. If any mean changed value, go to 2; otherwise, stop.

The modified ISODATA algorithm is

1. Choose some initial values for the means u_1, ..., u_c, and sizes z_1, ..., z_c, where c is the total number of classes.

Set ACTIVE(i) = 1, 1 <= i <= c.

2. Classify the samples by assigning them to the class of closest mean.

3. Recompute the means as the average of the samples in their class. Also compute the size of each class.

4. Set ACTIVE(i) = 0 if size of class i is greater than z_i. If no mean changed value or ACTIVE(i) = 0 for all i, stop.

5. For samples which are in class i with ACTIVE(i) = 1, classify them by assigning them to the class j of the closest mean, ACTIVE(j) = 1.

Go to 3.

Now the crucial point is how to select the initial mean and size for each class. One simple yet powerful method is developed by the inspiration of looking at the pseudocolor image [Moik, 1980] which is created by assigning a color to a pixel (x,y) according to the quantized value of $G(x,y,b_1)$, the quantized value of $G(x,y,b_2)$, and the quantized value of $G(x,y,b_3)$ for a three-band graytone imagery. The result even can be considered to be a reasonable cluster image.

Thus, the following recursive algorithm was developed to compute the total number of classes as well as initial class means and sizes.

1. For a n-band imagery, find the minimum graytone value MIN and maximum graytone value MAX for each band i and quantize the graytones $G(x,y,b_i)$ into q+1 values: $Q(1,b_i) = MIN$, ..., $Q(q+1,b_i)=MAX$.

   Calculate the total number of pixels in sets

$$
\begin{aligned}
\{(x,y) \mid\ & Q(a_1,b_1) <= G(x,y,b_1) <= Q(a_1+1,\ b_1),\ 1<= a_1 <= q, \text{ and}\\
& Q(a_2,b_2) <= G(x,y,b_2) <= Q(a_2+1,\ b_2),\ 1<= a_2 <= q, \text{ and}\\
& \qquad \ldots\\
& Q(a_n,b_n) <= G(x,y,b_n) <= Q(a_n+1,\ b_n),\ 1 <= a_n <= q\ \}
\end{aligned}
$$

   Also claculate the graytone means for each set.

   Mark all the sets as active.

2. Find the set with highest number of pixels among the active sets. This is a cluster center. Its mean can be used as initial class mean in ISODATA. An estimated size of this class is the sum of its number of pixels and the numbers of pixels of its neighbors.

3. If no peak can be found, stop; otherwise mark this peak set and its neighbors as inactive and go to 2.

Feeding the class means and sizes to the updated ISODATA, one gets material cluster image of Figure 2.4.

The subclustering of getting C_0, C_1 in Section 2.6 needs only the basic ISODATA program. In this case, the initial class mean for C_1 includes all the minimum graytones for four bands, and the initial mean for C_0 includes all the maximum graytones for four bands. This is how the shadow image of Figure 2.5 was created.

Chapter III

DETECTION OF VISIBLE RIVERS

Because one pixel in the LANDSAT image represents approximately a 57 meter by 80 meter area on the ground, the resolution is low. For the most part, it is not possible to directly observe the drainage network of the LANDSAT data. If rivers or lakes are visible to humans, they can be detected by spectral information as described in this section. Such rivers (including big lakes) are called visible rivers.

Once the material reflectance image is created by the technique in last chapter, it can be used to identify visible rivers by the following process [Alfoldi and Munday, 1978].

(1) A band 4 green coefficient x of every pixel is calculated as the ratio of the radiance of band 4 over the radiance sum of bands 4, 5 and 6. Similarly a band 5 red coefficient y is calculated for every pixel. x and y are called LANDSAT chromaticity coordinates.

(2) In this coordinate system, Munday [1974] has determined a curve (Figure 3.1) which is the locus of the positions of chromaticity values of water bodies. If, for some pixels, the x, y values calculated in 1 are close to this curve, then those pixels can be identified as portions of water bodies.

The chromaticity image created from Figure 2.10 is shown in Figure 3.2. The image of the distance between chromaticity coordinates (x,y) and Munday's curve is shown in Figure 3.3. Because the darker a pixel is, the lower the graytone (distance) is, darker pixels indicate they are water pixels.

Once the river segments are found, they can be chosen as having the lowest elevations in the image window and be used as a basis for elevation reconstruction. This is discussed fully in next chapter. In Chapter 5, the technique of determining flow directions of visible rivers will be described.

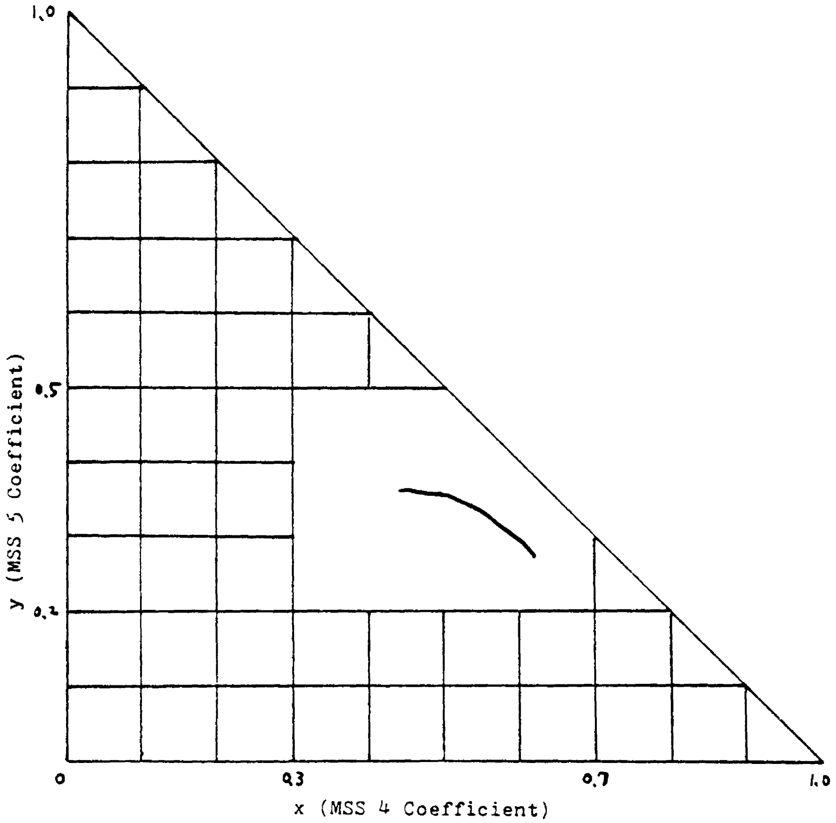


Figure 3.1 - Chromaticity plot.

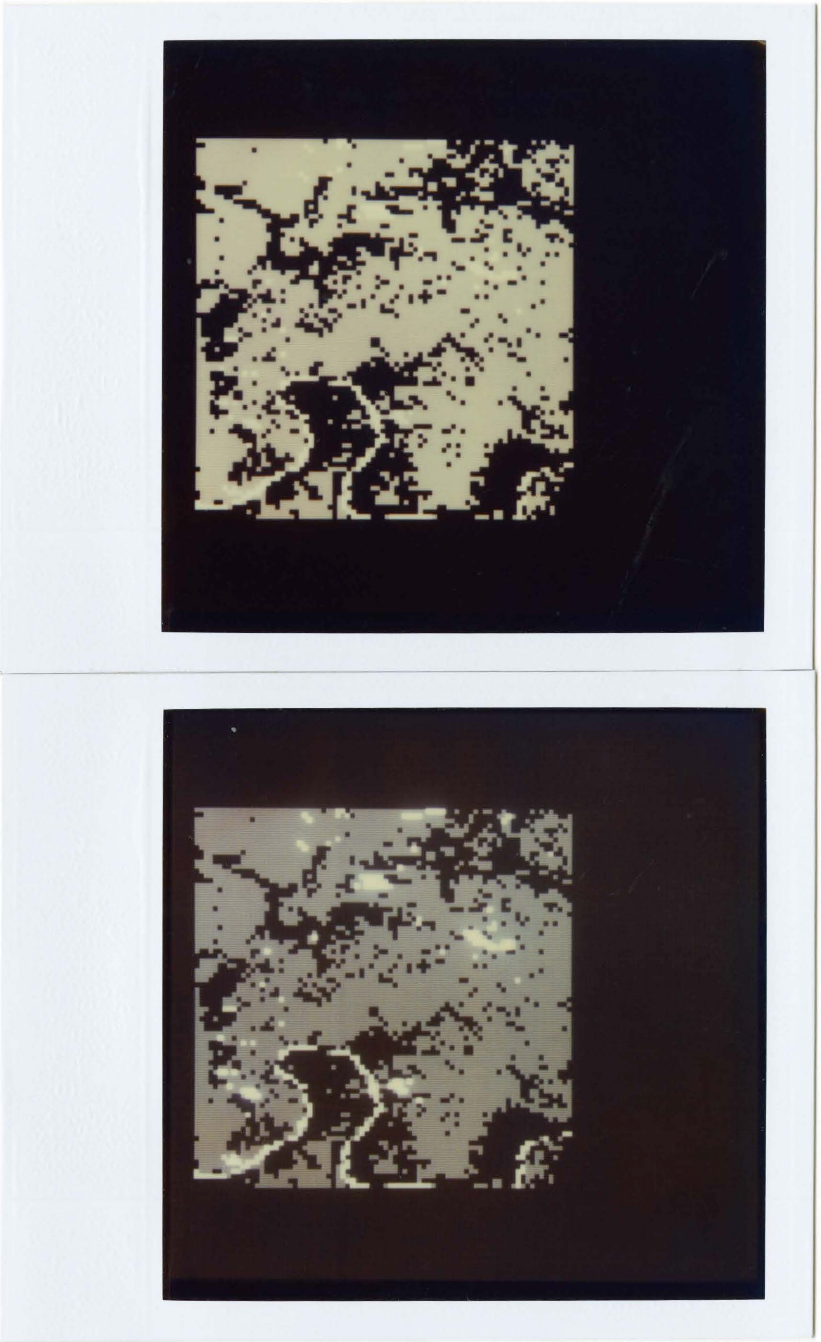


Figure 3.2 - Chromaticity coordinates image--y and x.

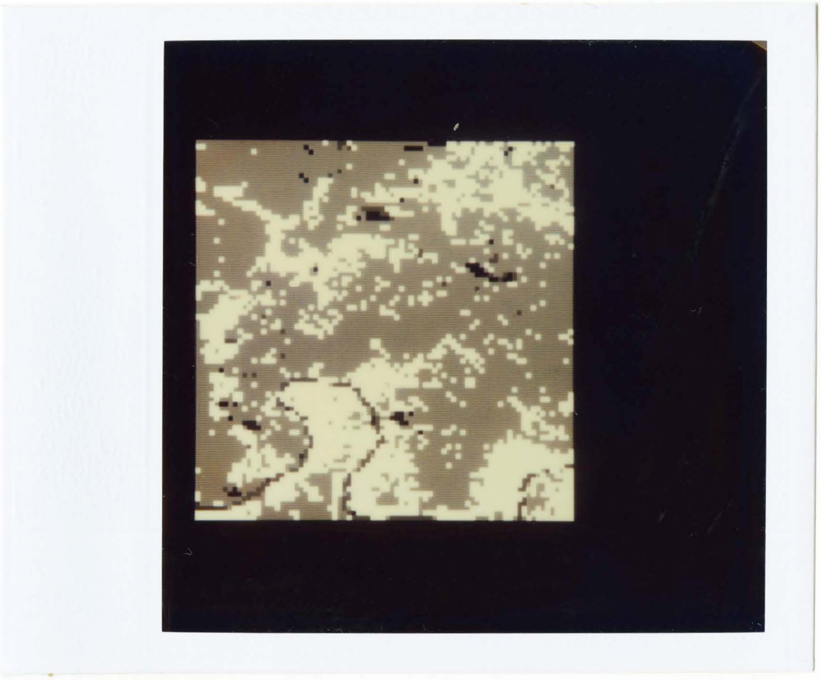


Figure 3.3 - Distance image. Dark pixels indicate water pixels.

Chapter IV

ELEVATION RECONSTRUCTION

The detection and use of spectral information for finding water regions is the major subject of the last two chapters. This chapter focuses on the use of spatial information for elevation reconstruction. Section 1 describes how ridges, valleys, and peak junctions are detected. In Section 2, it is shown how relative elevations of ridges and valleys are assigned by the elevation growing process. In Section 3, three techniques are used to find interpolated values for pixels between ridges and valleys. In Section 4, it is shown how an iterative Bayesian technique is used to improve the result of Section 3 based on the topographic modulation image of Chapter 2. After an elevation model is computed, Assumption 2 of Chapter 2 is evaluated and the original LANDSAT imagery is reconstructed to evaluate the illumination model of Chapter 2.

4.1 RIDGE-VALLEY MAP

Sides of hillsides facing the sun must be directly illuminated. Sides of hillsides facing away from the sun must be indirectly illuminated. Directly illuminated to indirectly illuminated transition in a direction moving away from the sun is a ridge. An indirectly illuminated to directly illuminated transition in a direction moving away from the sun is a valley. Then valleys and ridges exist on the borders between shadowed and directly illumnated areas. To find these areas one uses the binary shadow image. First, connected components are formed and small, noisy regions are eliminated. Figure 4.1 shows the connected component image after the elimination of small regions.

Next, the perimeters of these bright and shadowed regions are segmented into border segments according to their left regions, right regions, and orientations. A border segment is a maximally long sequence of connected pixels which are on the border between two given regions. Because the detection of ridges and valleys is highly dependent upon orientation and solar illumination comes from east in Figure 1.1, each border segment is further broken into several pieces according to orientation; all the east-west parts are separated from the north-south parts. The result is shown in Figure 4.2.

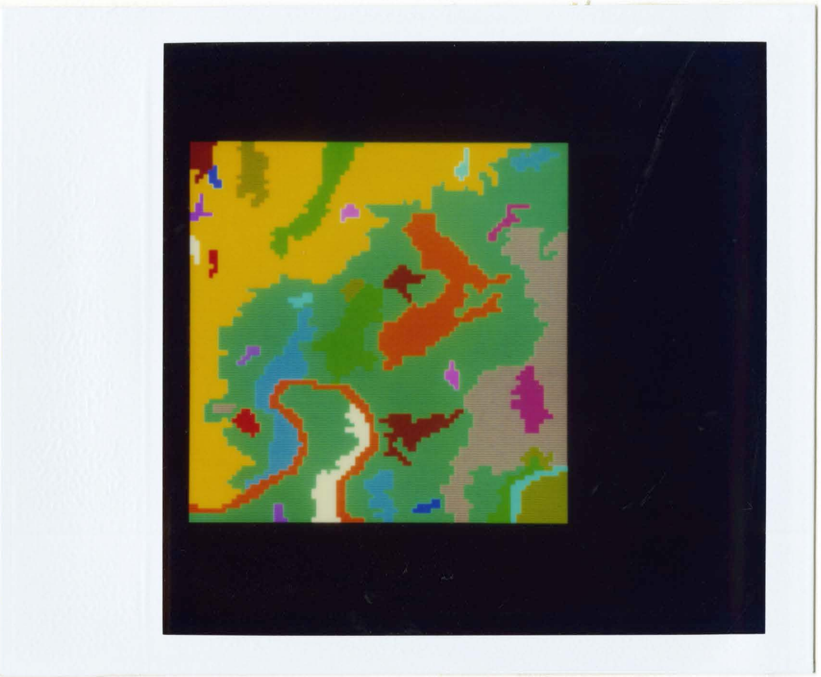


Figure 4.1 - Connected components from the shadow image.

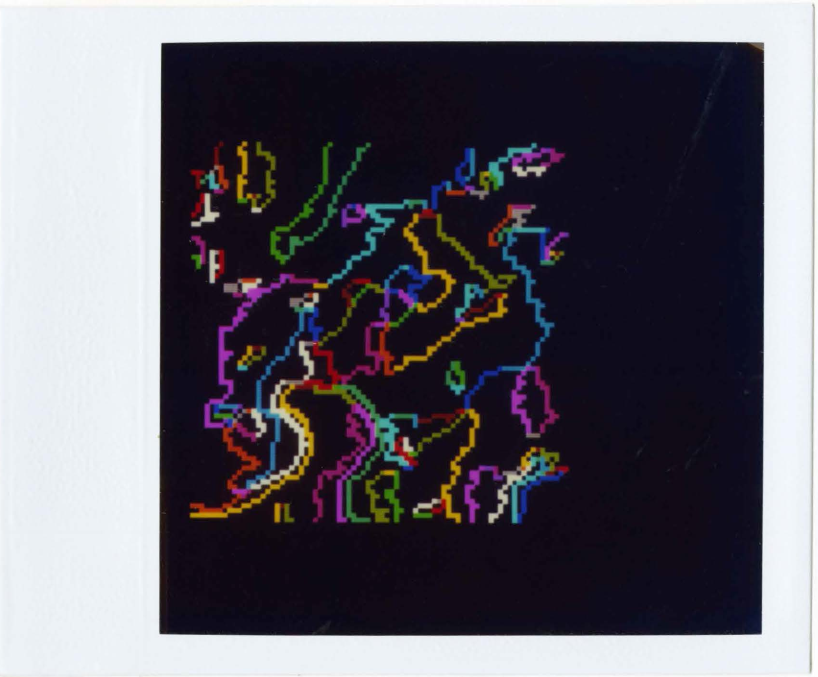


Figure 4.2 - Border Segments.

As the sun illumination is from the east in LANDSAT imagery, those border segments which are valley segments or ridge segments can be identified according to the brightness of the regions adjacent on the left and on the right. Because most of the trees in this area in April are unfoliated, the strongest region boundaries are shadow boundaries rather than reflectance boundaries, and the strongest boundaries are those at the extremes of steep slopes oriented normal to the sun direction. Because the sun illumination is predominantly east-west, a boundary that is dark on the left and bright on the right will correspond to a ridge, and the reverse will correspond to a valley.

East-west region boundaries are classified according to the labeling of neighboring north-south boundaries as well as their orientation relative to the east-west boundaries. As shown in Figure 4.3, each east-west boundary B_1 has a left intersecting north-south boundary B_2 and a right intersecting north-south boundary B_3. If the angle between B_1 and B_2 is smaller than the angle between B_1 and B_3, then the labeling of boundary B_2 is assigned to B_1; otherwise, the labeling of B_3 is assigned to B_1. The results of ridge-valley finding are shown in Figure 4.4.

