AD-A101 422 ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA F/G 8/2
SHADED RELIEF IMAGES FOR CARTOGRAPHIC APPLICATIONS.(U)
APR 81 C C TAYLOR
UNCLASSIFIED ETL-0259 NL

1 OF 3
AD A
101422

5

LEVEL II

ETL-0259

AD A101422

Shaded relief images for cartographic applications

Cyrus C. Taylor

APRIL 1981

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060

E
T
L

81 7

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0259	2. GOVT ACCESSION NO. AD-A207 422	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SHADED RELIEF IMAGES FOR CARTOGRAPHIC APPLICATIONS		5. TYPE OF REPORT & PERIOD COVERED Research Note
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) CYRUS C. TAYLOR		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4303
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060		12. REPORT DATE Apr [redacted] 81
		13. NUMBER OF PAGES 193
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Atmospheric Haze, Gray-Shade Image, Image Formation, Light Scattering, Orthonormal Projection, Perspective Projection, Photometry, Polynomial Data Base, Relief Contours, Shaded Relief, Variable Sun Angle

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computer generation of shaded relief images for cartographic applications is analyzed. The geometric theory of image formation is presented in some detail, and is used to motivate the discussion of the image simulation algorithms developed at the U.S. Army Engineer Topographic Laboratories (ETL). Several algorithms devised to address specific cartographic problems, such as variable sun angle, haze simulation, and relief contour algorithms, are also discussed. The successful implementation of these algorithms is described. Appendixes summarizing details of the theory are included, as are a Software User's Guide, sample images, and listings of the ETL software.

PREFACE

The work described in this report was performed in the Automated Cartography Branch, Mapping Developments Division, U.S. Army Engineer Topographic Laboratories by Mr. Cyrus C. Taylor. His training and technical guidance were provided by Mr. James R. Jancaitis, Project Engineer. The study was done during the summers of 1978 and 1979 under the supervision of Mr. W. Howard Carr, Chief, Automated Cartography Branch; Mr. Eugene P. Griffin, Chief, Mapping Developments Division; and Mr. Howard O. McComas, Director, Topographic Development Laboratory.

This work was supported by the Defense Mapping Agency under the research and development sub-task entitled "Advanced SACARTS Software," 64701B/4303.

COL Daniel L. Lycan, CE was Commander and Director and Mr. Robert P. Macchia was Technical Director of the Engineer Topographic Laboratories during the study and report preparation.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS
(Continued)

ILLUSTRATIONS
(Continued)

TABLES

SHADED RELIEF IMAGES FOR CARTOGRAPHIC APPLICATIONS

I. INTRODUCTION

It is the task of the cart‹ ,ıapher to display spatially distributed data in the most cost-effective and easily perceived manner available. Since the uses to which the data will be applied are varied, the optimum way of displaying the data will also vary. This report examines a variety of related means of displaying one type of data, terrain elevation, for which versatile and efficient software has been produced at the Automated Cartography Branch, U.S. Army Engineer Topograhic Laboratories (ETL).

The problem of presenting terrain relief is not new. The history of cartography has been a progression from crude symbols representing mountains, to hachures, and then to the familiar contour lines as a means of portraying the surface of the earth.[1] With clear advantages owing to the presence of quantitative elevation information and the ease of registration with non-hypsometric information, the contour map has become the standard cartographic product depicting terrain relief. Nevertheless, terrain form is often difficult to perceive quickly and accurately in a representation limited to contours. Consequently, it is often desirable to supplement the contours with additional terrain representations.[2] Additionally, contour maps are not ideal for all users of cartographic products. With the advent of high speed digital computers and the creation of comprehensive digital terrain models, a variety of inexpensive cartographic products can be created for special applications, such as flight simulations or cut and fill studies.

The research described in this report approaches these problems from the standpoint of the qualitative representation of terrain based on the generation of idealized images. Two major types of products are investigated: (1) the analytic creation of shaded relief overlays for contour maps, and (2) the production of synthetic photographs. Both products exploit the variation of brightness between different areas of terrain owing to varying inclinations of the source of illumination. Since the resulting cartographic products are dependent on very few variables, it should be easy to train people to use efficiently the wealth of qualitative information present in these products.

---

[1] A. Robinson, R. Sale, and J. Morrison, *Elements of Cartography*, Fourth Ed., Wiley, 1978, pp. 15–31.

[2] J. Deates, *Cartographic Design and Production*, Longman, 1973, p. 73.

This report systematically examines the problems associated with the creation of shaded relief overlays, synthetic photographs, relief contours, and related products. The theories of forming images by optical systems and of light scattering from solid surfaces are examined first. Constraints on the implementation of the results of the theory imposed by available equipment are discussed next. The algorithms developed at ETL to produce shaded relief cartographic products are then described, with emphasis on the versatility provided by the polynomial terrain data base that is used as the basis of the ETL software. Finally, the report examines the results of the implementation of these algorithms on the ETL–PDP–11/45.

II. THEORY

As outlined in the introduction of this report, the goal of the work described herein is to develop algorithms to produce terrain representations with significant qualitative informational content. As with most cartographic products, these terrain representations are visual representations, which are used because it is easier to interpret spatial data presented in a visual form. These qualitative representations are more or less highly specialized shaded relief images of the terrain. In this manner, information regarding landform can be quickly retrieved from the image on the basis of the continuous tone image and implicit lighting directions and surface characteristics.

In this section, the physical processes that we seek to simulate are examined. First, an analysis of the theoretical basis of the problem is essential. Although the basic theory is not new, a source that contains all aspects of the basic theory is needed.

At the outset, it is important to note the level at which we seek to understand the processes of image formation. We are interested only in the basic aspects of image formation, such as can be treated by means of classical physical quantities and by means of simple geometric optics. We are not interested in the more detailed understanding afforded by the use of electromagnetic theory, nor are we interested in the chemical details of image formation in real imaging systems such as the retina or photographic film. Thus, we will not consider diffraction and interference or quantum effects. This hardly seems a restriction for our purposes, since these effects are only important at a much smaller scale than we are concerned with.

We will divide the problem into conceptually and physically independant subproblems and will discuss briefly the solution of each. There are four intermediate problems that must be solved if we are to simulate shaded relief images of the terrain. These problems are

1. The description of the lighting source illuminating the terrain.
2. The interaction of the incident light with the surface of the terrain.
3. The propagation of the light from the terrain to the particular (idealized) imaging system.
4. The recording of the important properties of the reflected light in some permanent image.

The first and third problems deal only with the direction of the propagation of the light; consequently, we do not need to consider the intensity of the light rays with which we are dealing.

The first problem is only implicitly dealt with in this section of the report, since, as we shall see, it is sufficient to define the direction and intensity of the illuminating light locally; i.e. at each point of the surface which we are imaging. Throughout much of this report, we shall assume a uniform direction and intensity of illumination across the entire area imaged.

The third problem is explicitly considered in some detail, since it is important to understand the various types of idealized images that we shall simulate. Two image types will be examined, the orthonormal and perspective projections. The orthonormal projection must be understood if shaded relief overlays are to be produced, for example, contour maps. Perspective projections are the mathematical idealization of the familiar types of imaging, such as vision and photography. These two projection types are sufficient to produce a variety of cartographic products. A third projection type which is also of cartographic interest is the oblique projection. This type has been discussed in a previous report.[3]

In the two remaining problems, the interaction of light with the terrain surface and with a recording medium, one must consider the intensity of light as well as the

---

[3] C. Taylor, *Parallel Profile Plots for Visual Terrain Display*, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, ETL-0115, September 1977, AD-A051 483, pp. 8-11.

direction in which the light is propagating. We must therefore use some of the elements of photometry, which has developed the concepts necessary to discuss these problems mathematically. Since the reader may not be familiar with the basic concepts of photometry, a brief discussion of the elementary photometric variables is included. This will enable the two remaining problems to be examined.

The description of the second problem, the interaction of light with the surface of the earth, takes the form of a light-scattering "law." Such a law is a mathematical relationship between the brightness of the surface and a function of the relative positions of the observer and the source of illumination. It is important to note that we know of no law that accurately describes the interaction of light with all types of terrain under all lighting conditions, nor do we want such law for our purposes. Any such law would be very complex, since we would have to treat such variables as terrain composition. More importantly, such a law would be extremely difficult to interpret. Instead, we seek a light-scattering law that provides a good qualitative representation of the form of the terrain. In this report, two such laws are described. The two were selected for their simplicity of mathematical form, physical basis, and extensive empirical justification.[4] Although this description of the interaction of light with the terrain is not unique or in any sense the best, the two laws are representative and can be quickly understood.

The last problem is that of recording in a permanent image the important properties of the light reflected from terrain. The solution is a function relating the brightness of a given surface point with the illumination of the corresponding point of the image. For the case of the perspective projection, we derive an exact solution. This solution simulates such photographic phenomena as vignetting. Such effects are perceptually objectionable for our purposes since qualitative analysis of the image becomes more complex. Thus, we shall simplify our formulas by using reasonable approximations. The result will be readily amenable to algorithmic implementation and will be suitable for the simulation of both perspective and orthonormal shaded relief images.

Thus, in this section, the theoretical basis for the production of shaded relief images of the terrain is developed. This problem is divided into four independent subproblems. A theoretical solution is developed for each. Since the subproblems are independent, this approach results in an algorithmic solution to the problem of shaded relief images that is readily amendable to computer implementation.

---

[4] B. Horn. *Hill Shading and the Reflectance Map in Image Understanding in Proceedings*, Ed. by L. Baumann, Science Applications, Inc., Report SAI-80-895, 1979, pp. 79-120.

A. THE GEOMETRY OF IMAGE FORMATION. The first problem to be addressed is that of the geometry of image formation for the imaging systems of interest. This problem is treated first since it is the means of correlating points of the terrain with points of the image. Its solution plays a central role in any algorithms developed for shaded relief images. The process of image formation is, as noted before, best treated in terms of geometric optics. In such an approach, the formulation of the problems is in terms of projective geometry. Thus, the solutions derived below will be projection equations defining the geometry of image formation for the types of images of interest.