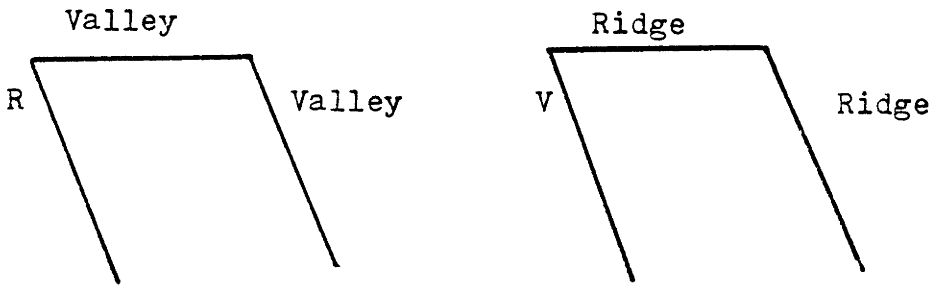


Figure 4.3 - Classifying east-west border segments.

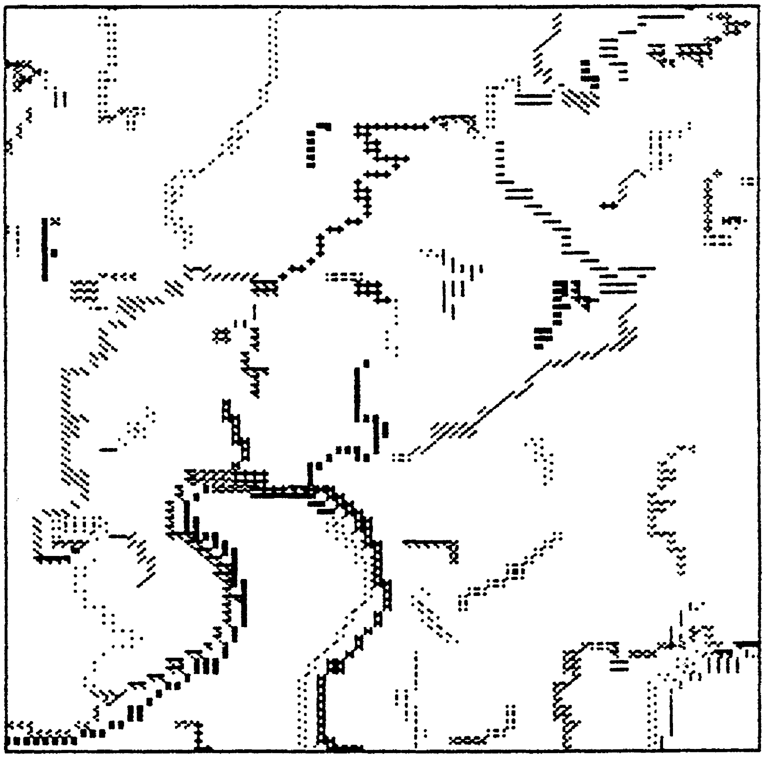


Figure 4.4.a - Valley map consisting of the border segments which are identified as valleys.

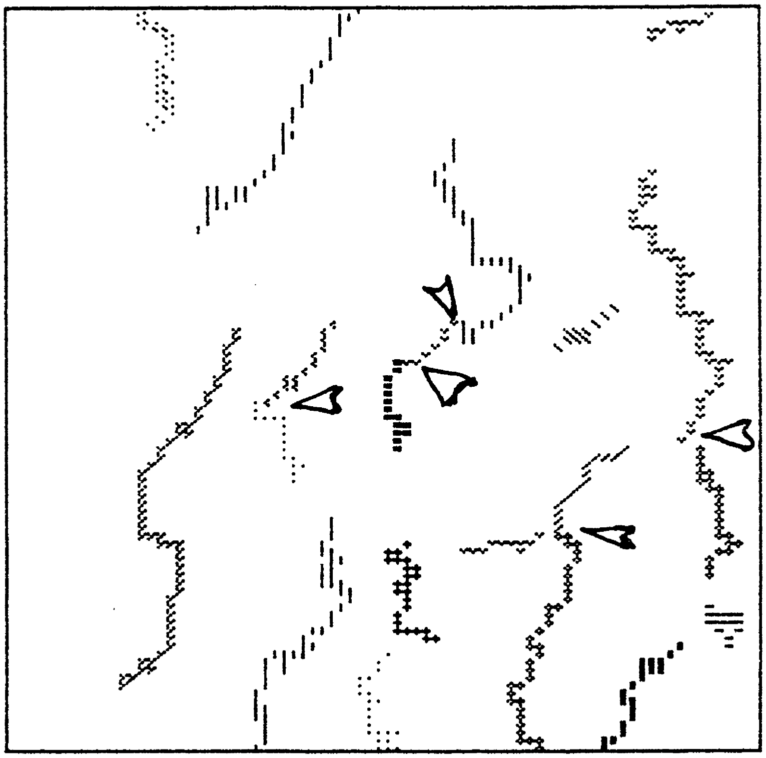
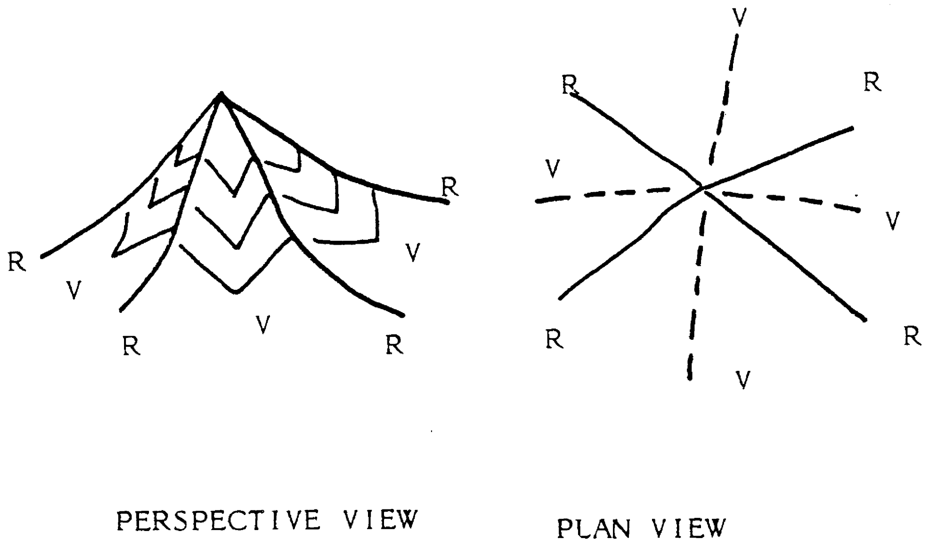


Figure 4.4.b - Ridge map consisting of the border segments which are identified as ridges.

4.1.1 Identification of Peak Junctions

In many cases, the topographic peaks are not like a perfect cone, but are surrounded by interleved concave (V-shaped) and convex (A-shaped) terrain features. The peak itself is formed by the junction of several ridges that radiate outward from the peak. (The idealized situation represented in Figure 4.5 shows four symmetrically oriented ridges; in our area, real peaks are often formed by junctions of two or three ridges.) Ridges of course are separated by valleys, so the higher tips of valley segments tend to point toward peaks. The ridge segments intersect to form a peak, whereas valley segments tend to point towards peaks, without actually joining. In this subsection, the criteria which can be used to identify peak junctions is discussed.

Because ridge segments are the major features of peaks, the constraint that the number of ridge segments at a junction is larger than the number of valley segments is made. For many situations, it seems reasonable to relate the heights of peaks to the lengths of ridges that form the peaks. For our class of topographic forms (for example), it is unlikely that very high peaks can be formed by the intersection of very short ridges. As a result, to exclude very low peaks and false peaks from consideration, a rather arbitrary constraint upon the definition of peaks is imposed. A peak junction is defined as a junction composed of exactly four border segments, with the number of its ridge segments larger than the number of valley segments, and the length of its longest ridge segment longer than 800 meters. The peaks thus located in Figure 1.1 are marked as triangles in Figure 4.4.b. The correspondence between this result and the topographical map is suprisingly good.



PERSPECTIVE VIEW

PLAN VIEW

Figure 4.5 - Idealized relationships between peaks, valleys, ridges.

4.2 RELATIVE ELEVATIONS OF RIDGE AND VALLEY SEGMENTS

The detection of the ridge and valley segments as discussed in the last section only assigns a ridge or valley label to them and does not assign relative elevations to them. This section describes how to estimate their relative elevations. Marr [1978] used a data representation called a primal sketch as a primitive representation of the information contained in a graytone image. Changes in graytone are represented by oriented edges. The image was reconstructed successfully from the primal sketch by scanning outwards from edges, assigning a constant brightness to points along the scan lines, and arresting the scan when it encounters another edge. The problem of reconstructing elevations from ridges and valleys is similar to Marr's problem if one replaces graytone by elevation. However, the following observation makes the difference.

The cross-sections of valleys are V-shaped, and the cross-sections of ridges are A-shaped. If one looks at topographic maps, the elevation contours of valleys such as those shown in Figure 4.6 can be frequently found. Thus, if one draws a line ab perpendicular to the valley Va, the elevations are increasing from point o to point a , and also from point o to point b. If the end point of a valley segment of smaller order is encountered during the growing, it is deduced that this end point is the lower end of this smaller valley segment. However, if a ridge point is encountered during the process, the increasing has to stop because the elevation starts to decrease. Based on this knowledge, an "elevation growing" model can be created.

Two differences exist between this model and Marr's reconstruction. First, instead of using constant values from edges to edges, local slopes are integrated from valleys to ridges. Second, there is a slope associated with the valley itself.

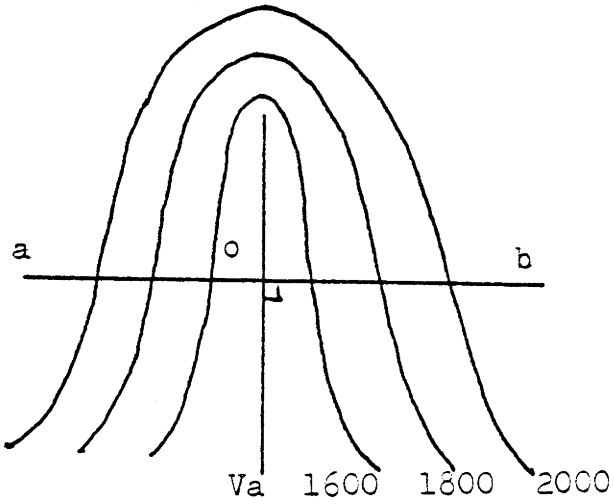


Figure 4.6 - The elevation pattern of valleys and its relation to elevation growing.

Neighboring pixels are related by local slopes. If a pixel (x,y) has elevation $E_l(x,y)$ and its local slopes at x, y directions are f_x, f_y, then the elevations of its neighbors can be calculated as follows.

$$
\begin{aligned}
E_l(x+1,y) &= E_l(x,y) + f_x\ G, && \text{for east pixel} \\
E_l(x-1,y) &= E_l(x,y) - f_x\ G, && \text{for west pixel} \\
E_l(x,y+1) &= E_l(x,y) + f_y\ G, && \text{for south pixel} \\
E_l(x,y-1) &= E_l(x,y) - F_y\ G, && \text{for north pixel} \qquad (4.1)
\end{aligned}
$$

where G is the ground distance between two pixels.

Initial estimates of f_x, f_y can be assigned for three different classes of pixels: pixels which are close to ridges, pixels which are close to valleys, and other pixels. A large local slope 0.4 m is assigned to pixels within 5 pixel distances to ridges so that one has steep hillsides. The slopes must be assigned in such a way that the elevations are decreasing from ridges in normal directions. Thus the east neighbor of a ridge pixel has f_x=-0.4m, and the west neighbor of a ridge pixel has f_x=0.4m. Similarly the south neighbor of a ridge pixel has f_y=-0.4m, and the north neighbor of a ridge pixel has f_y=0.4m.

A small local slope 0.2 m is assigned to the four-neighbors of valley pixels so that one has a wider valley bottom, and a medium local slope 0.1 m is assigned to the rest pixels. However, no directions are assigned to these pixels. It means

$$
\begin{aligned}
E_1(x+1,y) &= E_1(x,y) + |f_x(x,y)|\ G \\
E_1(x-1,y) &= E_1(x,y) + |f_x(x,y)|\ G \\
E_1(x,y+1) &= E_1(x,y) + |f_y(x,y)|\ G \\
E_1(x,y-1) &= E_1(x,y) + |f_y(x,y)|\ G \qquad (4.2)
\end{aligned}
$$

For a valley pixel, if it is within 10 pixel distances to a peak junction, then it is assigned a large local slope 0.4 m; otherwise, it is assigned a small local slope 0.02 m. No directions are assigned to these pixels. When Step 2.2 of the elevation growing algorithm is executed, there is a natural direction from the lower end of a valley segment to its higher end.

Using these local slopes and assuming the elevations of visible rivers are lowest in a small area, the elevation growing algorithm can be defined as follows:

1. Trace the border segments of visible rivers and give all

the pixels elevation E_o. E_o can be an arbitrary constant or a datum read from the map.

2. Repeat until all the pixels are elevation labeled

   2.1 Growing

   For a pixel having an elevation, calculate the elevations of its neighbors by Equation 4.1 if it has directional local slopes and by Equation 4.2 if it does not. However, elevations cannot be computed if

   a. An image boundary is encountered

   b. A ridge is encountered

   c. An elevation labeled pixel is encountered.

   2.2 Including new valley segments

   If any elevation-unlabeled valley segment is touched by an elevation-labeled pixel pe resulting from the growing, assign elevations to all the pixels of this segment. The end touching pe will have the same elevation as pe. Then, starting from this end, trace the whole segment and give every pixel linearly increasing elevation with some constant slope.

3. For a ridge pixel, take the maximum elevation value from its 4-neighbors' as its elevation.

Because the topographic shape of the hillsides from the valley to the ridge is not considered here, the initial relative elevations will be more accurate for the ridge or valley labeled pixels than the non-ridge and non-valley labeled pixels. Section 3 discusses a more realistic procedure for hillside elevation estimation using the ridge valley elevations calculated in this section, and Section 4 discusses how to use the topographic modulation image to improve the assignment of local slopes. The relative heights of valley segments created by the elevation growing model are indicated by arrows in Figure 4.7.a, and the ground truth is shown in Figure 4.7.b.

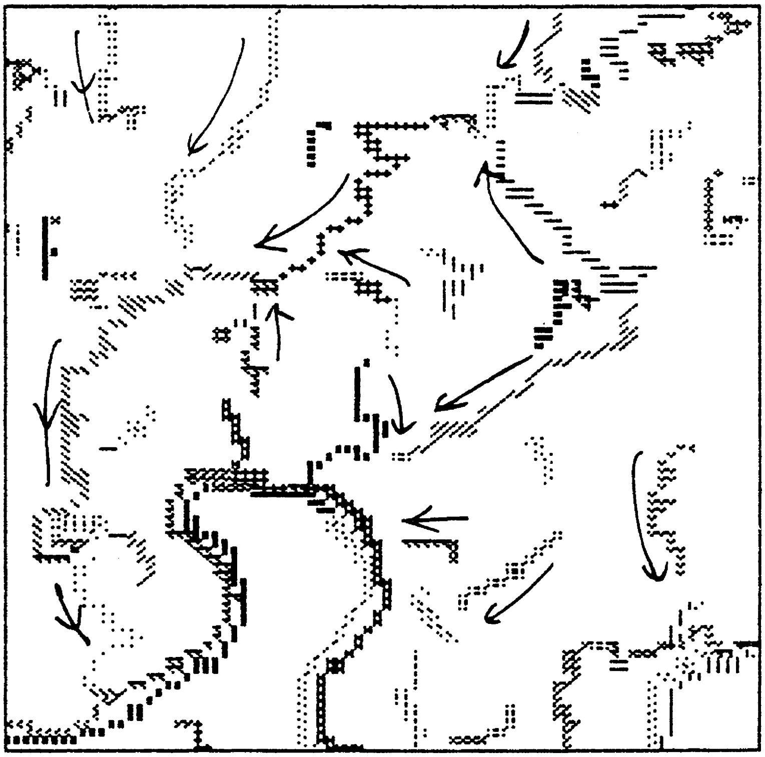


Figure 4.7.a - Relative elevations of valley segments. The arrow are from high ends to low ends.

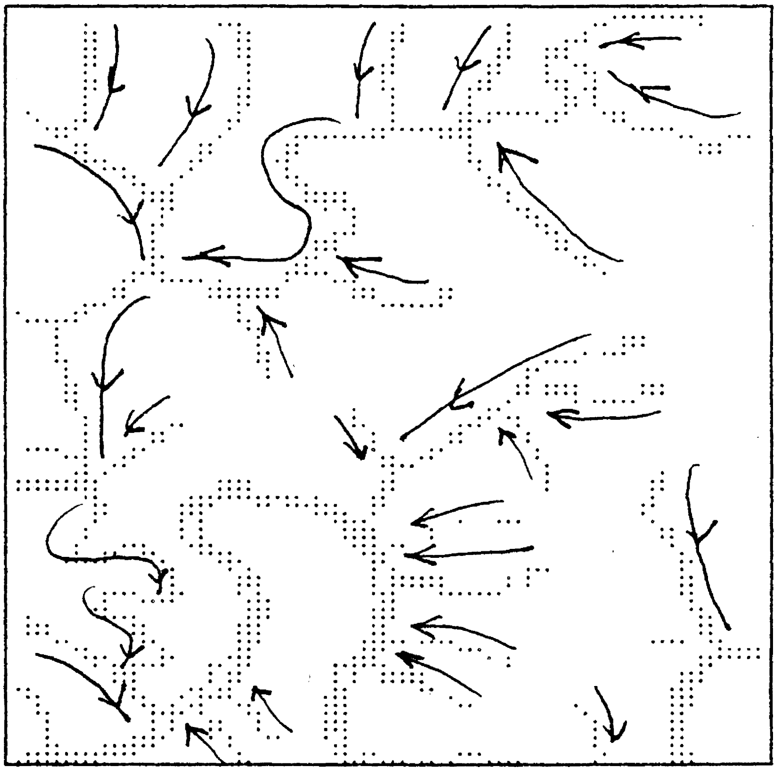


Figure 4.7.b - Stream map created from ground truth.

4.3 INTERPOLATING BETWEEN RIDGES AND VALLEYS

In the last section all pixels were assigned elevations, but because realistic shape of the hillsides from valleys to ridges were not taken into account, only the relative elevations of the ridges and valleys are held to be accurate. This section describes three interpolation procedures which permit more realistic elevation assignment to non-valley and non-ridge pixels. The three methods are Laplacian mask method, quadratic variation method, and distance method. They are described in this sequence in the following.

Using a 3X3 digital Laplacian mask as shown in Figure 4.8, the central pixel value x_o and the values of its 4-neighbors' x_1, x_2, x_4, and x_4 are related by

$$4x_o-x_1-x_2-x_3-x_4 = 0$$

If only one of the 4-neighbors, say the one with value x_4, is a ridge or valley pixel and has fixed elevation value c, then

$$4x_o-x_1-x_2-x_3 = c.$$

In general, the system of linear equations can be written as

$A\ x = b$ for the solution vector x.

For every pixel of unknown elevation, there is an entry in this vector x. A is a square matrix which contains 0, -1, or 4. b is the constant vector. For those pixels which do not have ridge or valley pixels as their 4-neighbors, there is a corresponding 0 in vector b. For those pixels which have ridge or valley pixels as their 4-neighbors, the sum of their elevations is put in vector b.

For a large linear system, an iterative method, Gauss-Seidel technique, is used to solve x. At each iteration, the pixel value is changed to make the sum of the mask values times the corresponding image values under them equal to zero. The iteration continues until the maximum change of values for all pixels is smaller than a given threshold. For the Laplacian surface, Neumann boundary conditions are enforced along the outside rows and columns of the elevation model image. That is, the outermost row or column is repeated so that the mask operator can be applied to the outside pixels.

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

Figure 4.8 - A digital Laplacian mask.

The second interpolating surface has the given boundary values and minimizes the quadratic variation of the resulting surface [Grimson, 1981]. The boundary conditions with which the surface must agree are elevation values along the zero-crossings. If the surface elevation function is E and subscripts denote partial differentiation, then the final surface E minimizes

$$\int\int (E_{xx}^2 + 2E_{xy}^2 + E_{yy}^2)\, dx\, dy$$

Since the surface function can be converted to a discrete grid format, the differential operators can be converted to difference operators, and the double integral can be converted to double summation, the solution of the above function can be formed by setting up a discrete corresponding set of linear equations

$$Q\, x = b.$$

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 (Figure 4.9) 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		

(1)

2		
-8		
8	-8	2

(2)

		2		
	4	-12	4	
	-8	20	-12	2

(3)

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

(4)

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

(5)

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

(6)

Figure 4.9 - Six masks for the quadratic variation method.

The third kind of interpolation surface can be created without using any mask. For each non-boundary pixel p, one can first find its distance d_v to the nearest valley pixel v and its distance d_r to the nearest ridge pixel r. Since the elevations of all ridge and valley pixels are known, one can name the elvation at pixel v as h_v and the elevation at pixel r as h_r. From d_v, d_r, h_v, and h_r, either a linear, cubic, or fifth order polynomial interpolation can be used to calculate the elevation of this non-boundary pixel. The configuration for linear fit is shown in Figure 4.10. If a cubic fit is used, the first order derivative is zero at ridge and valley pixels. If a fifth order fit is used, both the first and second order derivatives are zero at ridge and valley pixels. The resulting surface plots of these elevasections are shown in Figure 4.11. In the following three sections, only the result of the Laplacian surface will be used.

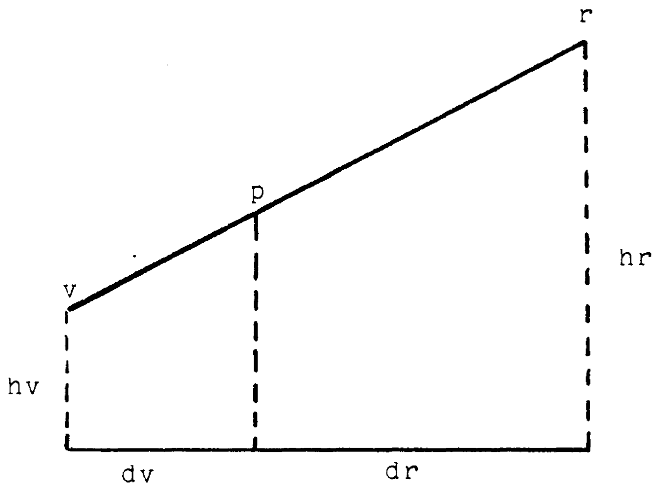


Figure 4.10 - Linear fit Interpolation.

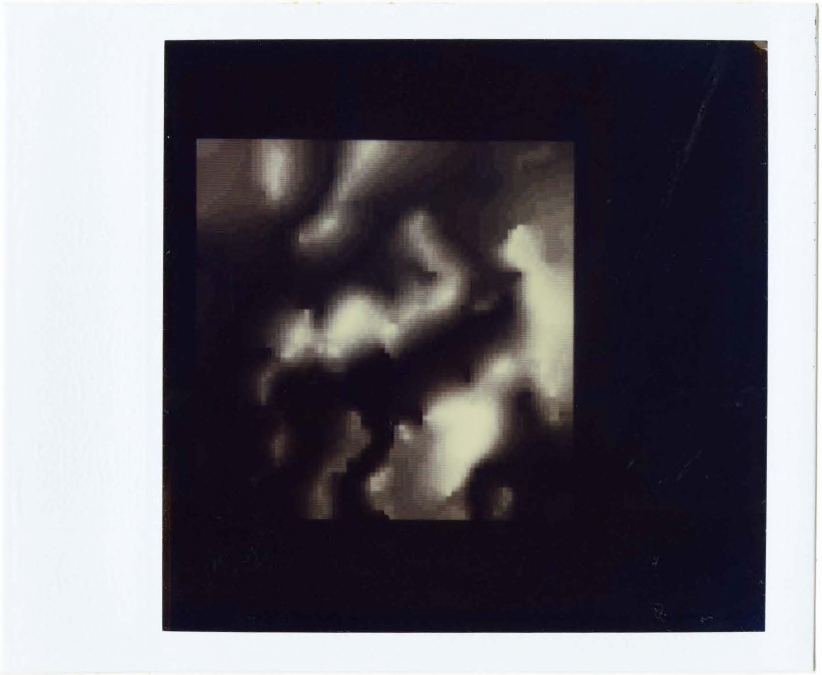


Figure 4.11.a - Elevation model by method 1, Laplacian mask.

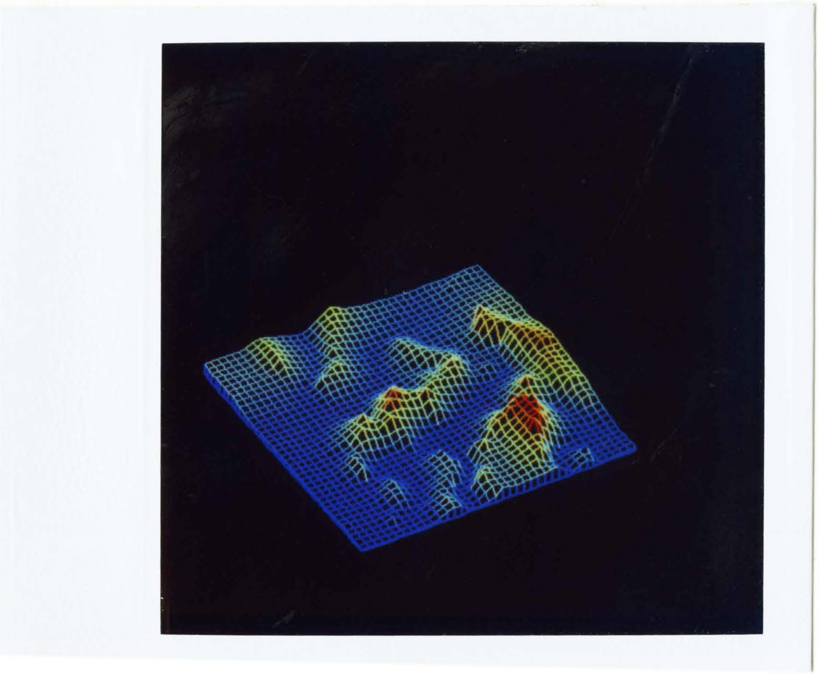


Figure 4.11.b - Surface plot of Figure 4.11.a.

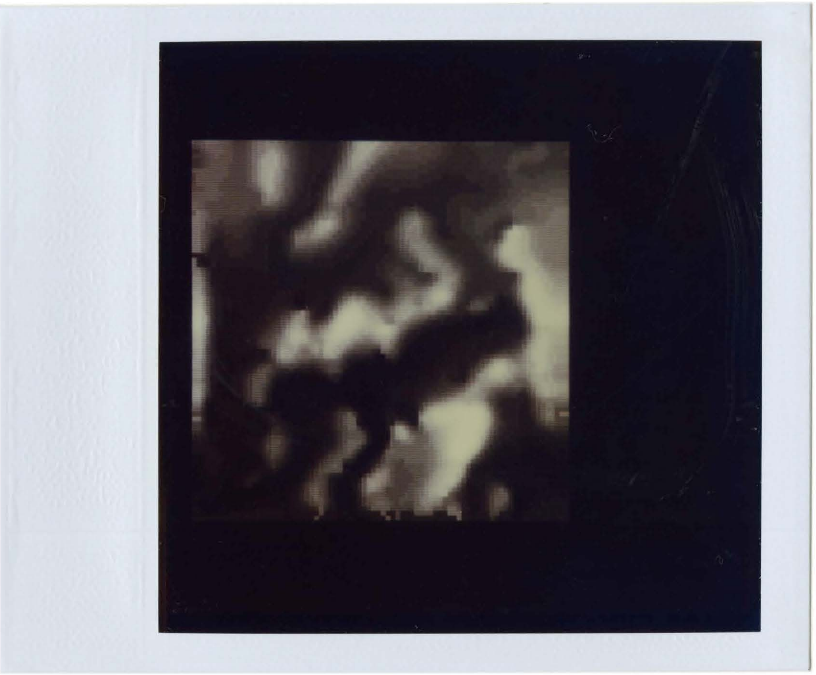


Figure 4.11.c - Elevation model by method 2, quadratic variation.

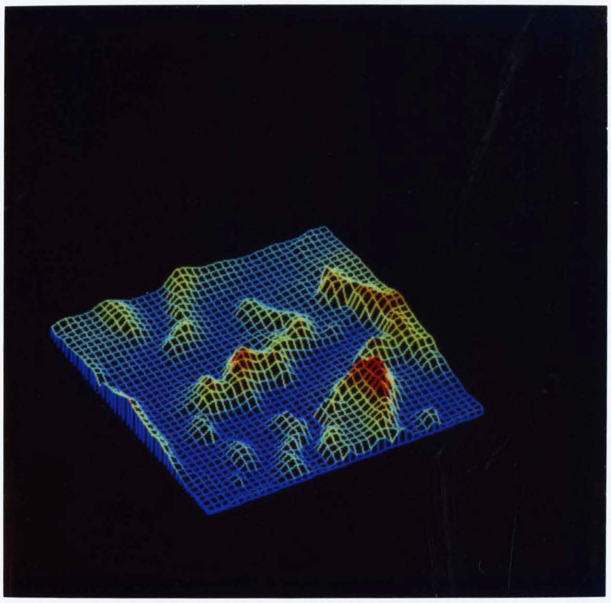


Figure 4.11.d - Surface plot of Figure 4.11.c.

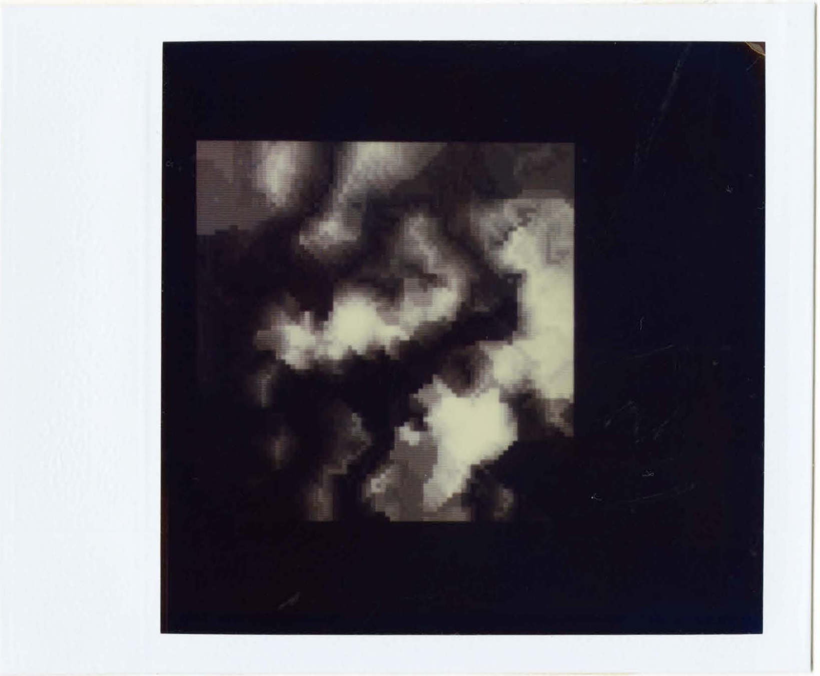


Figure 4.11.e - Elevation model by distance method, linear fit.

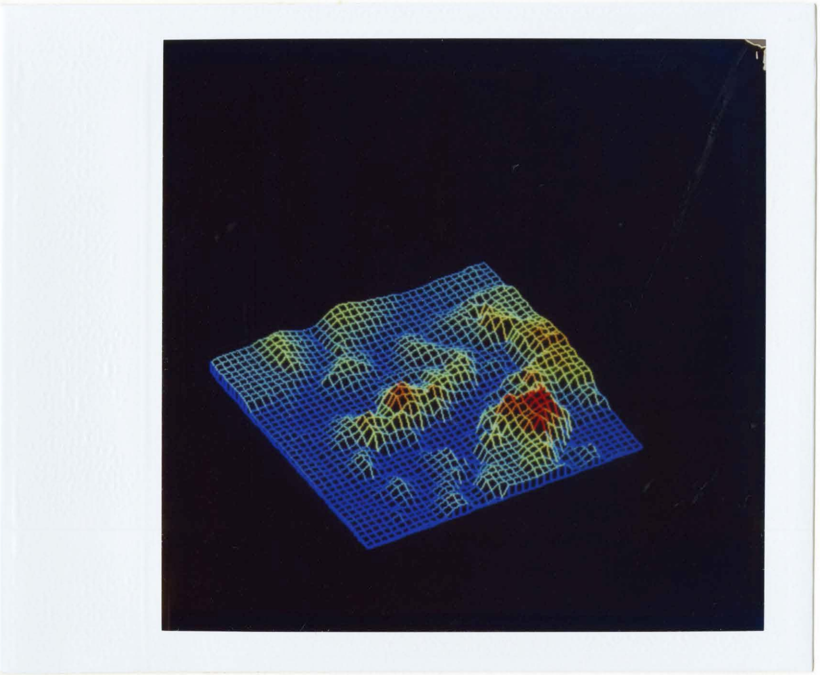


Figure 4.11.f - Surface plot of Figure 4.11.e.

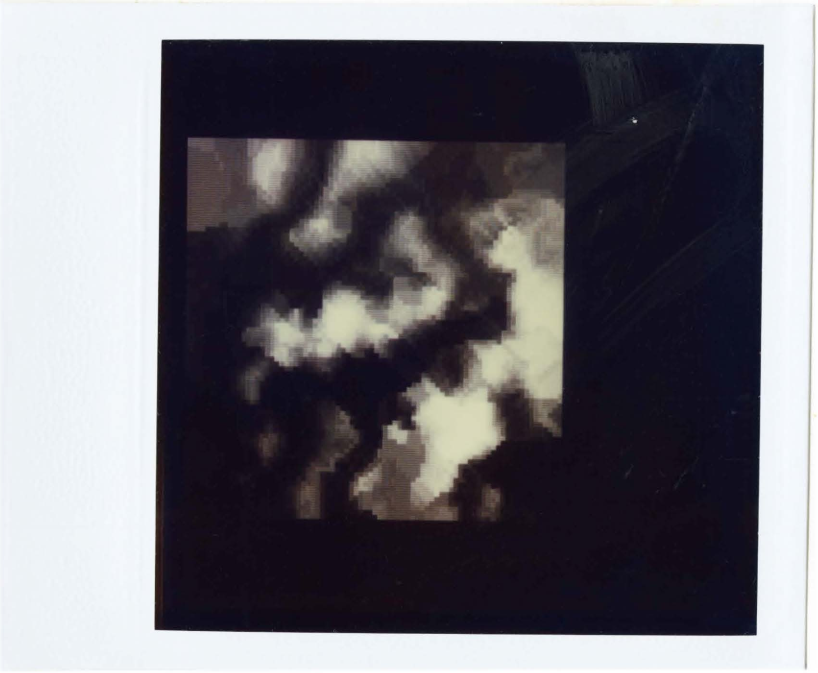


Figure 4.11.g - Elevation model by distance method, cubic fit.

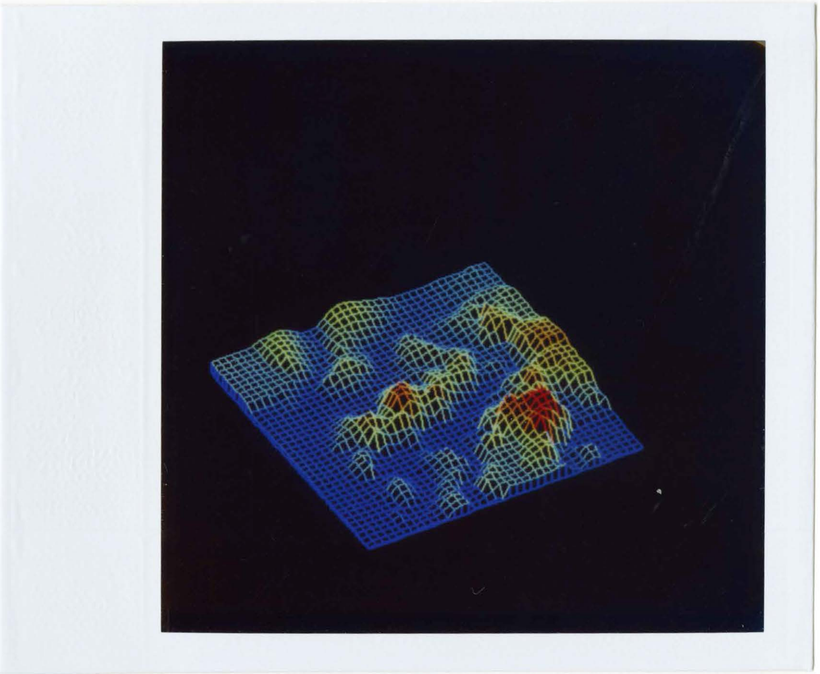


Figure 4.11.h - Surface plot of Figure 4.11.g.