Two types of projections to generate images are considered here: orthonormal projections and perspective projections. These two projections were chosen as being of primary cartographic interest at the present time. The orthonormal projection of gray shade information may be used as a contour map overlay, enabling a quick analysis of terrain form by aiding the task of interpreting contour lines. Perspective projections of gray shade data will be valuable wherever a relistic portrayal of the terrain, as seen from a given point, is called for.

1. Orthonormal Projections. An orthonormal projection may be defined as the projection from an object onto a projection plane parallel to the base of the object, using projectors that are uniformly perpendicular to the projection plane. (see figure 1). Thus, the projection equations will define a transformation from a three-dimensional space to a two-dimensional plane. Accordingly, a three-dimensional cartesian coordinate system (x, y, z) can be introduced that will be used to define the position of points of the object, and a two-dimensional (x′, y′) cartesian coordinate system can be used in the projection plane. For simplicity, let the x′, x and the y′, y axes be parallel. Then, the projection equations are simply

$$X' = X \tag{1}$$
$$Y' = Y \tag{2}$$

Note that the z-coordinate (elevation) drops out of the projection equations. Equations (1) and (2) apply only to maps created at a 1:1 scale. To allow for other map scales, a parameter, α, is introduced, such that for an α:1 map,

$$X' = 1/\alpha \; X \tag{3}$$
$$Y' = 1/\alpha \; Y \tag{4}$$

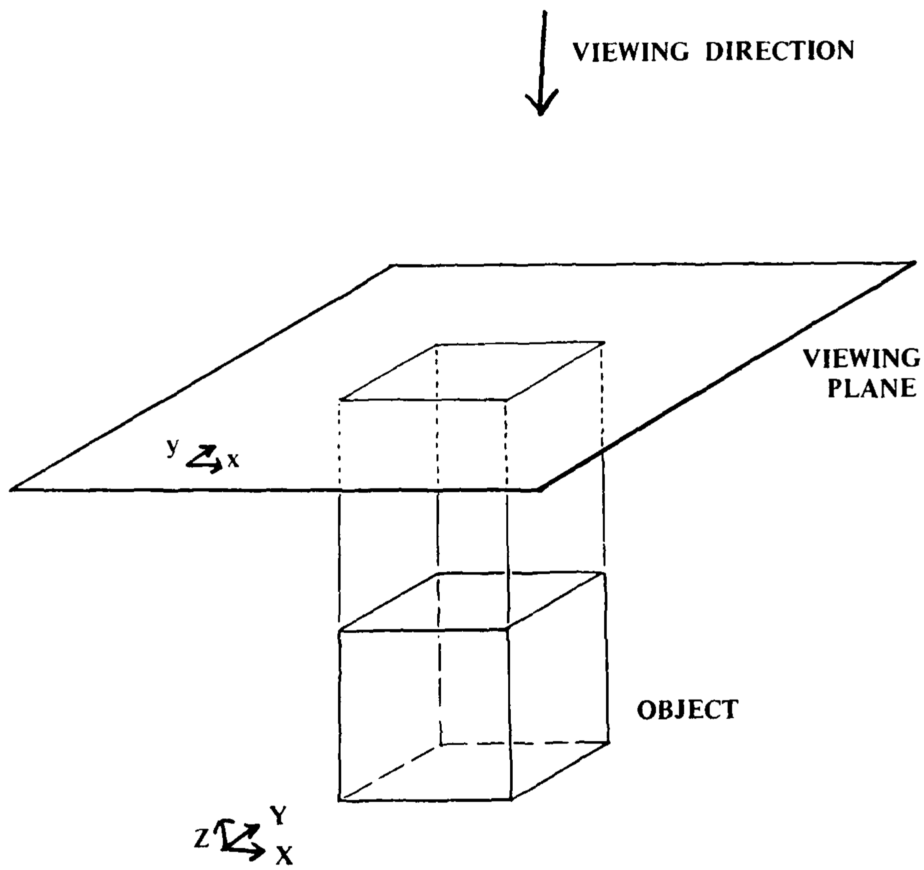


FIGURE 1. Orthonormal Projection.

It should be noted that the earth's surface cannot be depicted in an orthonormal projection where the projection plane is everywhere perpendicular to the earth's surface, since a sphere is a nondevelopable surface. All standard topographic maps are projections in which this problem is minimized, and it is assumed that the data base for preparing orthonormal shaded relief projections has been transformed in this manner.

2. **Perspective Projection.** A perspective projection is a projection of an object onto an image plane, characterized by nonparallel projectors that converge to a point behind the image plane (see figure 2). We are concerned with the particular case in which the image plane is parallel to the plane defined by the Z-and X-Axes of some coordinate system associated with the object. This is not a limitation on the generality of the transformation equations that will be derived: a suitable rotation of the coordinate system of the object will enable any perspective view to be produced.

We begin our derivation of the projection equations by defining the relevant parameters (see figures 2 through 6), assuming the usual right-handed orthogonal (X, Y, Z) coordinate system associated with the object. The origin of this coordinate system is at some point Q. At a point $X = 0$, $Y = \ell$; $Z = 0$, a projection plane (PP) parallel to the $XZ - Z$ plane of the coordinate system intersects the Y axis. The point F, located a distance h above the origin of the coordinate system plays a role analagous to that of the pinhole in a pinhole camera: all rays from the object will be assumed to converge to it. Two vertical planes, P and S, are used; P is perpendicular to the projection plane pp, and S is located at an angles to it. Both planes contain points Q and F. Finally, an image coordinate system (X', Y') is used in the projection plane, PP. This coordinate system has its origin at the point Q' with the X' axis parallel to the X axis, and the Y' axis parallel to the Z axis. The coordinates of Q' in terms of the (X, Y, Z) coordinate system are

$$X\,(Q') = X_o \tag{5}$$
$$Y\,(Q') = \ell \tag{6}$$
$$Z\,(Q') = Z \tag{7}$$

Consider first the X coordinate of some point of the object, Q, with coordinates (X, Y, Z). This point will be contained in some vertical plane S, inclined at an angle $\theta = \tan\,(x/y)$ to the vertical plane P (see figure 5). The projection of this point is seen to be

$$X' = \ell\,(x/y) - X_o \tag{8}$$

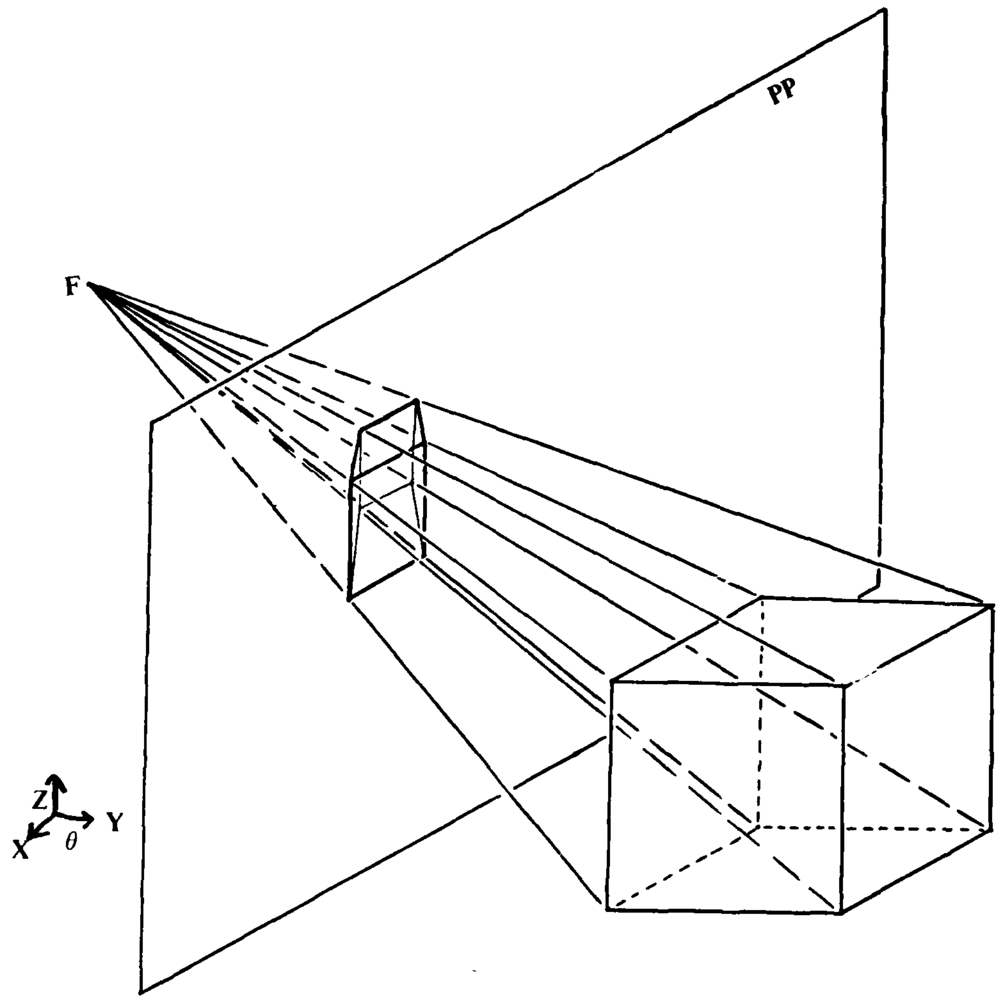


FIGURE 2. Perspective View.