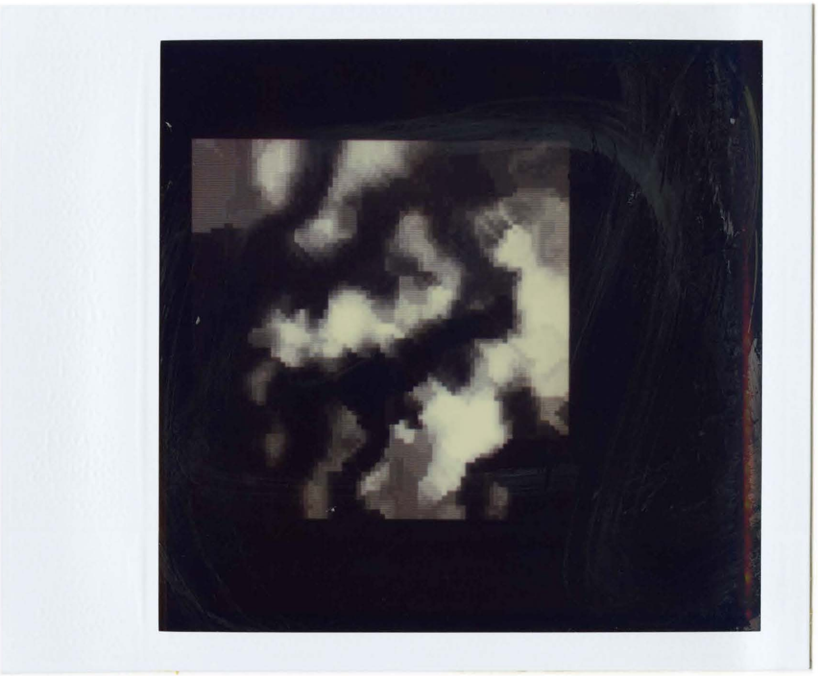


Figure 4.11.i - Elevation model by distance method, fifth order fit.

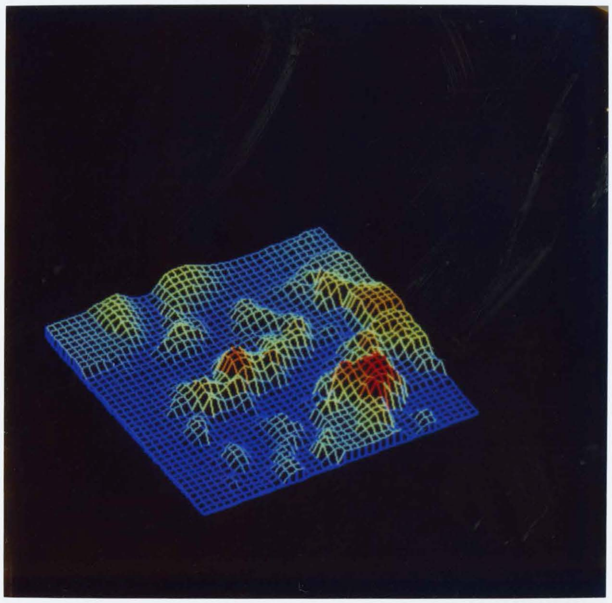


Figure 4.11.j - Surface plot of Figure 4.11.i.

4.4 USE OF TOPOGRAPHIC MODULATION IMAGE TO IMPROVE THE RESULT

As said in Section 1.2, one cannot obtain first order partial derivatives f_x, f_y of the elevation model f(x,y) from the topographic modulation image alone because the information of slope and aspect angle is mixed in the topographic modulation image. The relationship between f_x, f_y, and cosθ is contained in the following equation:

$$\cos\theta = \frac{f_x S_x + f_y S_y - S_z}{\mathrm{SQRT}\ (f_x f_x + f_y f_y + 1)} \qquad (4.3)$$

where $[S_x, S_y, S_z]$ is the known unit vector of sun illumination direction.

However, once a raw estimated elevation model f' is computed as described in the last section, the estimated partial derivatives f_x', f_y' can be calculated from f'. Using the known f_x', f_y', and the observed value cosθ' in the topographic modulation image, an a posteriori estimate f_x, f_y can be determined so that f_x and f_y maximize:

$$P(f_x, f_y \mid \cos\theta')$$

$$= \frac{P(\cos\theta' \mid f_x, f_y)\ P(f_x, f_y)}{\sqrt{\ } P(\cos\theta' \mid f_x, f_y)\ P(f_x, f_y)} \qquad (4.4)$$

Because one wants $\cos\theta$ in Equation 4.3 to be as close as possible to the observed value $\cos\theta'$, $P(\cos\theta' | f_x, f_y)$ can be written as

$$k_1 e^{k_2 \cos^2 (\theta - \theta')} \qquad (4.5)$$

Assuming the distributions of f_x, f_y are normal with means f_x', f_y' calculated from the estimated elevation model, then

$$P(f_x, f_y) = k_3 e^{-1/2((f_x - f_x')/\sigma)^2} e^{-1/2((f_y - f_y')/\sigma)^2} \qquad (4.6)$$

where σ is the standard deviation of the normal distribution.

Putting Equations 4.4, 4.5, and 4.6 together, one wants to find f_x and f_y which maximize

$$k e^{k_2 \cos^2 (\theta-\theta')} e^{-1/2((f_x-f_x')/\sigma)^2} e^{-1/2((f_y-f_y')/\sigma)^2} \qquad (4.7)$$

under condition (4.3) and $k = k_1 k_3$.

For smooth terrain surfaces, $f_x - f_x'$ and $f_y - f_y'$ are usually in the range from -2 m to 2 m. The maximizing f_x, f_y can be determined by exhaustive search over the range [-2m, 2m] x [-2m, 2m] of $(f_x - f_x', f_y - f_y')$ using an interval of 0.1

m. Given f_x-f_x' and f_y-f_y', Equation 4.6 can be calculated. Also since $f_x = (f_x-f_x') + f_x'$, and $f_y = (f_y-f_y') + f_y'$, cosθ in Equation 4.3 can be calculated. Thus Equations 4.5 and 4.7 can be calculated. By experiment, $k_2 = 1$ and $\sigma = 0.466$ give satisfactory results.

The maximizing f_x, f_y can be fed back to the procedure given in Section 4.2 to start another iteration of estimating the elevation model. This time all local slopes are directional, and Equation 4.1 is used for all pixels. Thus elevations may decrease for those pixels which are not close to ridges in the elevation growing algorithm. However, when the raw elevation estimates was first calculated using predetermined local slopes as in Section 4.2, elevation is strictly increasing from the big channels of rivers to ridges because Equation 4.2 was used for pixels which are not close to ridges and hillsides close to ridges are "protected" by those big directional local slopes. This processing of feedback can be repeated until the difference between the elevation averages of two consecutive iterations is small. The resulting elevation image as well as surface plot are shown in Figure 4.12.

Although one mainly wants to use cosθ to bias the calculation, the iterative method also has a feedback effect. In the initial elevation growing process, valleys are assigned small constant derivatives. However, if the end of a valley segment is located close to a ridge segment, then f_x' or f_y' is high. This high value is fed back to Equation 4.7, and the computed derivatives f_x and f_y will be much higher than the old small constant derivatives. As it can be seen in Figure 4.12, the reconstructed surface obtained after feedback is more consistent at locations where valleys are close to ridges.

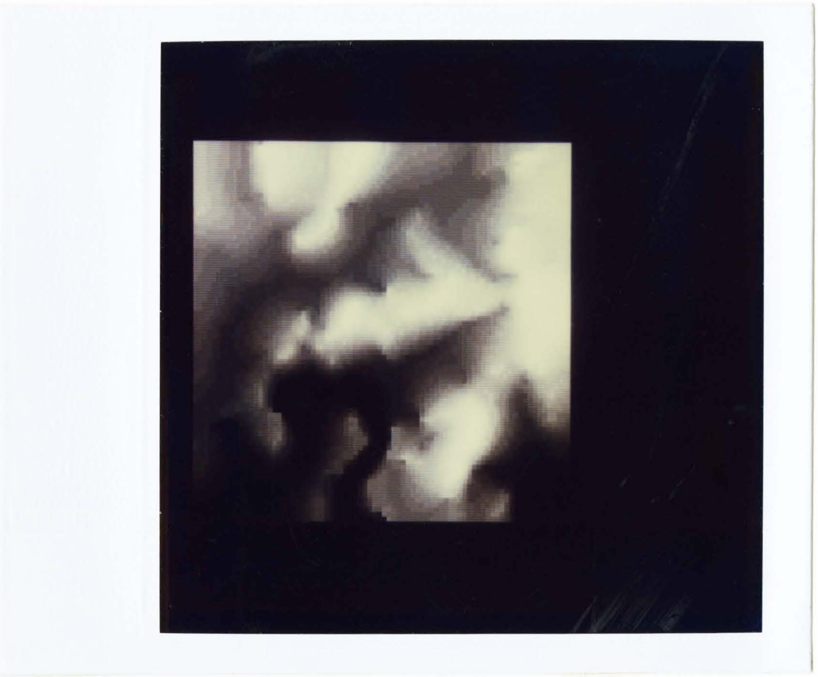


Figure 4.12.a - Optimal elevation image.

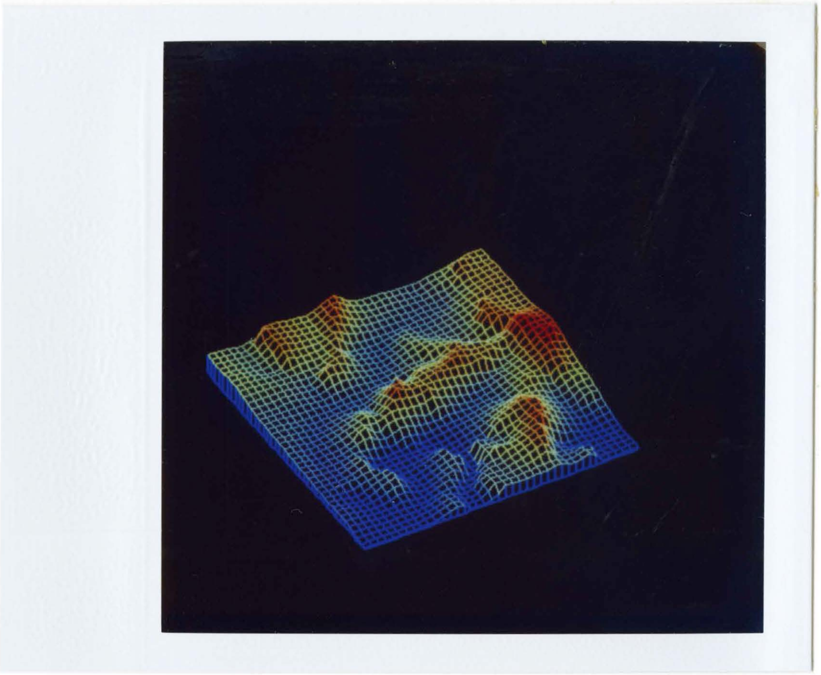


Figure 4.12.b - Surface plot of Figure 4.12.a.

4.5 EVALUATION OF ELEVATION MODEL AND ASSUMPTION 2

The image and surface plot of the elevation data read from Digital Terrain Tape [NCIC, 1980] for the area of Figure 1.1 are shown in Figure 4.13. The comparison of the minimum, maximum, and mean elevation values between the reconstructed elevation model and tape data for three test areas is listed in Table 4.1.

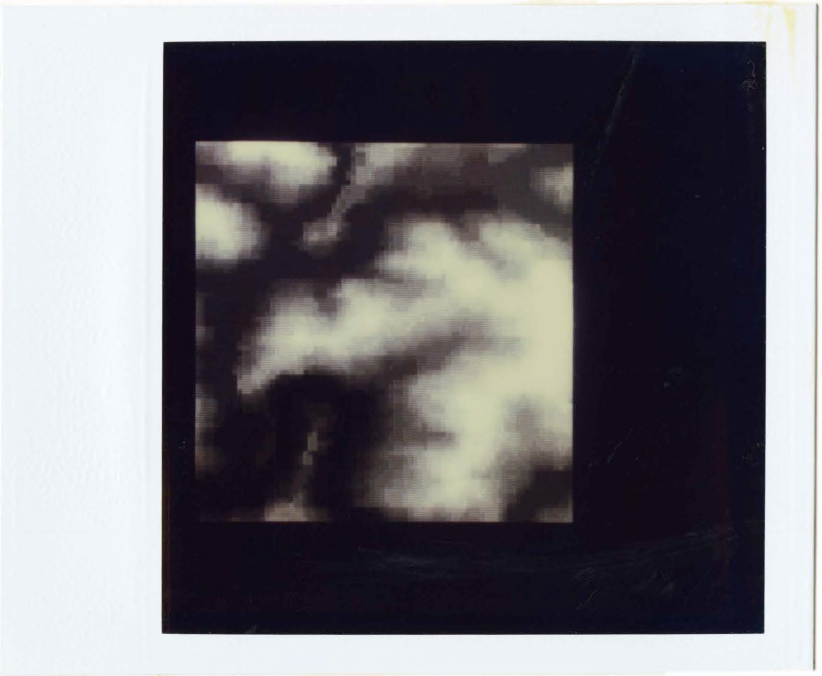


Figure 4.13.a - Elevation data form digital terrain tape.

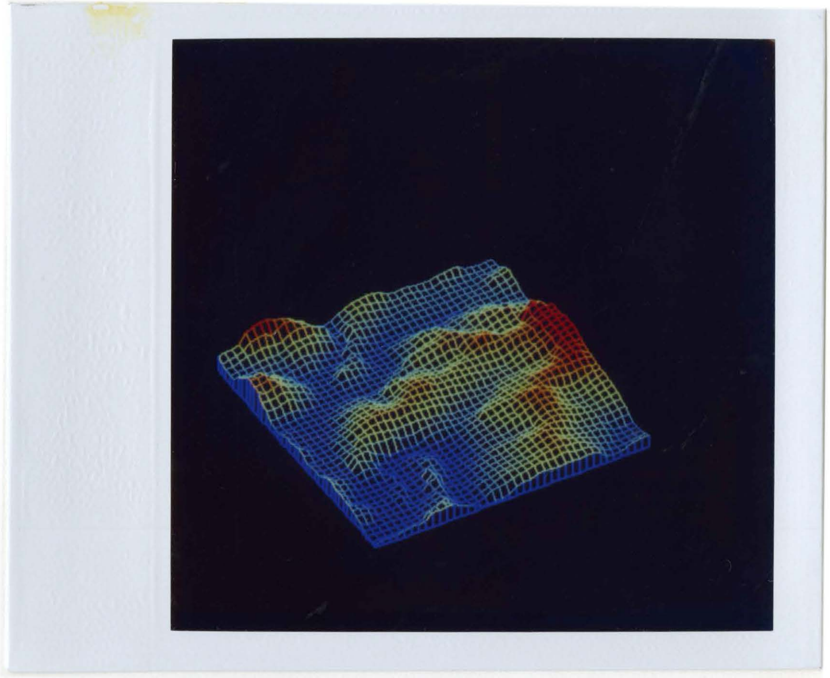


Figure 4.13.b - Surface plot of Figure 4.13.a.

Table 4.1 - Overall comparison of reconstructed elevation model and tape data. Unit is foot.

Test Imagery 1

	Low	High	Mean
Tape Data	950	2123	1350
Reconstruction	950	2033	1320

Test Imagery 2

	Low	High	Mean
Tape Data	675	2200	1150
Reconstruction	675	1899	938

Test Imagery 3

	Low	High	Mean
Tape Data	800	2100	1410
Reconstruction	800	2167	1280

Before a pixel-by-pixel comparison can be done, the estimated elevation values need to be scaled so that they are in the same range as the tape data. This is done by creating a mapping table which maps each value in the first data file into a value in the second data file and then computing a scaled data file from the first one by using this table. The mapping table can be created either by Least Square Fit or simply locating all the pixels in the first file for a particular pixel value and computing the mean values of the corresponding pixels in file two. [Elliott, 1982]

The latter method is used here. Let E_e be the file of estimated elevation values and E_t be the tape data. For any elevation e, a set of pixels having this value in E_e can be defined as

$$S_e(e) = \{(u,v) \mid E_e(u,v) = e\}.$$

Thus a scaled file E_s can be created as

$$E_s(x,y) = \frac{\sum\limits_{(u,v)\varepsilon S_e(E_e(x,y))} E_t(u,v)}{\# \; S_e(E_e(x,y))}$$

After the reconstructed elevation values are scaled, two kinds of data can be calculated from the scaled elevation estimates and tape data for three test areas. The first is the mean of the absolute difference image E_d. Since $E_d(x,y) = |E_s(x,y) - E_t(x,y)|$, the mean is

$$\sum_{(u,v)} E_d(u,v) \;/\; n$$

Where n is the total number of pixels. The second is root-mean-square (RMS) error:

$$\left(\sum_{(u,v)} E_d(u,v)^2 \;/\; n\right)^{1/2}$$

These results are listed in Table 4.2. The absolute difference image is shown in Figure 4.14. Also shown in Table 4.2 are the comparisons of local slope, NORM(f_x,f_y), and azimuth angle, ARCTAN(f_y/f_x), for first order partial derivatives f_x, f_y.

The results in Table 4.2 are largely degraded due to the imperfection of registrating the LANDSAT data with the map data. It is especially obvious for azimuth angles. If valleys or ridges in two images do not match, then a pixel in image one may face east while the corresponding pixel in image two is facing west. Thus the difference of azimuth angle is close to 180 degrees. As can be seen in Figure 4.15, most big errors (bright pixels) occur at pixels close to valleys or ridges. The reconstruction for test area 1 shows the best result because the terrain changes more frequently in this area and more ridge, valley segments are detected which makes the elevation growing algorithm work better.

In Assumption 2 of Section 2.7, the spatial average of cosθ is assumed to be the same for all material clusters. After the elevation model is reconstructed, all X_c values for three test areas are calculated and listed in Table 4.3. From this, Assumption 2 is a pretty reasonable one.

Table 4.2 - Pixel-by-pixel comparison of tape data and reconstructed elevation model

Difference of Elevation (Unit is foot)

	Mean of abs. difference	RMS error
Test Imagery 1	115	143.76
Test Imagery 2	225	284.4
Test Imagery 3	150	194.19

Difference of slope (Unit is foot)

	Mean of abs. difference	RMS error
Test Imagery 1	0.117	0.154
Test Imagery 2	0.211	0.27
Test Imagery 3	0.151	0.19

Difference of Azimuth Angle (Unit is degree)

	Mean of abs. difference	RMS error
Test Imagery 1	47.5	67.0
Test Imagery 2	46.9	66.7
Test Imagery 3	53.8	74.0

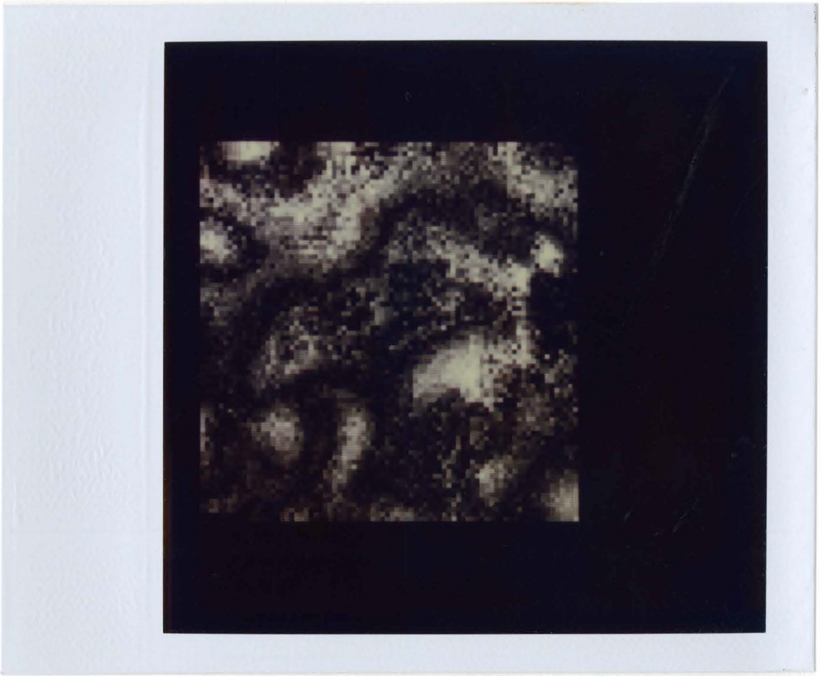


Figure 4.14 - Absolute elevation difference image

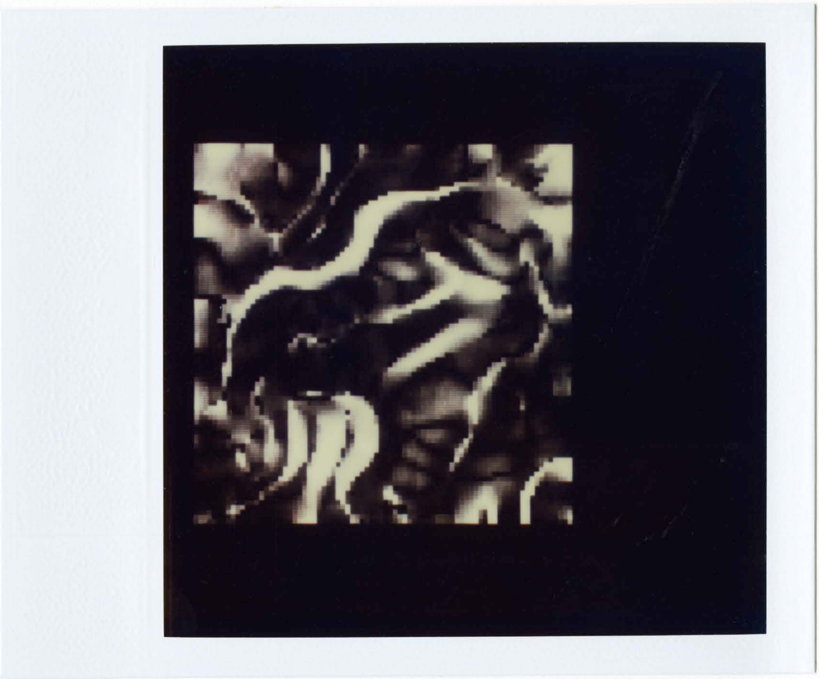


Figure 4.15 - Absolute difference image of azimuth angles

Table 4.3 - Listing of X_c values.

Test Imagery 1

Cluster	1	2	3	4	5
X_c	0.7193	0.3735	0.7495	0.7734	0.7042

Test Imagery 2

Cluster	1	2	3	4
X_c	0.7297	0.734	0.6702	0.7000

Test Imagery 3

Cluster	1	2	3
X_c	0.6677	0.6343	0.7600

4.6 RECONSTRUCTION OF THE LANDSAT IMAGERY

From Equation 2.6, the LANDSAT imagery can be reconstructed by the following steps:

1. Illuminate the reconstructed elevation model of Figure 4.12 by an artificial sun at specified azimuth and elevation angles.
2. Multiply the image of Step 1 by the reflectance image of Figure 2.10.
3. Add diffuse light image of Figure 2.6 to the image of Step 2.

The resulting imagery is shown in Figure 4.16.

For evaluation, the original and reconstructed imagery are scaled to the same graytone range [1, 64] and the difference image between them is computed. The means and variances of the difference imagery for three test areas are listed in Table 4.4. The result is reasonably good to show the correctness of the illumination model and the elevation growing model.

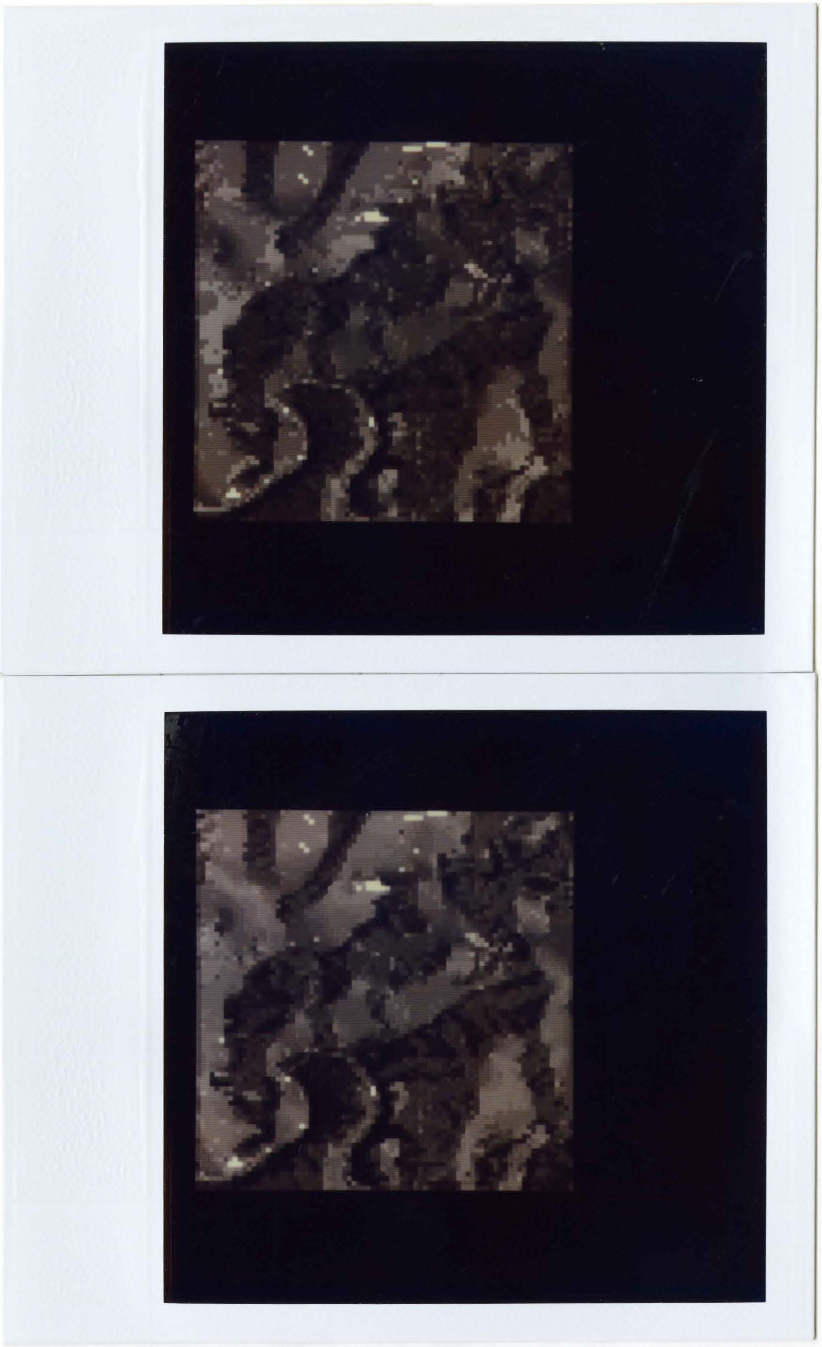


Figure 4 16 - Reconstructed LANDSAT imagery.

(Top: band 4; bottom: band 5)

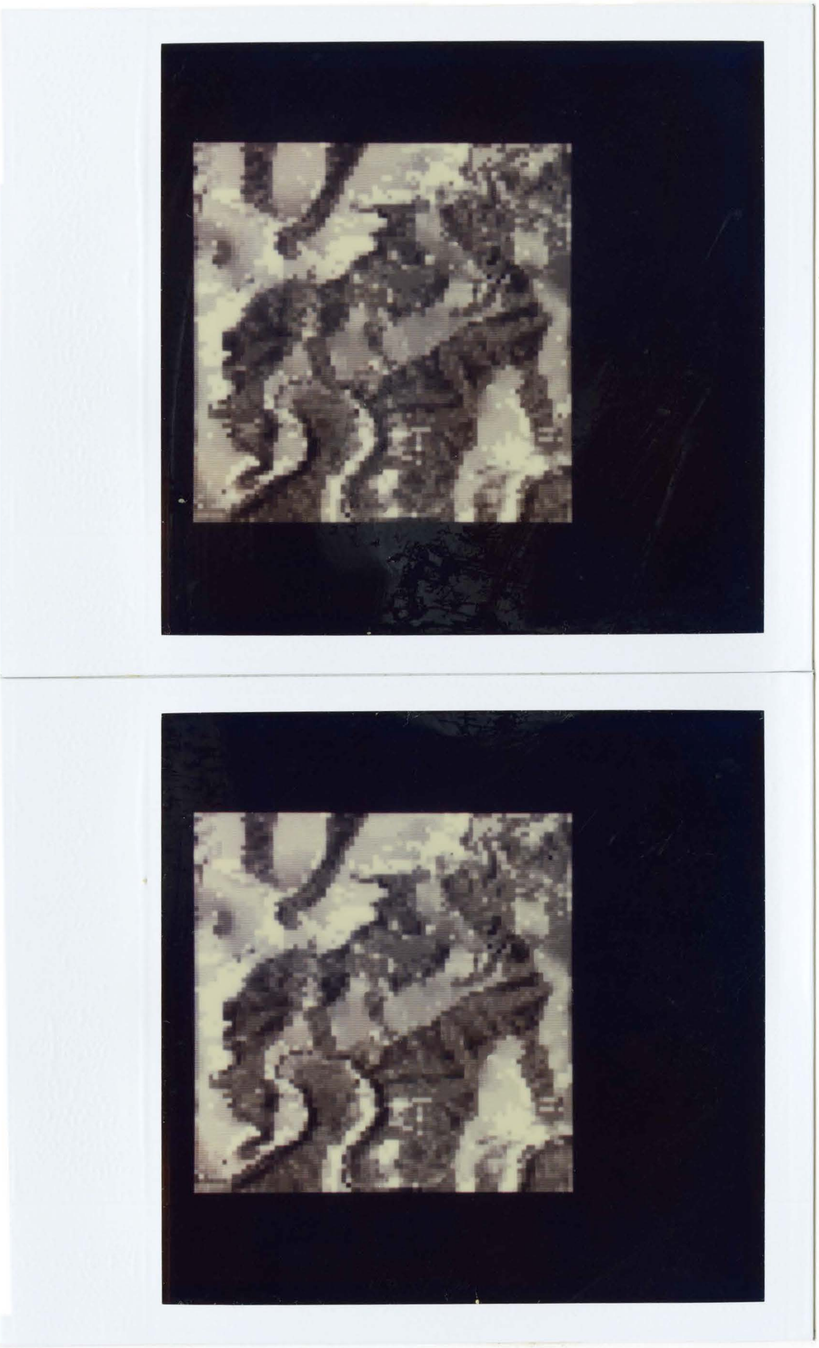


Figure 4.16 - Continued. (Top: band 6; bottom: band 7)

Table 4.4 - Mean and variance of the difference imagery between original and reconstructed LANDSAT imagery.

<u>Test Imagery 1</u>

	mean	variance
Band 4	4.1	18.9
Band 5	3.62	16.2
Band 6	5.13	32.2
Band 7	8.27	43.3

<u>Test Imagery 2</u>

	mean	variance
Band 4	5.33	31.1
Band 5	7.23	50.7
Band 6	5.52	35.4
Band 7	5.72	27.5

<u>Test Imagery 3</u>

	mean	variance
Band 4	4.9	29.2
Band 5	5.86	24.0
Band 6	7.53	46.6
Band 7	5.36	32.2

Chapter V

FLOW DIRECTIONS OF VISIBLE RIVERS

5.1 CONSTRAINTS AT JUNCTIONS

Visible rivers were detected in Chapter 3, and relative elevations of valley segments which are close to visible rivers were detected in Chapter 4. It remains to find the flow directions of visible rivers which are assigned constant elevations in the elevation growing model. This problem of assigning labels of {upstream, downstream} to the visible river segments is very much like the Waltz [1975] problem of labeling edges of polyhedra objects, and one needs to find constraints applicable to streams.

It is believed that when several stream segments join at a junction there are constraints based on orientation and length patterns. The most obvious and important one is the configuration in Figure 5.1. When a smaller stream s_2 flows into a larger stream which is composed of two segments s_1 and s_3 because of the junction, very often the angle between s_2 and s_1 is less than 90 degrees. General rules about flow directions at junctions are given in Table 5.1 where $A(s_1,s_2)$ indicates the angle between segments s_1 and s_2 and $L(s)$ indicates the length of a segment s.

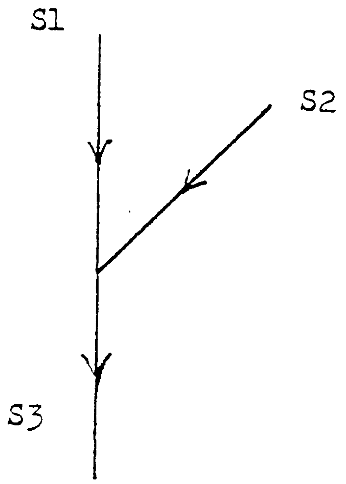


Figure 5.1 - One pattern of a stream junction.

Table 5.1 - Rules of flow directions at junctions for all possible patterns of segment lengths and orientations.

Pattern Number	$A(s_1,s_3)$	$A(s_1,s_2)$	$A(s_2,s_3)$	Upstream	Downstream
	=180	=90	=90		
1	$L(s_3)>=max(L(s_1),L(s_2))$			s_1 and s_2,	s_3
2	$L(s_2)<=min(L(s_1),L(s_3))$			s_2,	s_1 or s_3
3	$L(s_3)<=min(L(s_1),L(s_2))$			s_2 and s_3,	s_1
4	$L(s_1)=L(s_2)=L(s_3)$			s_2,	s_1 or s_3
5	=180	<90	>90	s_1 and s_2,	s_3
6	=180	>90	<90	s_2 and s_3,	s_1
7	<180	>=90	>=90	s_1 and s_3,	s_2
	>180	>=90	<=90		
8	$L(s_1)=L(s_2)=L(s_3)$			s_2 and s_3,	s_1
9	$L(s_2)<=min(L(s_1),L(s_3))$			?,	?
10	$L(s_1)<=min(L(s_2),L(s_3))$			s_2 and s_3,	s_1
11	$L(s_3)<=min(L(s_1),L(s_2))$			?,	?
	=120	=120	=120		
12	$L(s_1)=L(s_2)=L(s_3)$			?,	?
13	$L(s_2)<=min(L(s_1),L(s_3))$			s_1 and s_3,	s_2
14	$L(s_1)<=min(L(s_2),L(s_3))$			s_1 and s_3,	s_2
15	$L(s_3)<=min(L(s_1),L(s_2))$			s_1 and s_2,	s_3

Due to the low resolution of LANDSAT imagery, few visible stream segments can be detected. Thus few junctions can be detected. One can increase the number of junctions by including junctions where valley segments intersect the visible stream segments. The necessary assumption is that there is water flowing through these valley segments.

Now two kinds of junctions are of interest. Junctions of the first kind are vertices at which three stream segments meet. The set of such junctions is called J_3. Junctions of the second kind are vertices at which two stream segments and one valley segment meet. The set of such junctions is called J_2. Also S is the set of all stream segments and V is the set of all valley segments.

Let $J = J_3 \cup J_2$, and let X be the set of junction patterns in Table 5.1, and L = {upstream, downstream}. Then one can define a: J -> X as the function that assigns junction patterns to junctions. As an example, for the junction j in Figure 5.1, a(j)=5 because $A(s_1,s_3)=180$, $A(s_1,s_2)<90$ and $A(s_2,s_3)>90$. For each pattern in Table 5.1, the flow directions of segments s_1,s_2,s_3 can be put in a triple which is an element in LxLxL. Thus the mapping from pattern numbers to flow directions can be defined as b: X -> LxLxL. For the junction j in Figure 5.1, b(a(j))=(upstream, upstream, downstream).

If three stream segments meet at a junction, two constraint relations can be formally stated on the basis of Table 5.1 as follows. One concerns all stream segments that constrain each other because they meet at this junction; the other concerns all segment-label pairs for the stream segments that meet at the junction and the labels that are possible for that type of junction. For all junctions of the same junction pattern x, one can put their segments in a set T_x. Adding labels to the elements in T_x, another set R_x can be defined. They are

$T_x=\{\{s_1,s_2,s_3\}|s_1,s_2,s_3\varepsilon S,$ and s_1,s_2,s_3 meet at a junction $j\varepsilon\ J_3$ of type $x\}$

$R_x=\{\{(s_1,l_1),(s_2,l_2),(s_3,l_3)\}|\{s_1,s_2,s_3\}\varepsilon T_x$ and $(l_1,l_2,l_3)\ \varepsilon\ b(x)\}$

For a pattern number x in Table 5.1, T_x contains all the sets of three stream segments that meet at a junction of this pattern, and R_x contains all the sets of three segment-label pairs for the sets in T_x.