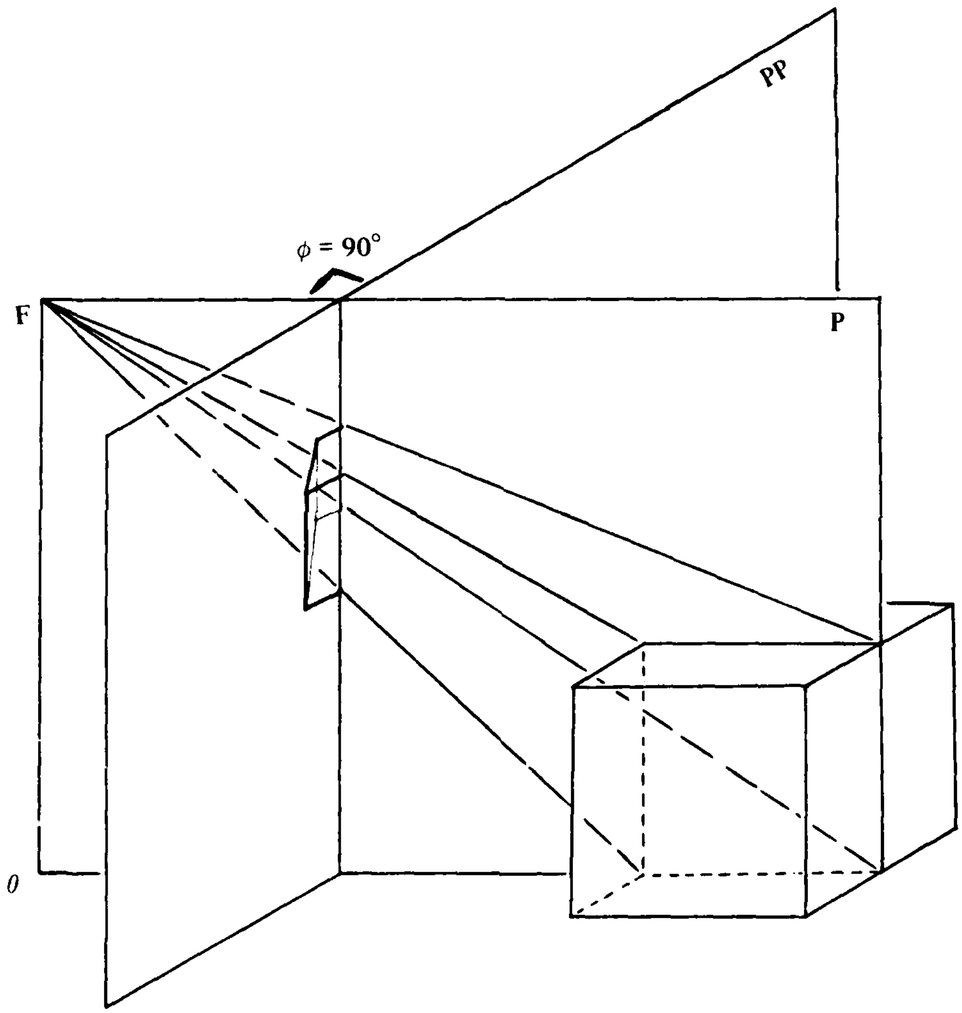


FIGURE 3. Perspective View (With Emphasis on Perpendicular Projecting Plane).

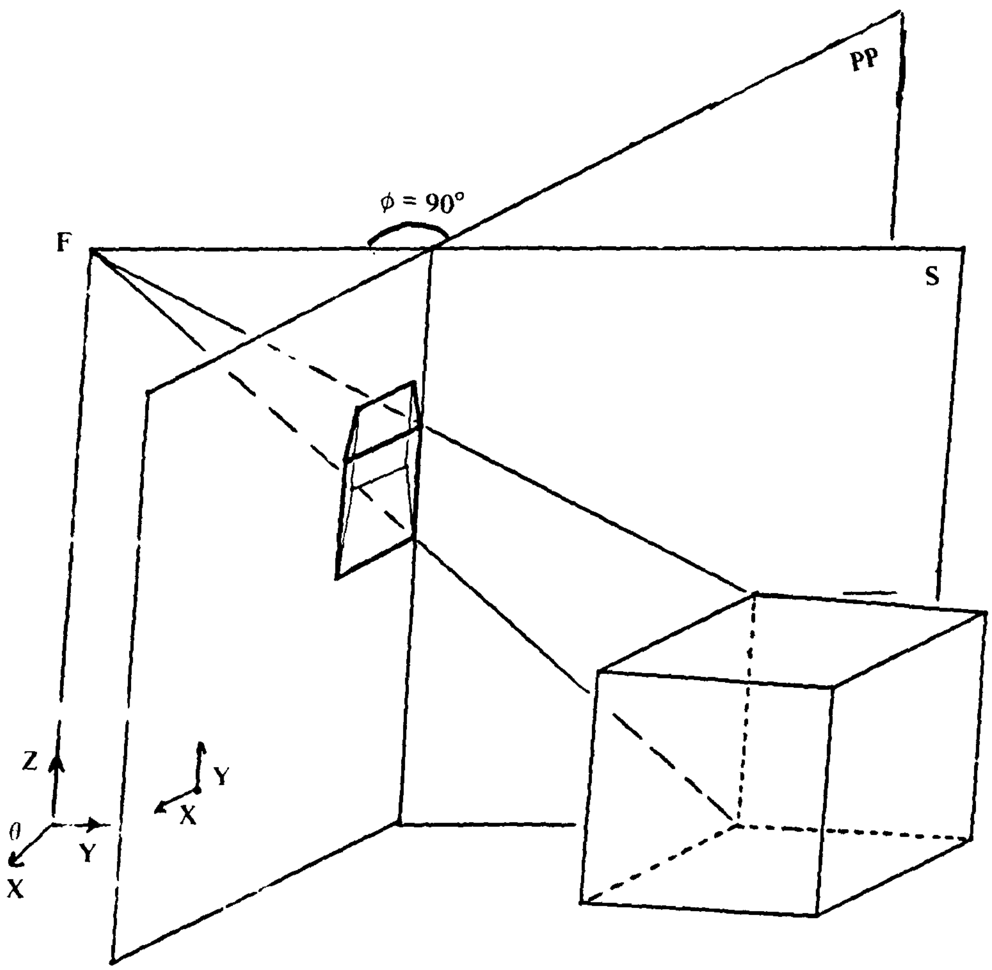


FIGURE 4. Perspective view (With Emphasis on Oblique Projecting Plane).

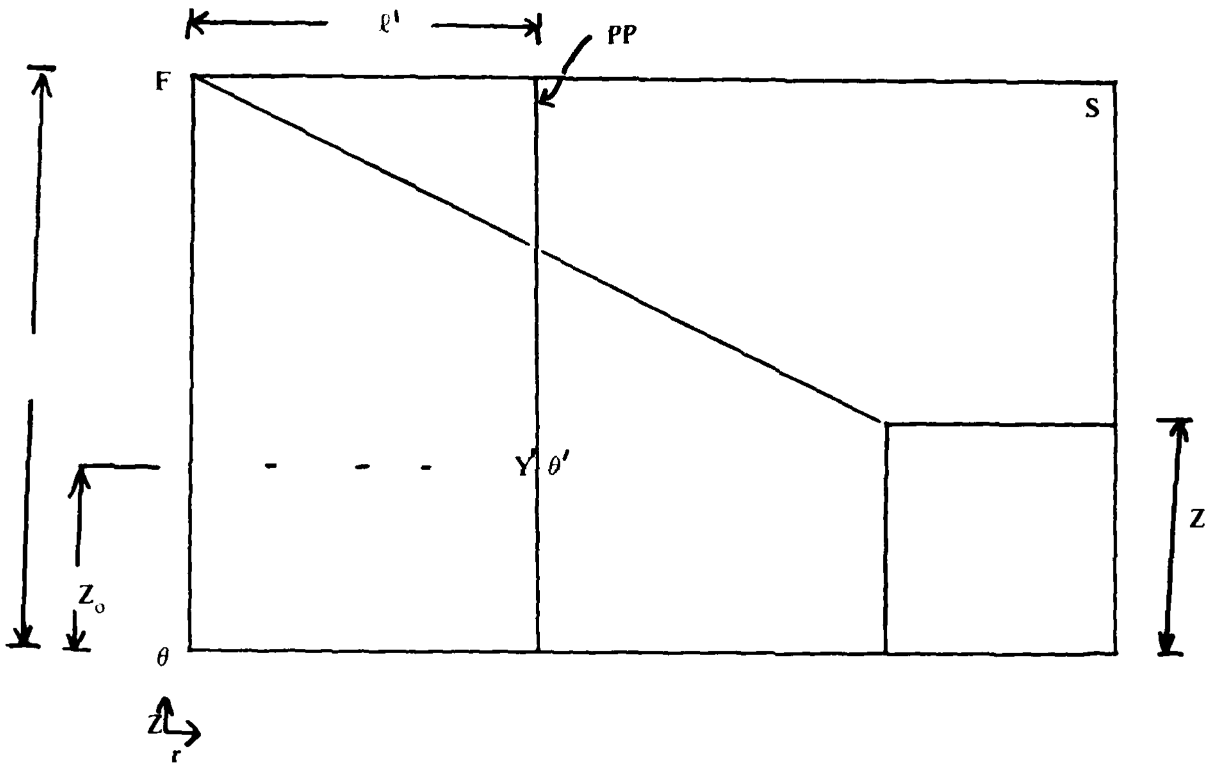


FIGURE 5. The Y Coordinate Determination in Perspective View.

Now, determine the Y' coordinate. Examine the vertical plane S, containing the point Q being projected and the focal point F (see figure 6). Again, simple geometry determines the transformed coordinate. As shown in figure 6, the radial variable $r = \sqrt{(X^2 + Y^2)}$ and the distance $\ell' = \ell/\cos\theta$, which is the distance from the line F − Q along the plane S to the plane PP, as introduced. The coordinate Y' is thus

$$Y = (Z - h)\,\ell/r + h - Z_o \qquad (9)$$

Equations 8 and 9 are the basic transformation equations. As with the orthonormal projection, some scale factors may be desirable. Here, two such scale factors will prove valuable: an image scaling factor β, and a vertical exaggeration γ. Equation 8 and 9 thus become

$$X' = -X_o + \beta\ell\,(x/y) \qquad (10)$$

$$Y' = Y_o + \beta\,(\ell/\cos\theta)\,(h - \gamma Z)/r \qquad (11)$$

where we let $Y_o = h - Z_o$.