If two stream segments and one valley segment meet at a junction, two similar constraint relations can be stated as follows. For each x in X,

$T_x' = \{\{s_1, s_2\} | s_1, s_2 \varepsilon S$, and there exists $s_3 \varepsilon V$ such that s_1, s_2, s_3 meet at a junction $j \varepsilon J_2$ of type $x\}$

$R_x' = \{\{(s_1, l_1), (s_2, l_2)\} | \{s_1, s_2\} \varepsilon T_x'$, and $(l_1, l_2, l_3) \varepsilon b(x)\}$.

For each pattern number x in Table 5.1, T_x' contains all the sets of two stream segments which meet at a junction togeth-er with a third valley segment. R_x' contains all the sets of two segment-label pairs for sets in T_x' and the labels defined by the function b.

Finally, the union of these sets for all the junctions can be defined as

$$T = (\underset{j \varepsilon J_3}{\breve{U}} T_{a(j)}) U (\underset{j \varepsilon J_2}{U} T'_{a(j)})$$

and $R = (\underset{j \varepsilon J_3}{U} R_{a(j)}) U (\underset{j \varepsilon J_2}{U} R'_{a(j)})$

which means T consists of all triples or pairs of stream segments that constrain each other at junctions, and R is the corresponding segment-label constraint relation.

Thus the labeling problem of assigning {upstream, down-stream} to all stream segments can be described by a compatibility model (S, L, T, R), which is a particular instance of the general consistent labeling problem [Haralick and

Shapiro, 1979 and 1980]. Because there are many spatial inference problems which are instances of the consistent labeling problem, the form of the general consistent labeling problem as given by Ullmann, Haralick and Shapiro [1982] is described in the next section.

5.2 CONSISTENT LABELING

Let U be a set of objects called units, and L be a set of possible labels for those units. Let $T \subseteq \{f | f \subseteq U\}$ be the collection of those subsets of units from U that mutually constrain one another. That is, if $f = \{u_1, u_2, . . . , u_k\}$ is an element of T, then not all possible labelings of $u_1,...,u_k$ are legal labelings. Thus there is at least one label assignment $l_1, l_2, ... , l_k$ so that u_1 having label l_1, u_2 having label l_2, . . . , u_k having label l_k is a forbidden labeling. T is called the unit constraint set. Finally, let $R \subset \{g| g \subset UxL$, g single-valued, and $Dom(g) \varepsilon T\}$ be the set of unit-label mappings in which constrained subsets of units are mapped to their allowable subsets of labels. If $g = \{(u_1,l_1),(u_2,l_2), . . . , (u_k,l_k)\}$ is an element of R, then $u_1, u_2, ... , u_k$ are distinct units, $\{u_1,u_2, ... , u_k\}$ is an element of T meaning $u_1, u_2, ... , u_k$ mutually constrain one another, and u_1 having label l_1, u_2 having label l_2, . . . , and u_k having label l_k are all simultaneously allowed.

In the consistent labeling problem, one is looking for functions that assign a label in L to each unit in U and satisfy the constraints imposed by T and R. That is, a consistent labeling is one which when restricted to any unit constraint subset in T yields a mapping in R. In order to state this more precisely, the restriction of a mapping is first defined. Let h:U-->L be a function that maps each unit in U to a label in L. Let $f \subseteq U$ be a subset of the units. The restriction h|f (read h restricted by f) is defined by $h|f = \{(u,l) \varepsilon h \mid u \varepsilon f\}$. With this notation, a consistent labeling is defined as follows.

A function h:U --> L is a consistent labeling if and only if for every f ε T, h|f is an element of R.

An example is given below. Suppose the inputs to the problem are as follows:

U = {1, 2, 3, 4, 5}

L = {a, b, c}

T = { {1},	unary constraint
{1, 2},	binary constraints
{2, 5},	
{1, 3, 4}}	ternary constraint

R = { {(1,a)},{(1,b)}, unary constraint

{(1,a),(2,a)},

{(1,a),(2,b)},

{(1,b),(2,b)}, binary constraints

{(2,a),(5,a)},

{(2,b),(5,c)},

{(1,a),(3,a),(4,c)}, ternary constraints

{(1,b),(3,a),(4,a)}}

Then h = {(1,a) (2,a) (3,a) (4,c) (5,a)} is a consistent labeling. To see this note that h | {1} = {(1,a)},

h | {1,2} = {(1,a),(2,a)}, h | {2,5} = {(2,a),(5,a)}, and

h | {1,3,4} = {(1,a),(3,a),(4,c)} are all elements of R.

If having l_1, ..., l_k applied to u_1, ..., u_k when {(u_1, l_1), ..., (u_k, l_k)} is not in R is allowed with a penalty, the process is called inexact consistent labeling [Shapiro and Haralick, 1981]. In order to include these mappings, an error function E_w is defined.

Let E_w : G -> [0,1] be the error weighting function where $G \subseteq \{g | g \subseteq U\times L$, g single-valued, Dom(g)εT}.

$E_w(\{(u_1,l_1), \ldots, (u_k,l_k)\})$ is the error which occurs when labels l_1, ..., l_k are applied to u_1, ..., u_k. If $\{(u_1,l_1),\ldots,(u_k,l_k)\}$ is in R, E_w is zero; otherwise, E_w is a constant e_c and usually is defined as the reciprocal of the square of the size of U. The mapping h : U -> L is an inexact consistent labeling if for all f in T, the sum of $E_W(h|f)$ is within some upper bound, usually 1.

5.3 SPATIAL REASONING MODEL AND FLOW DIRECTIONS OF VISIBLE RIVERS

In the last section, the very general consistent labeling model was introduced and the unit-label pairs in the elements of R were just assumed to be there. However, if one goes back to the flow direction problem and looks at Table 5.1, it is clear that one cannot talk about unit-label pairs without looking at the property values, such as angles and lengths, of these units. In the following, based on the very general consistent labeling model, a spatial reasoning model is defined to explicitly include these properties. However, these properties are only related to the creation of elements in the set R; the basic tree searching technique is just the same for both the general consistent labeling model and the spatial reasoning model. Thus, after the spatial reasoning model is shown applicable to the flow direction problem, the tree searching strategies designed by Shapiro and Haralick [1981] will be introduced.

In spatial reasoning problems, many spectral and geometrical properties can be computed for the locally detected units. Some frequently used properties are average gray level, size, and shape descriptors. For each unit, a list of property values can be computed. Considering all the units, these lists form an array which can be named P. Thus for a unit u, P[u] gives the list of property values for u. For stream junctions, the line length of one segment and the clockwise angle from one segment to the next one can be detected so that P[s]=(length, angle) for a unit s. For example, in Figure 5.1, $P[s_1]=(10,45)$, $P[s_2]=(10,135)$, and $P[s_3]=(15,180)$.

For each junction pattern in Table 5.1, the angles must be within certain ranges. With respect to pattern number 5, $P[s_1]$ must be in the property range ([0,89], $[1,u_b]$), $P[s_2]$ must be in the range ([91,179], $[1,u_b]$), and $P[s_3]$ must be in the range ([180,180], $[1,u_b]$) for some upper bound u_b on the line length.

However, simply specifying a range for each unit is not enough. Sometimes one needs to compare the property values for different units. One example for the stream junction is the pattern "$L(s_3)>=max(L(s_1),L(s_2))$". To handle this type of constraint, a relation $r(P[u_1],...,P[u_k])$ must be defined on the property lists of the related units.

Now the spatial reasoning model is a 6-tuple (U, P, L, T, R, E_w). U, L, T, E_w have the same meanings as before; however, the elements in R now have the form $\{(u_1, p_1, l_1), ..., (u_k, p_k, l_k); r(P[u_1],P[u_2],...,P[u_k])\}$ where p_i is the list of the required ranges of property value for all the properties in P for unit u_i, i = 1 to k. If the property values of u_i are within the ranges specified by pi for i = 1 to k, $\{u_1, \ldots, u_k\}$ is contained in T, and relation r is satisfied, then it is legal to assign label l_1 to u_1, ..., l_k to u_k at the same time.

The spatial reasoning model (U, P, L, T, R, E_w) can be applied to deduce the flow directions of visible rivers. U contains the units of visible rivers. P contains all the properties detectable from the stream segments. The most important properties are the length of a segment and the orientation of the segment at both ends because they are used in Table 5.1. The algorithm for computing the orientation of a segment at one end is given in Appendix A. L is {Upstream = 1, Downstream = 2}. T contains the junction relations. R contains the relations of legal flow directions defined in Table 5.1. For $\{u_1,\ldots,u_k\}$ in T, if $\{(u_1,l_1),\ldots,(u_k,l_k)\}$ is in R, the error function $E_w(\{(u_1,l_1),\ldots,(u_k,l_k)\})$ is defined to be zero; otherwise, it is e_c, the reciprocal of the square of the total number of stream segments.

A number of computational techniques for determing solutions to the inexact matching problem have been given in the literature. The following is a formulation of solutions to this problem in the frame work of [Shapiro and Haralick, 1981] who showed experimentally that a process called forward checking was most efficient. For reference, their detailed algorithm is given in Appendix B.

The inexact consistent labeling problem can be solved by a brute force backtracking tree search. Before the bottom of the tree is reached, only parts of the units are labeled, and thus only the error incurred against all units which have already been assigned labels can be calculated. Such a labeling is called a partial labeling; the labeld units are called past units, and the set of all past units is called U_p. Similarly, the units which have not been labeled are called future units, and the set of all future units is called U_f. Also let T_1 be the set of all sets composed of units which have already been assigned labels, i.e. $T_1 = \{\{u_1,u_2,\ldots,u_k\} | u_1,u_2,\ldots u_k \varepsilon U_p$ and $\{u_1,u_2,\ldots,u_k\} \varepsilon T\}$. Thus the error for past units, e_p, incurred in backtracking is

$$e_p(U_p,h)= \sum_{\substack{\{u_1,\ldots,u_k\} \\ \varepsilon T_1}} E_W(\{(u_1,h(u_1)),\ldots,(u_k,h(u_k))\}) \quad (5.1)$$

for a partial labeling h. If the error sum exceeds an error bound e_b, the tree search must either try the next label for the current unit or if there is no next label, it must backtrack.

As a simple example, let U={1,2,3,4,5}, L={a,b,c}, T={{1},{1,2},{1,4}}, R={{(1,a)}, {(1,a),(2,b)}, {(1,b),(2,b)}, {(1,b),(4,b)}}, error constant e_c=1/6, and error bound e_b=0.2. In the tree search, label a is assigned to 1 first. Thus U_p={1}, T_1={{1}}, $e_p(U_p,h)$=0 because {(1,a)} is in R.

Next, label a is assigned to 2 because backtracking is depth-first. Now U_p={1,2}, T_1={{1},{1,2}}, $e_p(U_p,h)$=1/6 because h={(1,a),(2,a)} is not in R. Since e_p is smaller than e_b=0.2, one can continue and assign label a to unit 3. As {1,3}, {2,3} are not in T, T_1 is not changed and $e_p(U_p,h)$ is not changed.

Next, label a is assigned to unit 4 which will cause U_p={1,2,3,4}, T_1={{1}, {1,2}, {1, 4}}, $e_p(U_p,h)$=1/6+1/6=1/3 because {(1,a),(4,a)} is not in R. At this point, e_p=1/3 which is larger than e_b=0.2, and one cannot continue with unit 5. Instead, next label b is assigned to unit 4. This is the trace for backtracking.

A technique called backtracking with forward checking can improve the speed of tree search. For the previous example, 3 units were assigned labels before a cutoff happened. Actually, by looking at sets T,R and doing some calculations described below, decision about cutoff can be made even after the first unit is assigned a label. Thus the searching is more efficient. This technique is similar to the branch and bound technique except that a fixed bound value is used.

The speed of tree search can be improved if one also considers the minimum error that the current labeling must incur against future units which have not been assigned labels. Thus the set in T containing only one future unit is of interest; for a future unit u and label l, define

$T(u,i;U_p)=\{\{u_1,\ldots,u_k\}$ in $T \mid u_i=u$ and $n\neg=i$ implies u_n in $U_p\}$.

For example, when $U_p=\{1\}$, $h=\{(1,a)\}$, $T(2,2;\{1\})=\{\{1,2\}\}$, $T(4,2;\{1\})=\{\{1,4\}\}$.

Using labeling h on all units except u and assigning label l to u, the error (e_{pf}, error for past and future units) is

$$e_{pf}(u,l;U_p,h)$$

$$= \sum_{i=1}^{k} \sum_{\substack{\{u_1,\ldots,u_k\} \\ \text{in } T(u,i;U_p)}} E_w(\{(u_1,h(u_1)),\ldots,(u_{i-1},h(u_{i-1})),(u,l),(u_{i+1},h(u_{i+1})),\ldots,(u_k,h(u_k))\}) \qquad (5.2)$$

In the continuing example, if $u=2$, $l=a$, then $e_{pf}(2,a;\{1\},\{(1,a)\}) = E_w(\{(1,a),(2,a)\}) = 1/6$ because $\{(1,a),(2,a))$ is not in R.

To be complete, one should also consider the smallest error of the units in the nodes with higher level numbers in the tree created by backtracking or the units other than u in U_f. It is

$$\sum_{\substack{v \varepsilon U_f \\ v \neg= u}} \min_{m \varepsilon L} e_{pf}(v,m;U_p,h) \qquad (5.3)$$

For the continuing example, when $v=3$,

$$\min_{m \varepsilon L} e_{pf}(3,m;\{1\},\{(1,a)\}) = 0$$

because $T(3,i;\{1\})$ is always empty.

When $v=4$,

$e_{pf}(4,a;\{1\},\{(1,a)\}) = 1/6$

$e_{pf}(4,b;\{1\},\{(1,a)\}) = 1/6$

$e_{pf}(4,c;\{1\},\{(1,a)\}) = 1/6$, and

$$\min_{m \varepsilon L} e_{pf}(4,m;\{1\},\{(1,a)\}) = 1/6.$$

When v=5,

$$\min_{m \varepsilon L} e_{pf}(5,m;\{1\},\{(1,a)\}) = 0$$

for the same reason as when v=3. Now the sum in Equation 5.3 is 0+1/6+0 = 1/6.

For current labeling h, if the summation of Equations 5.1,5.2 and 5.3 exceeds an error bound for any label l for the current unit u, then one needs either try the next label for the current unit or backtrack. This is called backtracking tree search with forward checking. From the above calculations, for u=2, l=a, 0+1/6+1/6 = 1/3 > 0.2, and one needs to try the next label b for current unit 2. Thus, it is clear that only one node is generated in the searching tree as opposed to three nodes in the case of backtracking.

Implementing the model (U,P,L,T,R,E_w) has two steps. Step one, check all the units by using the rules such as Table 5.1 or the lists of property ranges $p_1,p_2,...,p_k$ and relation $r(P[u_1],P[u_2],...,P[u_k])$, find out all possible labeling pairs $\{(u_1,l_1),(u_2,l_2),...,(u_k,l_k)\}$. These are put into some data files.

Step two, use the forward checking algorithm to find out the global inexact consistent labeling. In addition to the procedures in Appendix B, only two procedures are needed for each application. The first procedure, which is called by procedure UPDATE in Appendix B, reads the data file created in Step 1 and checks the unit-label constraint relations. The second procedure is a driver which sets up all the data structures and calls the procedures in Appendix B.

The flow directions of the streams for the three test areas are deduced and are labeled as in Figure 5.2. These flow directions are correct with respect to the ground truth. For test areas 1, 2, and 3, number of units are 11, 31, and 19 respectively. 0, 1, and 2 junctions are found to have different labelings from those predicted by Table 5.1.

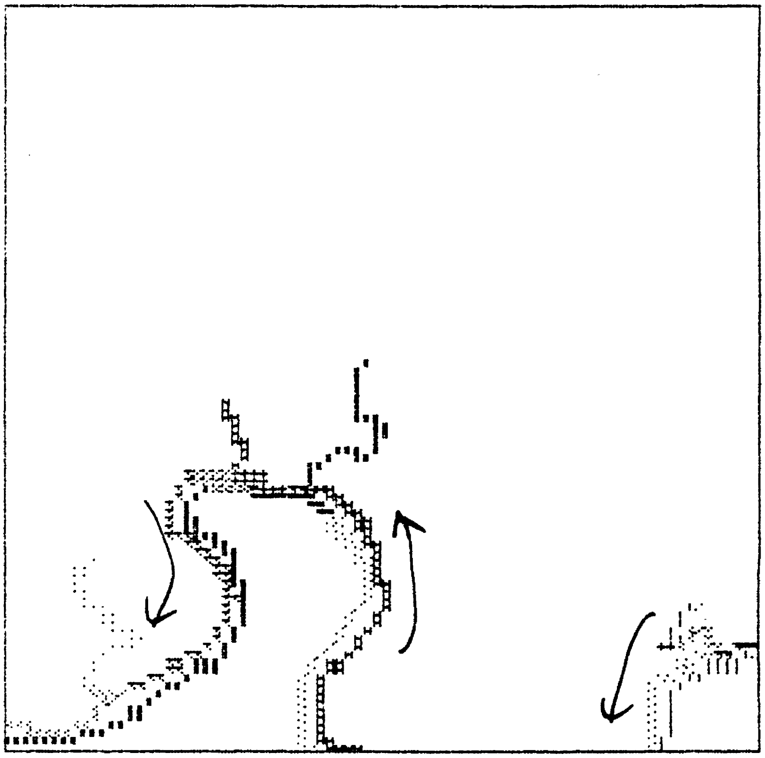


Figure 5.2.a - Drainage network for Figure 1.1.

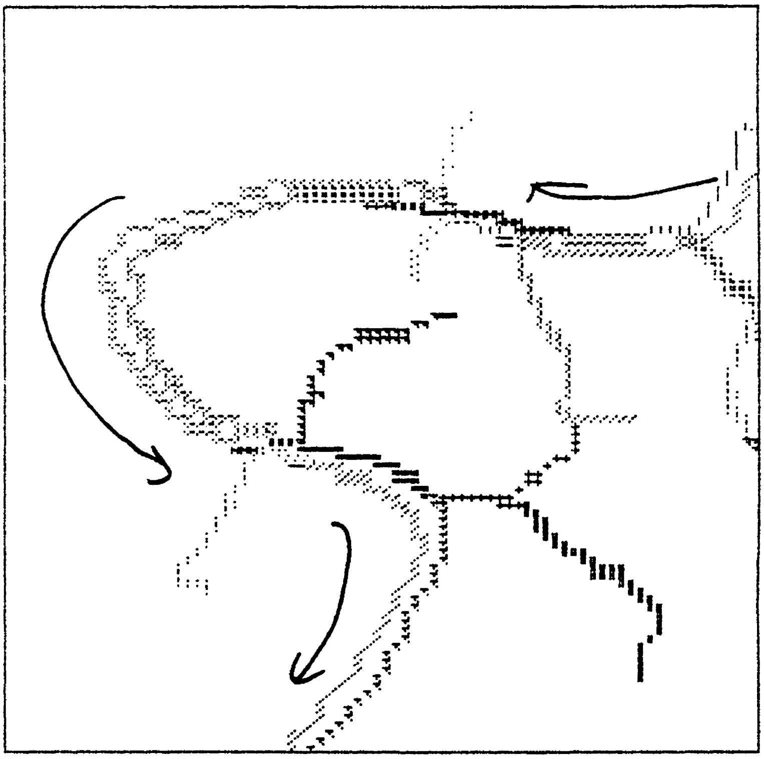


Figure 5.2.b - Drainage network for Figure 1.2.a.

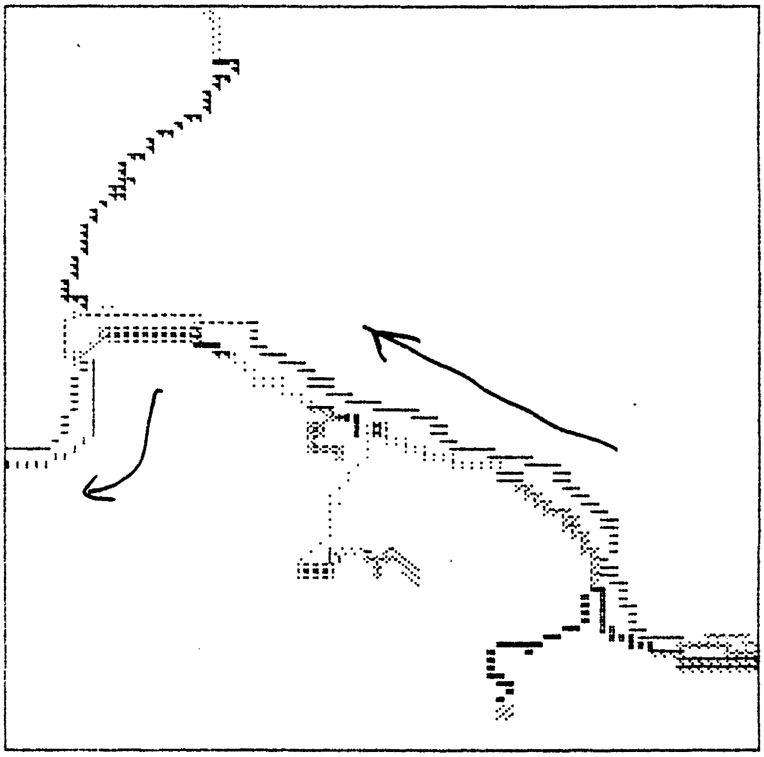


Figure 5.2.c - Drainage network for Figure 1.2.b.

Chapter VI

CONCLUSION

As said in Section 1.1, three problems are addressed in this dissertation: definition of an illumination model, understanding the elevation model, and deduction of drainage networks. Theories and tests were worked out, and reasonable results were obtained. Of these three problems, understanding the elevation model is found to be the most difficult one.

In Chapter 2, illumination model was defined and a series of steps was used to extract the diffuse light imagery, the reflectance imagery, and the topographic modulation image. The success of the process depends largely on clustering. ISODATA was modified in such a way that advanced information about characteristics of clusters can be used during clustering. More studies can be done for this technique. In Chapter 3, simple computation was done to obtain spectral information.

In Chapter 4, ridges, valleys, and peaks were found equivalently important in understanding elevation model. The elevation growing process was found efficient in reconstructing the elevation model. Ridges and valleys were used just as skeletons are used in 2-D shape analysis. The iterative method was used to improve the reconstructed elevation model. This method was found significant and can be applied to the reconstruction of graytone images too.

Finally in Chapter 5, constraints at junctions and the consistent labeling technique were used successfully in finding the flow directions of rivers. Even though the numbers of units in the test are not large, the mathematical model of the spatial reasoning model is precise and useful for many applications. The reasons that the numbers of units in the test are not large are:

First, the resolution of the LANDSAT imagery is low so that few stream segments can be detected. This will not be a problem as more advanced satellites are sent into space.

Second, even if the resolution is improved and more stream segments can be detected, most flow directions can be decided immediately by the fact that smaller streams always flow into large streams. Thus the only streams with unknown flow directions would be the largest streams in the imagery and the total number of units suitable for the consistent labeling problem is very limited.

Instead, more challenging tasks for the consistent labeling process can be found in the field of pattern recognition such as classifying ground objects in urban areas. Hundreds of elements in the unit constraint set T and unit-label constraint set R can be defined because of the diversity of ground objects.

The techniques of this dissertation work best for areas having clear shadows. If water areas cannot be found to help identify the lowest locations, the elevation understanding problem will be more complicated. Although only LANDSAT imagery was used in this dissertation, the principles and techniques can be applied to other kinds of imagery as long as they are multi-spectral. One good application area is the recognition of indoor objects because illumination conditions and spectral ranges can be controlled easily.

BIBLIOGRAPHY

T.T. Alfoldi and J.C. Munday, Jr., "Water Quality Analysis by Digital Chromaticity Mapping of LANDSAT Data", Canadian J. of Remote Sensing, 4 (2): pp. 108-126, 1978.

R. Bajcsy and M. Tavakoli, "Image Filtering - a Context Dependent Process", IEEE Trans. on Circuits and Systems, May 1975.

H.G. Barrow and J.M. Tenenbaum, "Recovering Intrinsic Scene Characteristics from Images", in Computer Vision Systems, A. Hanson and E. Riseman (Eds.), Academic Press, 1978.

J. Bryant, "On the Clustering of Multidimensional Pictorial Data", Pattern Recognition, Vol. 11, pp. 115-125, 1979.

P.S. Chavez, "Atmospheric, Solar, and MTF Corrections for ERTS Digital Imagery", Proc. Amer. Soc. of Photog., Falls Church, Va., Abstract p. 69-a, October 1975.

R.W. Ehrich, "Detection of Global Edges in Textured Images", IEEE Trans. on Computers, Vol. C-26, pp. 589-603, June 1977.

P.T. Eliason, L.A. Soderblom, and P.S. Chavez, Jr., "Extraction of Topographic and Spectral Albedo Information from Multispectral Images", Photogrammetric Engineering and Remote Sensing, Vol. 48, No. 11, pp. 1571-1579, November 1981.

D.B. Elliott, "The application of an illumination model to a mountainous LANDSAT scene", M.S. Thesis, Virginia Tech, 1982.

M.A. Fischler, J.M. Tenenbaum and H.C. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique", Computer Graphics and Image Processing, Vol. 15, pp. 201-223, 1981.

M. Goldberg and S. Shlien, "A Clustering Scheme for Multispectral Images", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 8, No. 2, 1978.

Golub and Reinsch, "Singular Value Decomposition and Least Square Solution", Num. Math., Vol. 14, pp. 403-420, 1970.

W.E.L. Grimson, "An Implementation of a Computational Theory of Visual Surface Interpolation", MIT AI Memo, 1981.

R.M. Haralick and L.G. Shapiro, "The Consistent Labeling Problem: Part 1", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 1, No. 2, April 1979.

R.M. Haralick and L.G. Shapiro, "The Consistent Labeling Problem: Part 2", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 2, No. 3, May 1980.

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

B.K.P. Horn and R.J. Woodham, "Destriping LANDSAT MSS Images by Histogram Modification", Computer Graphics and Image Processing, Vol. 10, pp. 69-83, 1979.

A.W. Kuchler, "Potential Natural Vegetation of the Conterminous United States", American Geographical Society Special Publication, No. 36, 1964.

R.Y. Li and K.S. Fu, "Tree System Approach for LANDSAT Data Intepretation", Proceedings of the IEEE Symposium in Machine Processing of Remotely Sensed Data June 1976.

D. Marr, "Representing Visual Information", in Computer Vision System, A. Hanson and E. Riseman (Eds.), Academic Press, 1978.

A. Martelli, "Edge Detection Using Heuristic Search Methods", Computer Graphics and Image Processing, Vol. 1, pp. 169-182, 1972.

J.G. Moik, Digital Processing of Remotely Sensed Images, NASA Sp-431, 1980.

U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures", CACM, Vol. 18, pp. 335-345, 1971.

J.C. Munday, Jr., "Lake Ontario Water Mass Determination from ERTS-1", Proc. 9th Intl. Symp. Remote Sensing of Environment, ERIM, Ann Arbor, pp. 1355-1368, 1974.

M. Nagao and T. Matsuyama, A Structural Analysis of Complex Aerial Photographs, Plenum Press, 1980.

National Cartographic Information Center, Digital Terrain Tapes, User Guide, Second Edition, 1980.

G.L. Raines, T.W. Offield, and E.S. Santos, "Remote Sensing and Subsurface Definition of Faceies and Structure Related to Uranium Deposits, Powder River Basin, Wyoming", Econ. Geol., Vol. 73, pp. 1706-1723, 1978.

L.G. Shapiro and R.M. Haralick, "Structural Description and Inexact Matching", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 3, No. 5, September 1981.

Y. Shirai, "A Context Sensitive Line Finder for Recognition of Polyhedra", Artificial Intelligence, Vol. 4, pp. 95-119, 1973.

P. Switzer, W.S. Kowalik and R.J.P. Lyon, "Estimation of Atmospheric Path-Radiance by the Covariance Matrix Method", Photogrammetric Engineering and Remote Sensing, pp. 1469-1476, October 1981.

W.D. Thornbury, Regional Geomorphology of the United States, Wiley, N.Y., 1967.

J.R. Ullmann, R.M. Haralick, and L.G. Shapiro, "Computer Architecture for Solving Consistent Labeling Problem", Proceedings 14th Southeastern Symposium on System Theory, Blacksburg, Va., pp. 14-19, April 1982.

G.J. Vanderbrug, "Line Detection in Satellite Imagery", IEEE Trans. on Geoscience Electronics, Vol. 14, pp. 37-44, January 1976.

[1977a] G.J. VanderBrug, "Experiments in Iterative Enhancement of Linear Features", Computer Graphics and Image Processing, Vol. 6, pp. 25-42, 1977.

[1977b] G.J. Vanderbrug, "Curve Representation and Mapping", Computer Science Technical Report TR-561, U. of Maryland, August 1977.

R.K. Vincent, "Ratio Maps of Iron Ore Deposits, Atlantic City District, Wyoming", Proc. of Symp. on Significant Results Obtained from the ERTS-1, pp. 379-386, 1973.

D. Waltz, "Understanding Line Drawings of Scenes with Shadows", in The Psychology of Computer Vision, edited by P.H. Winston, McGraw-Hill, 1975.

S. Wang, D.B. Elliott, J.B. Campbell, R.W. Ehrich, and R.M. Haralick, "Spatial Reasoning in Remotely Sensed Data", IEEE Transactions on Geosience and Remote Sensing, Vol. GE-21, No. 1, pp. 94-101, January 1983.

R.L. Wildey, "Generalized Photoclinometry for Mariner 9", Icarus, Vol. 75, pp. 613-626, 1975.

Appendix A

COMPUTING THE ORIENTATION OF A BORDER SEGMENT

As mentioned in Section 5.3, in order to examine the stream junction patterns and find the flow directions of visible rivers, it is necessary to compute the orientations of border segments at their two ends. A curve can be fit into an equation

$c_1 x^2 + c_2 xy + c_3 y^2 + c_4 x + c_5 y + c_6 = 0$ (A.1).

The derivative is

$dy/dx = -(2 c_1 x + c_2 y + c_4) / (c_2 x + 2c_3 y + c_5)$

under the condition $c_2 x + 2 c_3 y + c_5 \neg= 0$. Then the orientation angle can be computed as ARCTAN(dy/dx) at end points.

In order to calculate the unknown coefficients c_1 through c_6, equation A.1 is put in the form

$[x^2 \; xy \; y^2 \; x \; y \; 1] \; [c_1 \; c_2 \; c_3 \; c_4 \; c_5 \; c_6] = 0$

The first vector can be calculated easily for each point (x,y). If there are m points in the curve, the m linear equations can be put in the form

$A \; c = 0$

where A is a mx6 matrix and $c=[c_1\ c_2\ c_3\ c_4\ c_5\ c_6]$. The solution vector c can be computed by using the Sigular Value Decomposition method [Golub and Reinsch, 1970].

Appendix B

FORWARD CHECKING ALGORITHM

[Shapiro and Haralick, 1981]

```
/ULTAB and MINERR are stacks, one table per level;
CONTROL:=forward;
while CONTROL=forward or some units have been assigned labels
do begin
  if all units have a label then CONTROL := back;
  if CONTROL = back then back up one level;
  U:=next unit to try;
  CONTROL:=back;
  while there are labels to try for unit U
    do begin
      L:=next label for U;
      PERR := error of partial labeling so far;
      BERR:=FORER(ULTAB,U,L);
      FERR:=FUTMIN(future units)
      if PERR+BERR+FERR <=e then
      begin
        ERRF:=UPDATE(ULTAB,U,L,PERR+BERR);
        if UPDATE fails then try next label;
        CONTROL:=forward;
        add (U,L) to the partial labeling;
        if all units have labels then stop;
        move forward one level;
      end
    end
  end
/
procedure FUTMIN(future units);
FUTMIN:=0;
for each future unit UF do
  FUTMIN:=FUTMIN+MINERR(UF)
end FUTMIN
/
procedure UPDATE(ULTAB,U,L,PASTERR);
UPDATE:=0;
for each future unit UF
do begin
  SMALLERR:=99999.;
  for each label LF that is eligible for UF
  do begin
    if{(U,L),(Uf,Lf)}is in the unit-label constraint
    relation
    then ERR:=0
    else ERR:=WEIGHT(U,UF);
    ULTAB(UF,LF):=ULTAB(UF,LF)+ERR;
    if ULTAB(UF,LF)<SMALLERR
    then SMALLERR:=ULTAB(UF,LF)
  end
  UPDATE:=UPDATE+SMALLERR
  if UPDATE+PASTERR>e then fail return;
  MINERR(UF):=SMALLERR
end
end UPDATE
/
```

In the above algorithm, FORER calculates $e_{pf}(u,l;U_p,h)$ of Equation 5.2; MINERR calculate min $e_{pf}(v,m;U_p,h)$ of Equation 5.3. e is the error bound for tree search, and WEIGHT is the error weighting function E_w in the saptial reasoning model.

Appendix C

GIPSY RUNFILES

GIPSY is General Image Processing System which is developed under the leadership of R.M. Haralick in Spatial Data Analysis Lab, Virginia Polytechnic Institute and State University. This software system has a command interpreter and hundreds of commands. Each command line contains command name, input file names, output file names, and user flags. The first word in the command line is the command name. The file names to the left of > are input file names, and the file names to the right of > are output file names. The characters inside the parentheses are user flags.

The command line can be typed by the user, then the command interactively asks the user questions. An alternative is to use Runfiles which contain all the command lines as well as user responses to the questions. It is very useful when doing repetive operations that have only minor changes step to step. All command lines in a Runfile must begin with a dollar sign in column one. Data must be entered on lines without a dollar sign in column one. Comments begin with $!. The last line in the file must be $EXIT.

Before the detailed Runfile is presented, a running description of all steps is given first. Preprocessing is done on an original four-band LANDSAT imagery FOUR.ORG by destriping (command DSTRPE) and dehazing (command PRNCP and DEHAZE). The resulting image is FOUR.DEH.

Next, a binary shadow image is computed by creating a three-band ratio image (command RATIO) and performing clustering twice. The first clustering is done by locating cluster centers (command CCENTR) and executing ISODATA algorithm (command ISODAT). The second clustering is done by using command ISODAT with T flag. Image IIC.C is created in which each material cluster is separated into two subclusters. Finally, command PRPSIF is used to create a binary shadow image IIC.SHA.

After the shadow image is created, diffuse light image IIC.DFU can be generated easily by commands PRPSIF and MSKIMG, and estimated reflectance image IIC.R1 can be generated by PRPSIF. To get the estimated topographic modulation image IIC.T1, one simply divide the difference image IIC.SUN of the dehazed image and the diffuse light image by IIC.R1 (command RATIO). The difference image can be computed by commands ARITHM and SLICE. After very bright pixels are removed from IIC.T1 by command SLICE, the principal component image is the final topographic modulation image IIC.TOP (commands PRNCP and LCOMB).

The ratio image of IIC.SUN over IIC.TOP is IIC.R2. However, this is not the final reflectance image. One needs to assign the mean of the bright pixels from the same material cluster into the dark pixels. The maens for bright pixels can be found by command MNCOV which creates a property file IICR2.MN. To mask out the bright pixels from IIC.C, command MSKIMG is used with an all-zero image (command MKCHK). 1 is added (command ARITHM) to match the data in IICR2.MN. Command EXSIF is used to modify a symbolic image into a numeric image or in the reverse order. Now it is ready to assign the bright means into dark pixels by command PRPSIF. Finally, this result is combined with IIC.R2 to obtain the final reflectance image IIC.REF.

Next, it is necessary to detect water pixels. Munday's curve is created by command MKFNIM. The chromaticity coordinates are computed by commands ARITHM and RATIO. The pixel distances to the curve are computed by command PXLDST. Then pixels can be classified by command CLAS2D. The resulting image is thresholded by command SLICE to obtain a binary image of water pixels. The result can be cleaned up by expanding, shrinking, (commands FILL, SHRINK) and deleting small regions (command C0NCM, PRPMBI, and ELIMRG). The final result is WAT.ELI.

Next, one wants to detect ridge and valley segments from the shadow image. Before border segments can be detected, it is necessary to change the binary shadow image into a symbolic one (commands CONCM and RELBL). Noisy regions can be cleaned up by commands PRPMBI, ELIMRG, FILL, and RELBL. The resulting image SHA.RE1 is changed to a numeric one temporarily. To overlay the water regions on this connected components image SHA.RE1, WAT.ELI is changed to a numeric one (command EXSIF) and 100 is added to all pixels (command ARITHM). The overlay is done by MSKIMG. The result is changed back to a symbolic one (EXSIF) and relabeled as SHA.REL (command RELBL).

Border segments can be created by command BORCHN which creates a property file SHA.LP1 and a chain code file SHA.CC. Before ridge and valley segments are identified, more information is put into the property file. The first one is the orientations at ends of border segments (command BORORI). The second is junction information (command BORJCT). The third is brightness information (commands SBIMG, EXSIF, PRPMBI, and LBREG). The fourth is water information (commands EXSIF, PRPMBI, LBREG). To obtain ridge and valley segments, command LANDRV creates updated border segment property file SHA.LP5 and junction property file SHA.JP2. They are then used as input files of CHNSIF to create the symbolic image SHA.VR.