Equations 10 and 11 are the defining equations for a perspective projection. A generalization of these equations, incorporating a projection plane at an oblique angle to the vertical axis of the object coordinate system, is possible. Such a generalization is of little interest for most proposed applications, and, in any case, can be easily accomplished by an appropriate rotation of the object coordinate system.

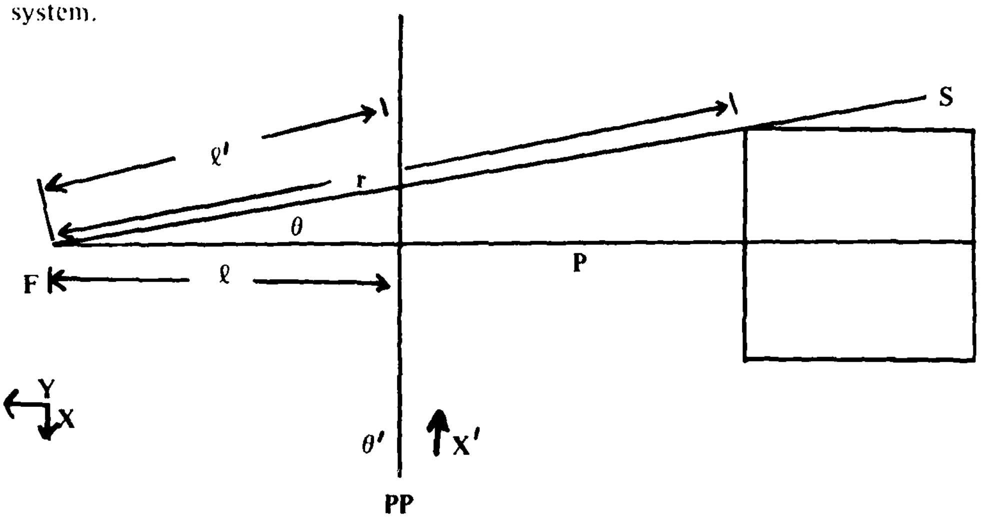


FIGURE 6. The X Coordinate Determination in Perspective View.

B. PHOTOMETRIC VARIABLES. In defining the fundamental photometric variables, the ones of greatest importance in the rest of the report are brightness, illumination, and luminous intensity. Although the variables represent concepts that are quite similar to the colloquial meanings of the terms, a detailed understanding of the terms is needed. In defining the terms, it is assumed that the concept of energy transport by electromagnetic radiation is well understood. This concept, together with elementary geometric considerations, will be used to define the needed variables.

It should be noted that the discussion of the elementary photometric concepts is given to provide a background adequate for the analysis of terrain brightness and image illumination. In this introduction, the concepts are developed in a logical and concise manner, but the approach is not intended to be detailed or exhaustive. The reader interested in a more detailed treatment is referred to texts such as Walsh.[5]

Photometry is the branch of optics concerned with the measurement of light. Although it is not, strictly speaking, a part of geometric optics, many practical applications are such that the geometric approximation is reasonable. The present work is such an application. Hence, we shall adopt the geometrical model of light by which light is regarded as the flow of luminous energy along geometric rays.[6] This flow is subject to the geometric law of conservation of energy, which requires that the energy transmitted in unit time across a section of a bundle of rays is constant. This definition will be used implicity. Consider the radiant energy emerging from a portion of some surface, Σ. This surface may be fictitious, the surface of a self-radiating (primary) source, or it may be an illuminated surface of a solid (a secondary source).

The time-averaged transfer of energy by a ray of light is defined by a vector

$$S = \frac{C}{4\pi} \vec{E} \times \vec{H} \qquad (12)$$

---

[5] J. Walsh, *Photometry*, Third Ed., Dover, 1958.

[6] M. Born and E. Wolf, *Principles of Optics*, Fourth Ed., Pergammon Press, 1970, pp. 115–116.

Where the Gaussian notation has been used, C is the speed of light, $\vec{E}$ is the electric field vector, and $\vec{H}$ is the magnetic vector of the electromagnetic wave.[7] It is sufficient to note that the vector $\vec{S}$, known as *Poyntings* vector, defines the direction of and represents the amount of energy that crosses a unit area perpendicular to the direction in which the ray is traveling per unit time.

Therefore, $\vec{S}$ may be interpreted as the density and direction of energy flow in a beam of electromagnetic radiation. The quantity with which we are primarily concerned at this point is the total energy per second that crosses a given surface. This quantity is known as the *flux* of radiant energy (i.e., the total energy per second) crossing the surface. The flux is defined by

$$F = \int_{surface} \langle \vec{S} \rangle \cdot n \, dA \tag{13}$$

where $\hat{n}$ is the outward unit normal to the surface, dA is an infinite*simal* area element of the surface, and $\langle \vec{S} \rangle$ is the time average of the *Poyntings* vector.[8] It will be recalled that the dot product of two vectors is

$$\vec{A} \cdot \vec{B} = |\vec{A}||\vec{B}| \cos\theta \tag{14}$$

where θ is the angle between the two vectors. Thus, substituting equation (14) in equation (13), we find that

$$F = \int_{surface} |S||\hat{n}| \cos\theta \, dA \tag{15}$$

or, since the normal vector was defined as the unit normal vector, $|\hat{n}| = 1$, and

$$F = \int_{surface} |S| \cos\theta \, dA \tag{16}$$

[7] M. Born and E. Wolf. *Principles of Optics,* Fourth Ed., Pergammon Press, 1970, pp. 115–116.

[8] R. Longhurst. *Geometrical and Physical Optics,* Second Ed., Longmann, Green and Co., 1967, p. 434.

The above discussion assumes that the light with which we are concerned is traveling in a single, specific direction and that it has no divergence. However, this is an ideal situation, which does not exist in physical reality. Instead, the light traversing our surface will be composed of one or more rays of light occupying an infinitesimal solid angle. In general, for any direction defined by the polar angles (α, β), there will be defined some differential *Poynting* vector (differential because it occupies a differential solid angle). Thus, in general, we have the differential *Poynting* vector in any given direction (α, β), defined by an arbitrary function $B(\alpha, \beta)$, such that

$$d\vec{S}(\alpha, \beta) \cdot B(\alpha, \beta)\, d\Omega \tag{17}$$

where we assume that $B(\alpha, \beta)$ has a unique, (time-averaged) value for all (α, β).

To determine the flux of radiant energy across a differential element of our surface, one must integrate over all possible infinitestimal cones of light. Thus, one integrates with respect to the differential *Poynting* vectors, or equivalently, with respect to the differential solid angle. For the differential flux across a differential surface element with unit normal $\hat{n}$,

$$\begin{aligned} dF &= \int_{\alpha,\beta} d\vec{S}(\alpha, \beta) \cdot \hat{n}\, dA \\ &= \int_{\alpha,\beta} B(\alpha, \beta) \cos\theta\,(\alpha, \beta)\, d\Omega\, dA \end{aligned} \tag{18}$$

where the dependence of θ on α and β has been explicity noted.

The determination of the total flux of radiant energy across our surface requires integration over the differential surface element. It should be obvious that our differential *Poynting* function $S(\alpha, \beta)$ will also, in general, be a function of position. Thus, the total flux across some surface **A** with local coordinates (ξ, η), and polar angles (α, β) will be

$$\begin{aligned} F &= \int_{\xi,\eta} F(\xi, \eta)\, dA \\ &= \int_{\xi,\eta} \int_{\alpha,\beta} B(\alpha,\beta;\xi,\eta) \operatorname{Cos}\theta\,(\alpha,\beta;\xi,\eta)\, d\Omega\, dA \end{aligned} \tag{19}$$

Note the generalization that has been made. Since the differential flux incident on our surface will, in general, vary from point to point, so will the defining function B. Similarly, since our surface may be curved, the value of the angle θ between the normal to the surface and some vector defined by the polar angles (α, β) will also be a function of position. This additional functional dependence has been made explicit in equations (20) and (21), with the two types of dependence differentiated by semicolons.

Let us now interpret these mathematical results in terms of photometrically defined variables. The only variable entering into the mathematical definition of the flux that is arbitrary is $B(\alpha, \beta)$. This variable is the photometric brightness,[9] which is the photometric quantity that will be of concern throughout the rest of this report. It is analagous to the familiar concept of brightness, differing from it only in that (1) the efficiency of the human eye is a function of the intensity of the incident light, and (2) the efficiency of the human eye is a function of the frequency of the light.

Brightness is the most important photometric variable for our purposes, since it is independent of distance. Qualitatively, this can be seen by considering a luminous surface element $d\Sigma$ at a distance $|\vec{r}|$ from the surface element of a photodetector, da. We know from elementary physics that the flux upon the detector area element obeys the inverse square law. Noting that the solid angle subtended by the surface element also obeys the inverse square law and applying the differential form of equation (19), one can see that brightness is independent of the distance of the observer:

$$B = dF/(dA \cos\theta \; d\Omega) \tag{20}$$

Explicity writing the inverse square dependence of dF as

$$dF = dF_0/|\vec{r}|^2 \tag{21}$$

and of $d\Omega$ as

$$d\Omega_0 = d\Omega_0 \; |\vec{r}|^2 \tag{22}$$

[9] R. Longhurst, *Geometrical and Physical Optics*, Second Ed., Longmann, Green and Co., 1967, p. 434.

where we have assumed all other quantities held constant, one can substitute into (20) and see that

$$B = dF_o/(dA \cos\theta \, d\Omega_o) \tag{23}$$

thus demonstrating the truth of our claim that brightness is independent of distance.

After having defined flux and photometric brightness, two other photometric quantities can be defined that will prove useful. Rewriting equation (20), we have the basic photometric equation of

$$dF = B dA \cos\theta \, d\Omega \tag{24}$$

where the quantities are as defined above.

Consider a small luminous surface element, dA, as shown in figure 7. This luminous surface does not necessarily radiate equally in all directions. To define the intensity of the element's radiation in a given direction, we define *luminous intensity*, I, as the flux emitted per unit solid angle in a given direction.[10] Mathematically, this is written as

$$dI = dF/d\Omega = B \cos\theta \, dA \tag{25}$$

where dI is the differential luminous intensity of the surface element in a direction inclined at an angle θ to the normal to the surface element.