Next, one concentrates on how to reconstruct an elevation model. A rough estimated elevation model is computed by commands ELESLP and ELEGRO first. Three different methods of interpolation can be performed by using commands INTERP, QVINTP and INTRPD. Binary images of valley segments (SHA.1V), ridge segments (SHA.1R) or both (SHA.OR) are required inputs for these commands. They are created by commands MKCHK, EXSIF, and BOOL. For the distance method (INTRPD), command PXLDST is executed to compute the distances. In order to improve this rough estimated elevation model, IIC.TOP and a derivative image (command CNSFCT) are sent to command BAYTOP. The result is combined with the previous result from ELESLP, and ELEGRO is run again to create an updated elevation model.

After one has an optimal surface, LANDSAT imagery can be reconstructed by first using a shadow image (command SHADOW), local slope image (command CBFCT), and reflectance image IIC.REF in the command SHADE. Then the resulting image is added by the diffuse light image (ARITHM). Finally, drainage network can be obtained by commands STRUD, STRCL, and CHNSIF.

```
$!
$!      DESTRIPING
$!
$ DSTRPE FOUR.ORG > FOUR.DSP
$!
$!      HAZE REMOVAL
$!
$ PRNCP FOUR.DSP > FOUR.COV, FOUR.EIG (A)
  1. /! fraction of area which will be sampled
$!
$ DEHAZE FOUR.DSP, FOUR.COV, FOUR.EIG > FOUR.DEH (A)
  4 /! the band with zero haze
$!
$!      TAKE RATIO
$!
$ RATIO FOUR.DEH > FOUR.R30
  3 /! number of bands
  2 3 4 /! bands of numerator image
  1 2 3 /! bands of denominator image
  30. 30. 30. /! constants to multiply the numerator image
  N /! do not change the number of bits
$!
$!      FIND CLUSTER CENTERS AND SIZES
$!
$ CCENTR FOUR.R30 > FOURR30.CC (AQ)
```

```
  7 /! number of quantizations
  10 /! threshold of population count
$!
$!      CREATE MATERIAL CLUSTER IMAGE
$!
$ ISODAT FOUR.R30, FOURR30.CC > FOUR.CC, FOUR.PP (AP)
  1 /! allowed difference of means in 2 consecutive iterations
$!
$!      SEPARATE DARK AND BRIGHT PIXELS FOR EACH CLUSTER
$!
$ ISODAT FOUR.DEH, FOUR.CC > IIC.C, IIC.P (AT)
  1. /! allowed difference of means in 2 consecutive iterations
$!
$!       CREATE SHADOW IMAGE
$!
$  PRPSIF IIC.C, IIC.P > IIC.SHA (S)
  3 /! property number
$!
$!      PUT MEANS OF DARK SUBCLUSTERS INTO EACH PIXEL
$!
  PRPSIF FOUR.CC, IIC.P > IIC.D1 (MDSN)
  2 /! number of sub-regions inside each original region
  1 /! sub-region number used to represent the whole region
  4 /! number of properties
  5 6 7 8 /! property numbers
```

```
$!
$!      CREATE DIFFUSE LIGHT IMAGERY
$!
$ MSKIMG FOUR.DEH, IIC.D1, IIC.SHA > IIC.DFU (A)
  4 /! number of bands
  1 /! code of predicate
  1 /! constants in the predicate
$!
$!      CREATE ESTIMATED REFLECTANCE IMAGERY
$!
$ PRPSIF FOUR.CC, IIC.P > IIC.R1 (NSMD)
  2 /! number of sub-regions inside each original region
  1 /! sub-region number used to represent the whole region
  4 /! number of properties
  19 20 21 22 /! property numbers
$!
$!      SUBSTRACT DIFFUSE LIGHT FROM DEHAZED IMAGERY
$!
$ ARITHM FOUR.DEH, IIC.DFU > IIC.S1 (BS)
  N /! do not change the number of bits
$!
$!      RESET NEGATIVE NUMBERS TO 0
$!
$ SLICE IIC.S1 > IIC.SUN (AS)
  1 /! semislice at dark end
  0 /! threshold for dark end
$!
$!      ESTIMATED TOPOGRAPHIC MODULATION IMAGERY
$!
$ RATIO IIC.SUN, IIC.R1 > IIC.T1 (B)
  40 40 40 40 /! constants to multiply the numerator image
  N           /! do not change the number of bits
$!
$!    REMOVE VERY BRIGHT PIXELS
$!
$ SLICE IIC.T1 > IIC.T2 (AH)
  2 /! semislice at bright end
  10 /! min. population for bright end
$!
$!      TOPOGRAPHIC MODULATION IMAGE
$!
$ PRNCP IIC.T2 > IICT2.MC, IICT2.EIG (A)
  1 /! fraction of area which will be sampled
$ LCOMB IIC.T2, IICT2.EIG > IIC.TOP (A)
$!
$!      DIVIDE (DEHAZED-DIFFUSE LIGHT) BY TOPO. MOD.
$!
$ RATIO IIC.SUN, IIC.TOP > IIC.R2
  4 /! number of bands
  1 2 3 4 /! bands of numerator image
  200 200 200 200 /! constants to multiply the numerator image
  N               /! do not change the number of bits
$!
$!      FIND MEANS FOR DARK AND BRIGHT SUBCLUSTERS
$!
$ MNCOV IIC.R2, IIC.C > IICR2.MN (AM)
  1. /! percentage of points sampled
$!
$!      CREATE AN ALL-0 IMAGE
$!
$ MKCHK FOUR.Z
  80 80 /! image size
  80 80 /! checker size
  0 0 /! min., max. graytone
  0     /! percentage of noisy pixels
$!
$!      GET THE CLUSTER NUMBERS FOR DARK PIXELS ONLY
$!
$ MSKIMG IIC.C, FOUR.Z, IIC.SHA > IIC.C1
  1 /! code of predicate
  1 /! constant in the predicate
$ EXSIF IIC.C1
  MID 18 0
  Y DONE
$!
```

```
$!      ADD 1 BECAUSE OF THE STRUCTURE OF PROPERTY FILE
$!      IICR2.MN
$!
$ ARITHM IIC.C1 > IIC.C2 (A)
  1 /! constant to be added
  N /! do not change number of bits
$ EXSIF IIC.C2
  MID 18 1
 Y DONE
$!
$!      PUT BRIGHT MEANS INTO DARK PIXELS
$!
$ PRPSIF IIC.C2, IICR2.MN > IIC.R3 (NSM)
  4 /! number of properties
  3 4 5 6 /! property numbers
$!
$!      COMBIN BRIGHT AND DARK PIXELS
$!
$ MSKIMG IIC.R2, IIC.R3, IIC.SHA > IIC.REF (A)
  4 /! number of bands
  1 /! predicate code
  2 /! constant in the predicate
$!
$!      MAKE A CURVE
$!
```

```
$ MKFNIM IIC.FUN
  80 80 /! image size
  250 270 /! min., max. for x axis
  250 500 /! min., max. for y axis
$!
$!      COMPUTE DISTANCES FROM THE CURVE
$!
$ PXLDST IIC.FUN > IIC.DST (E)
  1 /! number of region to be computed from
$!
$!      CHROMATICITY COORDINATES
$!
$ ARITHM IIC.R1 > IICR1.MUL (M)
  3 /! number of bands
  1 2 3 /! band specification
  34 27.3 45.8 /! multiplying constants
  n             /! do not change the number of bits
$ RATIO IICR1.MUL > IIC.CHR (BSN)
  1 1 1 /! denominator is a linear combination of the numerator
image
  1000 1000 0 /! constants to multiply the numerator image
$!
$!      CLASSIFY WATER PIXELS
$!
$ CLAS2D IIC.DST, IIC.CHR > IIC.WAT (W)
$!
```

```
$!      GET BINARY IMAGE OF WATER PIXELS
$!
$ SLICE IIC.WAT > WAT.BIN
  1 /! highest number of slices
  1 /! number of non-background slices
  0 30 /! low, high thresholds
$!
$!      EXPAND AND SHRINK
$!
$ FILL WAT.BIN > WAT.FIL (E)
$ SHRINK WAT.FIL > WAT.SHR (F)
$!
$!      CLEAN UP SMALL REGIONS
$!
$ CONCM WAT.SHR > WAT.CON
  1 /! select one label
$ PRPMBI WAT.CON, FOUR.DEH > WAT.PRP (L)
  1 /! band number of input image 1
  1 /! band number of input image 2
  1 100 /! low, high label
$ ELIMRG WAT.CON, WAT.PRP > WAT.ELI
  1 /! property number
  10 /! threshold
$!
$!      CONNECTED COMPONENTS FROM SHADOW IMAGE
```

```
$!
$ CONCM IIC.SHA > JUNK.JK (SF)
$!
$ RELBL JUNK.JK > SHA.CON
$!
$!      CLEAN UP SMALL REGIONS
$!
$ PRPMBI SHA.CON, FOUR.DEH > SHA.PRP
  1 /! band number of input image 1
  1 /! band number of input image 2
$ ELIMRG SHA.CON, SHA.PRP > SHA.ELI
  1 /! property number
  10 /! threshold
$ FILL SHA.ELI > SHA.FIL
$!
$ RELBL SHA.FIL > SHA.RE1
$!
$ EXSIF SHA.RE1
  MID 18 0
  Y DONE
$!
$!      GET NON-BACKGROUND PIXELS
$!
$ SLICE WAT.ELI > WAT.B1
  1 /! highest number of slices
  1 /! number of non-background slices
  2 10 /! low, high threshold
$ EXSIF WAT.ELI
  MID 18 0
  Y DONE
$ ARITHM WAT.ELI > WAT.100 (A)
  100 /! constant to be added
  N   /! do not change the number of bits
$!
$!      OVERLAY WATER PIXELS ON CONNECTED COMPONENTS
$!      IMAGE OF SHADOW IMAGE
$!
$ MSKIMG SHA.RE1, WAT.100, WAT.B1 > SHA.RE2
  1 /! code of predicate
  0 /! constant in the predicate
$ EXSIF SHA.RE2
  MID 18 1
  Y DONE
$ RELBL SHA.RE2 > SHA.REL
$!
$!      CREATE BORDER SEGMENTS
$!
$ BORCHN SHA.REL > SHA.LP1, SHA.CC (G)
$!
$!      SIF IMAGE OF BORDER SEGMENTS
```

```
$!
$ CHNSIF SHA.LP1, SHA.CC > SHA.BB (A)
$!
$!      COMPUTE ORIENTATIONS AT ENDS OF BORDER SEGMENT
$!
$ BORORI SHA.LP1, SHA.CC > SHA.LP2
$!
$!      CREATE JUNCTION PROPERTY FILE
$!
$ BORJCT SHA.REL, SHA.BB, SHA.LP2 > SHA.JP1, SHA.LP3
$!
$!      PUT REGION BRIGHTNESS INFORMATION INTO PROPERTY FILE
$!      FOR LEFT AND RIGHT REGIONS
$!
$ SBIMG IIC.SHA > IIC.NUM (B)
$!
$ EXSIF IIC.NUM
  MID 18 0
  Y DONE
$ PRPMBI SHA.REL, IIC.NUM > IICNUM.PRP
$!
$ LBREG SHA.LP3, IICNUM.PRP > SHA.L31 (I)
  4 /! number of properties to be updated
  8 /! property number for input file 1
  24 /! output property number
  4  /! property number for input file 2
  3  /! predicate code
  1.5 /! constant in the predicate
  1 2 /! true, false label
  8 30 1 0 /! similar as above
  9 25 4 3 1.5 1 2
  9 31 1 0
$!
$!      PUT WATER INFORMATION INTO PROPERTY FILE
$!
$ EXSIF WAT.B1
  MID 18 0
  Y DONE
$ PRPMBI SHA.REL, WAT.B1 > WAT.PRP
$!
$ LBREG SHA.L31, WAT.PRP > SHA.LP4
  2 /! number of properties to be updated
  8 /! property number for input file 1
  33 /! output property number
  4  /! property number for input file 2
  3  /! predicate code
  0.5 /! constant in the predicate
  1 2 /! true, false label
  9 34 4 3 0.5 1 2 /! similar as above
$!
```

```
$!      CLASSIFY RIDGE/VALLEY SEGMENTS
$!
$ LANDRV SHA.JP1, SHA.LP4 > SHA.JP2, SHA.LP5 (B)
$!
$!      SIF IMAGE OF RIDGE/VALLEY SEGMENTS
$!
$ CHNSIF SHA.LP5, SHA.CC > SHA.VR (PA)
  26 /! select one property
  3  /! how many bands for this property
  1 2 3 /! property value for each band
$!
$!      CREATE ESTIMATED FIRST ORDER DERIVATIVES FOR
$!      ELEVATION GROWING
$!
$ ELESLP SHA.LP5, SHA.VR, SHA.CC > SHA.DEC
  0.4 /! slope for pixels close to ridges
  0.02 /! slope for pixels close to valleys
  0.1 /! slope for other pixels
  200 /! ground distance between 2 pixels
$!
$!      ELEVATION GROWING
$!
$ ELEGRO SHA.LP5, SHA.CC, SHA.DEC > SHA.LP5, SHA.GRO
  950    /! lowest elvation
$!
```

```
$!      CREATE BINARY IMAGE OF RIDGE/VALLEY SEGMENTS
$!      TO BE USED IN INTERPOLATION
$!
$ MKCHK FOUR.ONE
  80 80 /! image size
  80 80 /! checker size
  1 1   /! low, high graytones
  0     /! percentage of noisy pixels
$ EXSIF FOUR.ONE
  MID 18 1
 Y DONE
$ BOOL SHA.VR, FOUR.ONE > SHA.1V
  1 /! how many band from image 1
  1 /! band selection
  1 /! logical and
$ BOOL SHA.VR, FOUR.ONE > SHA.TR
  1 /! how many band from image 1
  3 /! band selection
  1 /! logical and
$ BOOL SHA.1V, SHA.TR > SHA.OR
  5 /! logical or
$ BOOL SHA.1V, SHA.TR > SHA.AND
  1 /! logical and
$ BOOL SHA.TR, SHA.AND > SHA.1R
  2 /! image 1 .and. (.not. image 2)
```

```
$!
$!      INTERPOLATION BETWEEN RIDGES AND VALLEYS
$!
$!      LAPLACIAN SURFACE
$!
$ INTERP SHA.GRO, SHA.OR > SHA.LAP (B)
  3 3 /! mask size
  0 1 0 /! row one of mask
  1 -4 1 /! row two of mask
  0 1 0  /! row three of mask
  Y      /! grid is correct
  20     /! number of iteration
  1      /! cutoff value for the max. difference between
iterations
  N      /! do not continue
$!
$!      QUADRATIC VARIATION
$!
$ QVINTP SHA.GRO, SHA.OR > SHA.QV (B)
  20 /! number of iterations
  1  /! cutoff value for the max. difference between iterations
  N  /! do not continue
$!
$!      DISTANCE METHOD
$!
$ PXLDST SHA.1V, SHA.GRO > V.DST
```

```
  1 /! number of region to be computed from
$ PXLDST SHA.1R, SHA.GRO > R.DST
  1 /! number of region to be computed from
$ INTRPD V.DST, R.DST > SHA.D1 (V)
$!
$ INTRPD V.DSTY, R.DST > SHA.D3 (CV)
$!
$ INTRPD V.DST, R.DST > SHA.D5 (FV)
$!
$!      COMPUTE FIRST ORDER DERIVATIVES FROM LAPLACIAN
$!      SURFACE
$!
$ CNSFCT SHA.LAP > SHA.SL
  3 3 /! window size
  17  /! number of bits
$!
$!      USE TOPOGRAPHIC MODULATION IMAGE TO COMPUTE
$!      BETTER FIRST ORDER DERIVATIVES
$!
$ BAYTOP SHA.SL, IIC.TOP > SHA.BAY
  119 /! corrected sun azimuth
  45  /! sun elevation angle
  200 /! ground distance
  1. /! multiplying constant
  .466 /! standard deviation of the Normal distribution
```

```
$!
$!      REPLACE THE OLD FIRST ORDER DERIVATIVES
$!
$ COMBIN SHA.DEC, SHA.BAY > SHA.DB
  3 /! number of bands from image 1
  1 2 3 /! band specification
  2     /! number of bands from image 2
  1 2   /! band specification
$!
$!      RUN THE ELEVATION GROWING AGAIN
$!
$ ELEGRO SHA.LP5, SHA.CC, SHA.DB > SHA.LP1, SHA.GR1 (B)
  950 /! lowest elevation
$!
$!      AFTER SEVERAL ITERATIONS, THIS IS THE OPTIMAL
$!      SURFACE
$!
$ INTERP SHA.GR1, SHA.OR > SHA.LAP (B)
  3 3 /! mask size
  0 1 0 /! row 1 of mask
  1 -4 1 /! row 2 of mask
  0 1 0  /! row 3 of mask
  Y      /! grid is correct
  20     /! number of iterations
  1      /! cutoff value
  N      /! do not continue
$!
$!      RECONSTRUCTION OF LANDSAT IMAGERY
$!
$ SHADOW SHALAP.OPT > SHALAP.SUN
  12 /! orientation of the top of the image
  119 /! azimuth of sun
  1   /! ground distance multiplier
  45  /! elevation of sun
$ CBFCT SHALAP.OPT > SHALAP.CBF (A)
  5 5 /! window size
  17  /! number of bits
  1   /! iteration
$ SHADE SHALAP.SUN, SHALAP.CBF, IIC.REF > SHALAP.RC1 (A)
  107 /! corrected sun azimuth
  45  /! sun elevation
  0.  /! diffuse light ratio
  10  /! direct sun illumination
  1   /! ground distance multiplier
$!
$!      ADD DIFFUSE LIGHT
$!
$! ARITHM IIC.DFU > IICDFU.M10 (BM)
  10 10 10 10 /! multiplying constants
  N           /! do not change number of bits
```

```
$ ARITHM SHALAP.RC1, IICDFU.M10 > SHALAP.REC (AB)
  N /! do not change number of bits
$!
$!      PUT CONSTRAINTS AT JUNCTIONS INTO PROPERTY FILE
$!
$ STRUD SHA.LP6, SHA.JP2 > SHA.JP3 (A)
$!
$!      RUN CONSISTENT LABELING TO FIND FLOW DIRECTIONS
$!
$ STRCL SHA.LP6, SHA.JP3 > SHA.LP7
  100 /! upper limit of the number of units
$!
$!      CREATE DARINAGE NETWORK IMAGE
$!
$ CHNSIF SHA.LP7, SHA.CC > SHA.NET (PA)
  35 /! select one property
  1  /! how many band for this property
  1  /! property value
$ EXIT
```