Now consider another surface element, dA, which is illuminated: i.e. there is a nonzero flux of radiation upon the surface (see figure 8). Then the incident flux per unit area upon the surface element is defined as the *illumination* of the element dA.[11] Thus, if we consider the illumination of the surface element by a source of illumination positioned at an angle θ to the normal to the surface element, the differential illumination, dI, of the surface element may be written as

$$dE = dF/dA = B \cos\theta \, d\Omega \tag{26}$$

These definitions complete the introduction to the basic photometric variables and will suffice for the succeeding discussions of light scattering by surfaces and of image photometry.

---

[10] M. Born and E. Wolf, *Principles of Optics*, Fourth Ed., Pergamon Press, 1970, p. 182.

[11] Ibid.

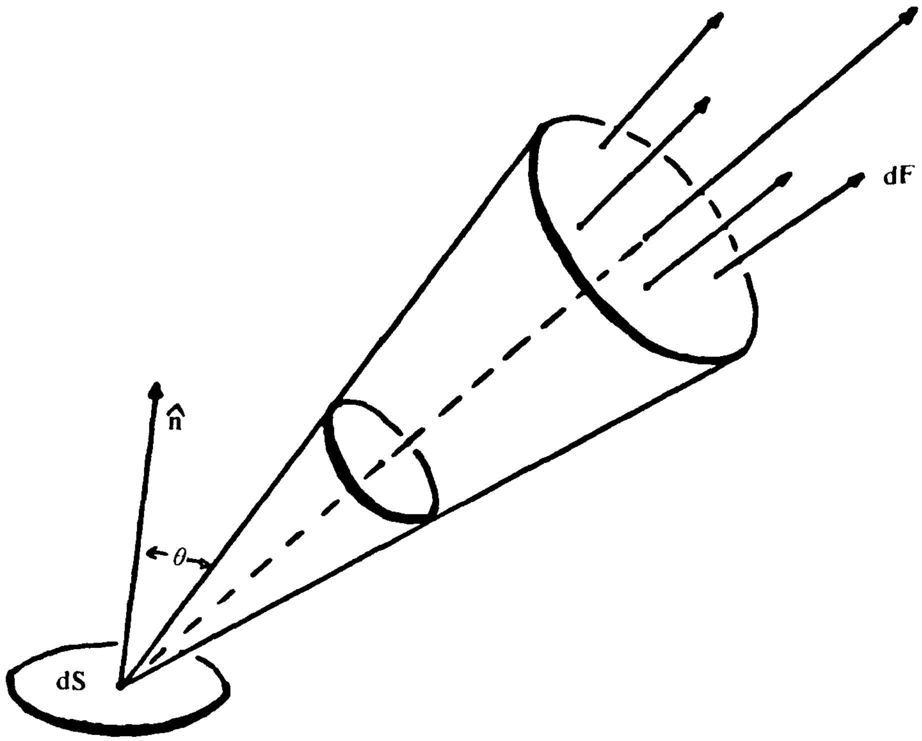


FIGURE 7. Luminous Intensity.

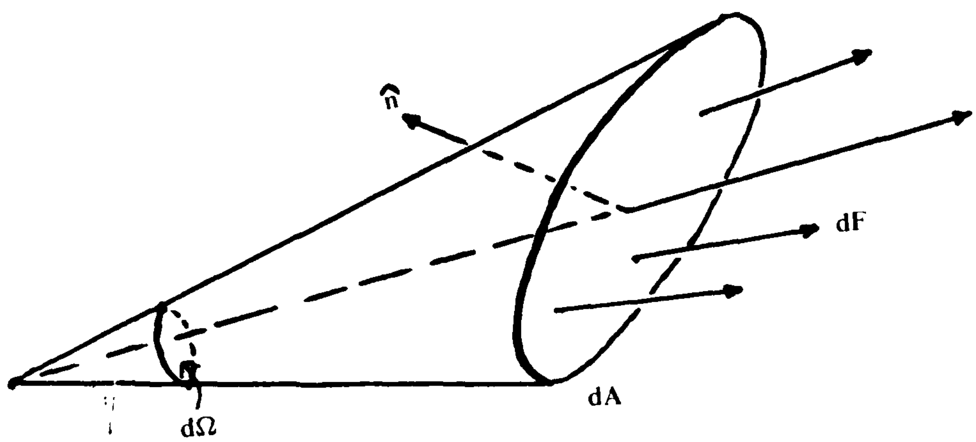


FIGURE 8. Illumination.

C. **LIGHT–SCATTERING.** The description of the scattering of light by a terrain surface is a central component of the theory of shaded relief images. It was shown previously that photometry provides the appropriate language for this description. Within the context of this language, one variable, surface brightness, was shown to embody all information of interest concerning light emerging from any surface. The discussion of photometry, however, left "brightness" as an abstract and arbitrary function. Thus, the meaning of "brightness" must be defined as it relates to light-scattering by surfaces. This is done by examining the two specific light-scattering functions selected for use in the ETL software.

The light-scattering function chosen is really the heart of any synthetic image. This formula determines how the terrain model is shaded. This shading of the terrain model is, in turn, the principal guide to the form of the terrain available to a viewer of a synthetic image. Such an observer will interpret the image on the basis of a number of poorly defined preconceptions based on years of visual experience. Thus, the light-scattering function chosen will, in a poorly understood manner, define the apparent relief of any synthetic image generated.

Two criteria were used to select the two light-scattering functions used at ETL:

1. Simplicity of mathematical form.
2. Extensive empirical justification.

The first is essential if the chosen law is to be used in efficient software to generate shaded relief images. The second requires an extensive collection of familiar physical objects that are well described by the chosen functions. Any such formula should therefore satisfy the preconceptions of an observer interpreting the resulting image.

From the criteria, a deficiency exists in understanding the scattering of light: empirical results can rarely be expressed in terms of simple functions; however, theoretical formulate rarely provide a good description of actual surfaces. The two light-scattering functions chosen by ETL are complementary attempts to meet both criteria. Thus, Lambert's law accurately describes the light-scattering behaviour of many common surfaces, and it can be simply expressed. However, no entirely satisfactory derivation of it has yet been found. The Lommel-Seeliger law, on the other hand, was first derived from theoretical considerations. It is thus of simple mathematical form, but it is not broadly applicable. Empirical justification of both laws is presented below and discussion of the theory of each is included in appendix 1.

The discussion below presents both Lambert's law and the Lommel-Seeliger law as functions describing the brightness of a terrain surface element. These functions depend on the relative orientation of the surface element to the source of illumination, and in the case of the Lommel-Seeliger law, upon the inclination of the surface to the observer. To use this formulation, one needs only to calculate the flux on a photosensitive surface. These results can then be qualitatively interpreted, emphasizing the characteristic properties of each.

1. Lambert's Law. The first of the light-scattering functions chosen for implementation in the ETL software was Lambert's law. Lambert's law describes the interaction of light with a surface that, by definition, is perfectly diffuse. In contrast with a perfectly reflecting surface, which is infinitely bright in the direction of reflection and totally dark elsewhere, a perfectly diffuse surface is equally bright, regardless of the direction from which it is viewed. However, this brightness is not independent of the source of illumination. This follows from the requirement that no more light can be reflected from a surface than is incident on it. Since the flux on a surface element will decline as the cosine of the angle i between the direction of propagation of the incident light and the normal to the surface, Lambert's law may thus be written[12]

$$B = B_0 \cos i \qquad (27)$$

The B_0 term is the maximum brightness of the surface and will depend upon the total reflectivity of the terrain and upon the illumination of the surface.

[12] J. Walsh, *Photometry*, Third Ed., Dover, 1958, p. 137.

Many natural surfaces, which are of cartographic interest, obey Lambert's law to a high degree of accuracy. The surface of any body composed of discrete particles that are translucent will obey Lambert's law to a good approximation.[13] Thus, the law accurately describes the light-scattering properties of natural surfaces such as snow, sand, pumice, hoarfrost, and some vegetation. Lambert's law also accurately describes the diffusing properties of a surface that is rough or corrugated and that is composed of discrete low-albedo particles, each of which scatters light equally in all directions.[14] Most rocks can be accurately modeled as such, and Lambert's law is thus applicable to them. This extensive collection of natural materials obeying Lambert's law suggests that, in addition to serving as a theoretical photometric standard, the law may also serve as a natural standard for diffuse surfaces for most people.

The two models given above for surfaces obeying Lambert's law are empirical in nature and cannot be justified in a rigorous theoretical manner.[15] This is not a significant drawback for cartographic purposes, since Lambert's law is both empirical in nature and a simple mathematical form. However, the lack of theoretical justification does indicate a deficiency in the current understanding of the scattering of light. Appendix I includes, in a discussion parallel to the derivation of the Lommel-Seeliger law, a brief mathematical discussion of the first model of Lambert's law, indicating the deficiency of this model.

Applying Lambert's law to calculations of flux is straightforward, but illustrates several features of interest. The features can be demonstrated by examining a simple device for measuring the flux from a surface. The device consists of a photosensitive surface of area a, which through the use of apertures or some other system, has a light acceptance cone subtending a solid angle of $d\omega$ (see figure 9). This photodetector is situated a distance z above a surface obeying Lambert's law, and the detector acceptance cone is inclined at an angle θ to the normal to the surface. Consequently, the distance from the surface area within the detector acceptance cone to the detector is $r = z \sec \theta$. The surface element within the detector acceptance cone is thus of area

$$dA = r^2 \, d\omega / \cos\theta \qquad (28)$$

---

[13] B. Hapke and H. Van Horn, *J. Geophys. Res.*, 68(1963), p. 4552.

[14] Ibid., p. 4553.

[15] P. Beckmann, *Scattering of Light by Rough Surface* in *Progress in Optics*, V. VI, Ed. by E. Wolf, North-Holland, 1967, p. 57.

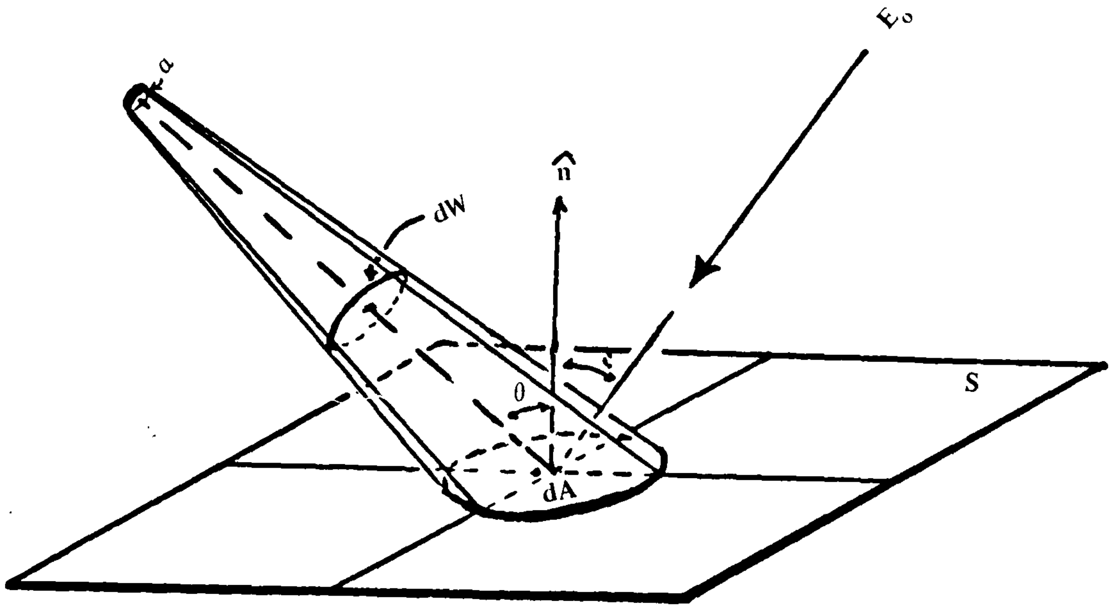


FIGURE 9. Measurement of Luminous Flux.

From the surface, the photodetector subtends a solid angle of

$$d\Omega = a/r^2 \tag{29}$$

If one uses the basic photometric identity, equation (24), the flux upon the detector is equal to the fraction of the light from each point of the surface that falls upon the detector ($d\Omega$), times the effective area of the surface element ($dA \cos\theta$), times the brightness of the surface:

$$\begin{aligned} F &= B\, dA \cos\theta\, d\Omega \\ &= B_o \cos\theta\, (r^2\, d\omega / \cos\theta)\, \cos\theta\, (a/r^2) \\ &= B_o \cos\theta\, d\omega \end{aligned} \tag{30}$$

Several implications of this equation should be noted. First, if one assumes that the surface is uniform and of infinite extent, the flux upon the detector is independent of the viewing angle θ. Second, if one makes the same assumptions, the flux is independent of distance. The first of these is of greater general significance, since for surface elements of apparent area $dA = dA_o/\cos\theta$, it is also true. As the observer moves farther away, the flux will decline as $1/r^2$, but the independence of observation angle is still true. This property is discussed in the section on image photometry.

The implications of this discussion for the generation of synthetic images is clear. Slopes facing the source of illumination will be bright, slopes facing away will be dark, and flat areas or slopes parallel to the source of illumination will be neutral. The precise nature of the shading should resemble that of actual terrain.

2. **The Lommel-Seeliger Law.** The second light-scattering function chosen for implementation in the ETL software was the Lommel-Seeliger law. Theoretical in origin, the function is based on a simplified model of a low-albedo scattering body. A slightly more general treatment leads to a widely applicable light-scattering

function, which, however, cannot be evaluated in closed form in terms of known functions.[16] The simpler form of the Lommel-Seeliger law thus provides clear advantages for any simulation and, as discussed below, provides an approximate description of the light-scattering properties of a variety of natural surfaces. The principle feature of the Lommel-Seeliger law that differentiates it from Lambert's law is the dependence on viewing angle. This dependence is expressed in the Lommel-Seeliger law:[17]

$$B = B_o/(1 + \cos\theta/\cos i) \tag{31}$$

where B_o, i, and θ are as defined in the description of Lambert's law.

As noted, the Lommel-Seeliger law is derived from a simple model of low albedo objects (see appendix 1). The principle criticism that this model is open to is that it neglects the porous structure of many such natural bodies.[18] This neglect means that the pronounced backscatter common to many real low-albedo scatterers is not modeled by the Lommel-Seeliger law. It should be noted that the degree of this backscatter is an independent parameter in the detailed model. Thus, the Lommel-Seeliger law may be considered as the limit of a more general model as the porous structure of that model is reduced to insignificance. An additional parameter in both the simplified and general models should be noted: the scattering law of the individual particles composing the body is also an integral part of the general law. This additional term is relatively insignificant, however, since both models are rather insensitive to the term's precise form, for moderate valves of the illumination and viewing angles.[19]

Empirical justification of the Lommel-Seeliger law is essential if the second criteria for its selection is to be met. Just as the general model of low-albedo scattering bodies includes a parameter governing the significance of the porous structure, physical scatterers seem to embody such a parameter.[20] Thus, the significance

---

[16] B. Hapke. *J. Geophys. Res.*, 68(1963), p. 4575.

[17] Ibid., p. 4573.

[18] Ibid., p. 4573.

[19] Ibid., p. 4577.

[20] B. Hapke and H. Van Horn. *J. Geophys. Res.* 68(1963), p. 4552.

of the backscatter is highly variable, ranging from a sharp peak for some vegetation, rocks, and the moon to broad, nearly insignificant peaks for other materials. Since in any application, the form of the light-scattering law of the particles of which the body is composed must be somewhat arbitrarily chosen, as does the structure parameter, there is at present no good empirical reason for not choosing the limiting case of the Lommel-Seeliger law. Since it is simple, the Lommel–Seeliger law was chosen for use at FTL, as well as for at least one previous study.[21]

The application of the Lommel-Seeliger formula for terrain brightness is quite similar to that for Lambert's law. If we model the same device introduced in the discussion of Lambert's law, the determination of the total flux incident upon a photodetector of area a is straightforward (see figure 9). Situated a distance r from the surface, with a detector acceptance cone subtending a solid angle $d\omega$ at an angle θ to the normal to the surface, the detector will accept light from a differential area of $dA = r^2 \, d\omega / \cos\theta$. Since the detector subtends solid angle of $d\Omega = a/r^2$ at the surface, the total flux on the detector is

$$F = B \, d\Omega \cos\theta \, dA \tag{32}$$

B is a function of i and θ, as defined above. Hence,

$$= \frac{B_o}{1 + \cos\theta / \cos i} \cos\theta \, dA \, d\Omega$$

$$\frac{B_o \, d\omega \, a}{1 + \cos\theta / \cos i} \tag{33}$$

This result is substantially different from the corresponding result for Lambert's law. It is dependent on both the angle of illumination and the angle of observation. The formula does meet the elementary esthetic criteria mentioned in the discussion of Lamber's law: slopes facing away from the source of illumination are dark, and slopes facing the source of illumination are bright. The dependence on θ is less intuitive, since the brightness of the terrain reaches a minimum for an orthonormal view, and a maximum for high $(\theta \simeq 90°)$ values of θ, all else held constant. This can be readily understood within the context of the model from which the Lommel-Seeliger law is derived, and accurately describes the behavior of many physical objects. Thus, any discussion of the ease of interpretation of an image generated using the Lommel-Seeliger law (or any other must be based on such images, and not on the relative complexity of the light-scattering law.

---

[21] R. Batson, K. Edwards, and E. Eliason, *J. Res. U.S. Geol. Surv.*, 3(1975), p. 401.

D. IMAGE PHOTOMETRY. The remaining theoretical problem to be addressed is that of recording, in a permanent image, the important properties of the light reflected by the terrain. There are two aspects of this problem. First, what quantity should be used to define the light reflected by the terrain? Second, what quantity is a real recording medium, such as photographic film or the human eye, sensitive to? The solution to the problem will be a mathematical formula relating these two quantities for a particular imaging system.

We saw in our discussion of photometry that the brightness, B, of a surface is a quantity that is independent of distance. All other basic photometric variables, such as flux, luminous intensity, and illumination, can be expressed in terms of the brightness and the variables that describe the geometry of a particular situation. Further, light-scattering laws that define the interaction of the terrain surface with the incident light may easily be cast in a form defining the brightness of the terrain. Thus, the brightness will be used as the basic quantity defining the important properties of the interaction of light with the terrain surface.

The other basic quantity to which a recording medium is sensitive to can be chosen from the photographic theory. The density of a photographic image is, in general, a complicated function of the illumination as a function of time.[22] In normal photochemical reactions at light levels, such as those which we seek to simulate, the amount of product per unit area that is formed is directly proportional to the product of the illumination of the area and the time of illumination.[23] By assuming a unit time for all exposures, we may therefore use the illumination of a given point of an image as the important property of the light incident on that point.

First, the problem for the case of perspective imaging systems, such as a camera, will be discussed. The formula derived is more exact than is desirable for our purposes. Consequently, several approximations will be introduced that are reasonable for the cases of interest. The approximations will be used for both the perspective and orthonormal projections.

---

[22] J. Walsh. *Photometry*, Third Ed., Dover, 1958, p. 435.

[23] H. Baines. *The Photographic Process* in *Photography for the Scientist*, Ed. by C. Engel, Academic Press, 1968, p. 7.

As before, the variables must be defined (see figure 10). An imaging system, such as a camera, is composed of a lens of aperture a with a light-sensitive surface parallel to the lens a distance $\mathbf{r}_0$ behind it. This system is imaging a luminous surface dS located a distance $\mathbf{r}$ from the center of the lens, at an angle θ to the central axis of the optical system. The normal to the luminous surface $\hat{\mathbf{n}}$ is oriented at an angle φ to the ray connecting dS to the center of the camera lens (assuming $\hat{\mathbf{n}}$ to be coplanar with $\mathbf{r}$ for simplicity).

The flux emitted by the luminous surface into a solid angle $d\omega$, oriented at an angle φ to the normal to the surface, is given by

$$dF = \beta \cos\varphi \, d\omega \, dS \tag{34}$$

Now, for the case of the camera-object system described above,

$$d\omega = \frac{da \cos\theta}{r^2} \tag{35}$$

and hence,

$$dF = \frac{\beta \cos\varphi \, dS \, da \cos\theta}{r^2} \tag{36}$$

Integrating over the area of the camera lens, we thus see that the total flux incident upon the camera lens is given by

$$F = \frac{\beta \cos\varphi \, dS \, a \cos\theta}{r^2} \tag{37}$$

If we assume that all the light incident on the lens from the surface element dS passing through the lens is concentrated upon the corresponding image element dS! and that a factor of $(1 - \gamma)$ of the incident flux is scattered or absorbed during passage through the lens, then the illumination of the image element is given by

$$E = \frac{dS}{dS'} \frac{dF}{dS} (1 - \gamma) \tag{38}$$

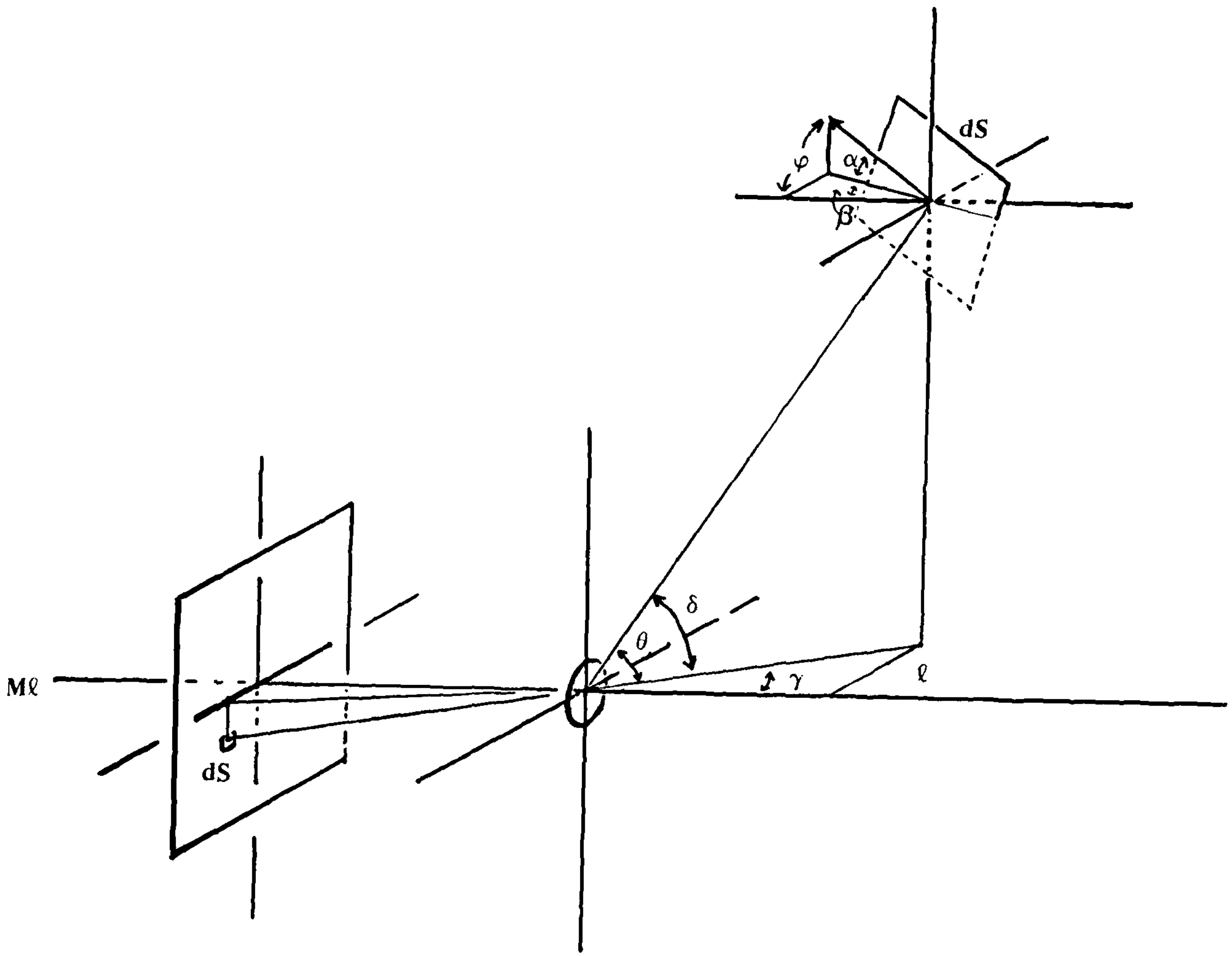


FIGURE 10. Geometry of Image Formation.

The factor of dS/dS' must now be evaluated. Our surface element dS is positioned such that the unit normal may be characterized by a polar angle α and an azimuthal angle β. Similarly the ray will be defined that connects the center of the camera lens to dS by a polar angle δ and an azimuthal angle γ. First, determine the angle θ that the radial vector makes with the central axis of the optical system. Since the radial vector is treated as a unit vector, the x-component of the vector is

$$r_x = \cos\delta \cos\gamma \tag{39}$$

Noting the definition of θ, one can see that it is defined by

$$\theta = \cos^{-1} ({}^{r_x}/r) = \cos^{-1} (\cos\delta \cos\gamma) \tag{40}$$

Now, in an ideal imaging system, linear features of an object parallel to the image plane are transformed onto linear features in the image plane by a factor of proportionality known as the lateral magnification, **M**.[24] Area elements are similarly transformed by a factor of $\mathbf{M}^2$. As defined, $\mathbf{M} = dS_o/dS'$ is inversely proportional to the radius r. Features oriented to dS_o, as dS is in the figure, will be transformed as their projection onto dS_o, which is equal to the dot product of the respective normals. Thus

$$\begin{aligned} dS_o &= \hat{n}\,(dS_o) \cdot \hat{n}\,(dS)\; dS \\ &= (-1, 0, 0) \cdot (^-\cos\alpha \cos\beta, \cos\alpha \sin\beta, \sin\alpha) dS \\ &= \cos\alpha \cos\beta \; dS \end{aligned} \tag{41}$$

Hence,

$$\begin{aligned} dS/dS &= dS/dS_o \quad dS_o/dS \\ &= \frac{1}{\cos\alpha \cos\beta} \quad \frac{1}{M^2} \\ &= \frac{r^2}{M_o^2} \quad \frac{1}{\cos\alpha \cos\beta} \end{aligned} \tag{42}$$

[24] G. Franke, *Physical Optics in Photography*, Focal Press, 1966, p. 12.

where M_o is the magnification for some nominal distance r_o. Substituting (42) into (39) and (40). One has the illumination on dS' as

$$E = \frac{r^2}{M_o^2 \cos\alpha \cos\beta} \quad \frac{\alpha\,\beta \cos\varphi \cos\theta}{r^2} \quad (1-\gamma) dS$$
$$= \frac{\alpha\,\beta \cos\varphi \cos\theta \; dS}{M_o^2 \cos\alpha \cos\beta} \; (1-\gamma) \tag{43}$$

Equation 46 is the rigorous formula for the perspective projection. Note that the illumination falls off rapidly as θ increases. In fact, since the other variables are not independent of θ, the actual dependence, holding all else constant, goes as $\cos^4\theta$.[25] This effect is known as vignetting and results in a bright central image, which darkens rapidly towards the edges. As such, the effect is undersirable. Hence, several approximations shall be considered to eliminate this effect.

First, since

$$\cos\varphi = \hat{n} \cdot r \tag{44}$$

one may write

$$\begin{aligned} \cos\varphi = \; & \cos\alpha \cos\beta \cos\gamma \cos\delta \\ & - \cos\alpha \cos\beta \cos\delta \sin\gamma \\ & - \sin\alpha \sin\delta \end{aligned} \tag{45}$$

For small γ and δ, such as is the case near the center of the image, $\cos\gamma \simeq \cos\delta \simeq 1$ an $\sin\gamma \simeq \sin\delta \simeq 0$. Substituting for this case,

$$E \simeq \frac{\alpha\,\beta \cos\alpha \cos\beta \cos\theta}{M_o^2 \cos\alpha \cos\beta} \; dS\,(1-\gamma) \tag{46}$$

[25] R. Longhurst, *Geometrical and Physical Optics*, Second Ed., Longmann, Green and Co., 1967, p. 41?

Since for small γ and δ, θ is also small, we have

$$E \simeq \frac{q\alpha\,\beta\,dS}{M_o^2} \tag{47}$$

where we have let

$$\gamma = 0 \tag{48}$$

The result, which is a reasonable approximation for the central region of a perspective view, is independent of r and is independent of any of the defining angles. Since the limiting case of the approximations used corresponds to the orthonormal view, the result is also valid for the orthonormal projection of gray shade data. Thus, for both projections of interest in this report, a formula now exists that relates the brightness of a surface element to the illumination of the corresponding point of the image. Since the illumination defines the density of an actual image, recording the illumination as a function of position stores the important information of the image needed for simulation.

III. ALGORITHMS

In the previous section of this report, the theoretical basis was examined of the creation of shaded relief images for cartographic applications. This theoretical basis was divided into four independent components, each of which was examined in detail. Now, the task of assembling the components of the theory within a unified algorithm for the generation of shaded relief images, must be undertaken. The algorithms must be understood before use of the software developed in undertaken, which is essential before any major modifications of the software created are attempted.

First, some notes should be made regarding the nature of the model embodied in the algorithms described below. In nature, the process of image formation may be described as a continous process, at least at the level at which we seek to simulate it. Every visible point on the object contributes light to a single point on the image plane. Mathematically, this process could be defined by a function, a 1:1 mapping of visible points on the object onto the points of the image plane. The image function will be determined by the geometry of the imaging system, the geometric model of light, and the geometries of the object and image spaces. The continuous nature of the process is clearly defined. Although discontinuties may exist in the visible object space of the perspective projection, as when a valley dips out of sight, every point contributing to the image has another such point infinitesimally far away from it. Correspondingly, the image plane is the basis for a continuous image. Thus, no inherently discrete aspects to the process of image formation exists.

This continuity is to be contrasted with the inherently discrete nature of any digital simulation. The image plane in such a simulation of the imaging process must be modeled as an array of discrete pixels, each of which can assume one of a set of discrete brightness levels. This discrete nature of the image space means that an inverse function mapping the image space into the object space is needed, at least implicitly. Such a function cannot, in general, be analytically determined for the perspective projection because of the complicated nature of the phenomena being modeled. The inverse function must therefore be algorithmically determined by a series of successive approximations for each pixel of the image. Similar approximations must be developed for each element for the theory embodied in the model.

The guiding principles in the development of the algorithms for the generation of shaded relief images must, for our purposes, be efficient and versatile when implemented in the form of computer software. Since each calculation involved in an algorithm takes a finite amount of time, severe constraints are placed on the form of any algorithm developed, and hence on the nature of the approximate solutions used. Thus, these constraints help to determine the specific content of any algorithms developed.

The discussion of the algorithms embodying the various aspects of the theory proceeds in a manner reflecting the role of each algorithm within the ETL software. The structure of this software for perspective and orthonormal images is quite similar. As can be seen, there are four major computational components of the software, each embodying a corresponding theoretical component. Thus, there are four conceptual divisions to the unified algorithms developed.

These parameters are determined once for each image, and they define the particular characteristics of the shaded relief image to be generated. The following four components must be sequentially done for each element of the image:

1. Define global image parameters (observer position, imaging system characteristics, vector of illumination).
2. Correlate image/object.
3. Determine local surface orientation.
4. Compute surface brightness/image density.

Since the coded digital image must be output in some graphic format, the fifth component is:

5. Image generation.

The discussion of the first point is essentially limited to a brief treatment of the mathematical definition of the vectors of illumination and observation. The vectors are necessary, since together with the orientation of a local surface element, they may be used to define the brightness of that surface element by means of Lambert's law or the Lommel-Seeliger law. Thus, vectors define the manner in which the illumination of the terrain is simulated, knowledge of which is crucial to interprete accurately the image produced. Other global variables that must be defined, such as the focal length of a perspective imaging system, should require no detailed explanation.

Having specified the global variables defining the image to be produced, one can generate a simulated image by performing a series of local calculations for each element of the image. Once a given image element (pixel) is specified, some point must be determined on the surface of the terrain whose projection lies within the pixel. Since an analytic solution to this problem is not available for the perspective case, an algorithm yielding an approximate solution is discussed. This algorithm exploits the projection of a radial elevation profile that lies on a vertical line in the image plane. By sampling the elevation profile corresponding to a given column of pixels, a point on the terrain with the required characteristics can be found by a process of successive approximations. Elevation data points may be readily accessed as needed, a minor problem when the polynomial data base is used. The situation is much simpler in the case of the orthonormal projection, since projection equations (3) and (4) may easily be inverted.

Having found a point on the surface of the terrain corresponding to a given pixel, one may use the point as a representation sample of the pixel. After having found a representation point, the orientation of the surface at the chosen point must be determined. This orientation may be defined by the vector normal to the surface at the point of interest. The determination of the normal is discussed for two types of terrain elevation data bases, (1) uniformly gridded data, and (2) the polynomial terrain data base. Gridded data yields an approximate solution, and the polynomial terrain data base yields an analytic solution.

All that remains at this point is to use a given light-scattering law and the results of the above calculations to determine the illumination of the pixel of interest. Since most commercially available gray shade output devices require input data specifying the density of a photographic image and not the illumination, this conversion is discussed, as implemented in the ETL software.

Once the shaded relief image has been generated and coded in the form of image density values, it must be output to some device capable of producing the graphic product. Three such output devices are available at ETL: (1) an electron beam recorder (EBR) for very high resolution, near-continuous tone images; (2) a line printer for the generation of proofing images; and (3) a Versatec plotter, which is used to generage moderate in resolution halftone images. The requirements of each device are discussed, and the specific algorithms developed to meet these requirements are outlined.

At the end of this section, several algorithms are discussed that are devised to meet specific cartographic problems. These algorithms enable a variable (i.e. non-global) sun angle to be used to generate perspective or orthonormal images, the simulation of atmospheric haze in perspective views, and the orthonormal relief contour images. The variable sun angle algorithm was introduced to accent terrain features that would otherwise wash out because of poor orientation relative to a given source of illumination. This is accomplished by locally varying the azimuth of the source of illumination around a principle source, a common cartographic technique. The result is an enhanced representation of terrain form.[26,27] Atmospheric haze is simulated by a simple model designed to provide additional distance clues to the viewer of a shaded relief perspective image. Finally, a simple algorithm enabling the production of relief contours is discussed. Based on a cartographic product first developed by Kitiro Tanaka,[28] one can combine the resulting image of the quantitative information of contours with a striking visual representation of terrain form. These special purpose algorithms substantially enhance the versatility of the ETL software.

The implementation of the theoretical results of the last section are discussed in a set of algorithms for the computer-generated shaded relief images. Corresponding to each part of the theoretical solution to the problem of shaded relief images is an algorithm embodying that solution, typically in an approximate manner. One section, defining the parameters of the image to be produced, is executed once for any image; the other component algorithms are executed sequentially for each image element. Although these algorithms were developed with the intention of efficiently using the polynomial terrain data base, they are generally applicable to any digital terrain model. Thus, the algorithms serve as the basis for versatile and efficient software to generate a variety ot shaded relief terrain images.

---

[26]P. Yoeli. "Die richtung des licht bei analytischen," *Kartographiche Nachricten Guetersloh* 17(1967), pp. 537–544.

[27]K. Brassel. "A Model for Automatic Hill Shading," *Am. Cart.* 1(1974), pp. 15–27.

[28]K. Tanaka. "The Orthographic Relief Method of Representing Topography on Maps," *Geogr. Rev.* 40(1950), pp. 444-456.

A. DEFINITION OF GLOBAL VARIABLES. Most calculations made in generating a shaded relief image are local in nature. The calculations for a given pixel are independent of the calculations for any other pixel and are independent of any terrain except that which is projected into the pixel of interest. Certain parameters are global in nature and must be defined if the image generated is to appear coherent to a viewer. These parameters govern those aspects of the image that are invariant across it, and they define the characteristics of the imaging system, such as focal length and field of view, the location and direction of view of the imaging system, and the position of the source of illumination.

In discussing the local algorithms below, each of these parameters is treated as a constant, which it is for a given image. The mathematical details of each parameter are introduced as it seems convenient, since any formal discussion at this point would be largely unmotivated. None of the parameters are complicated so that at this point no more is required than to call attention to the role they play in unifying the local calculations.

B. VISIBILITY ALGORITHMS.

1. **Assumption.** As noted, the visibility problem is one of areas. Thus, visibile areas of an object corresponding to the area of a given pixel are sought. The solution to the visibility problem that is used at ETL is based on the observation that any visible point with a projection lying within a given image pixel may be used to approximate the characteristics of that pixel. In essence, this means that the plane tangent to the topographic surface at any such visible point may be used to model the surface over the visible region of the object corresponding to the given pixel (see appendix A). Although this may prove somewhat arbitrary, as when terrain dips out of view and then reappears, any refinement of the assumption will require global knowledge of the object. As a practical matter, for images of sufficiently high resolution, the assumption is quite reasonable.

2. **Algorithmic Solutions.** Having made the assumption, one reduces the visibility problem to determining a point (X, Y) on the terrain surface $z(x, y)$, subject to the requirement that the projection of this point, (X', Y'), lies within an image pixel defined by image coordinates $(x \pm \Delta x,\ y \pm \Delta y)$. As noted in the introduction, two cases are of interest, the perspective and orthonormal projections.

a. Orthonormal Projection. The orthonormal projection of a terrain surface $z(x, y)$ is defined by the orthonormal projection euqations (3) and (4). Since z is a function of (x, y) and since the projection equations are independent of z, there is no visibility problem for the case of the orthonormal projection. One, and only one, point of the surface is projected onto a given point of the image. Given the (X', Y') coordinates of a pixel of interest and the scale factor defining the projection, by inverting the projection equations, a point on the surface is defined that meets the acceptance criteria. This point is given by

$$X = \alpha X' \tag{49}$$

$$Y = \alpha Y' \tag{50}$$

Thus, for the case of the orthonormal projection of a terrain data base, the problem of finding points on the surface that may be used to define an image is relatively simple. Once such a point is found corresponding to a given image element, processing proceeds as outlined below.

b. Perspective Projection. The perspective projection of a topographic surface $z(x, y)$ is defined by the perspective projection equations (10) and (11). These projection equations are not independent of z, and portions of the terrain surface may be obscured by other areas of the surface. The first observation precludes a simple analytic object/image correlation algorithm, such as is available with the orthonormal projection, since inversion of the perspective equations requires knowledge of the surface $z(x, y)$. The second observation indicates that not any point with a projection lying within a given pixel is acceptable, for it may be obscured by terrain in the foreground. Thus, in considering whether or not a given point is acceptable, some knowledge of the terrain between i and the observer is required.

Since an analytic solution to the visibility problem is not possible, the ETL software addresses this problem algorithmically. This algorithm is an iterative procedure for finding a visible point on the terrain surface with a projection lying within specified bounds of a given point on the image. This is illustrated in figure 11. In this figure, any surface point within the hatched region is acceptable, in that

1. It will be visible.
2. Its projection lies within a tolerance Δy of the prescribed point YS.

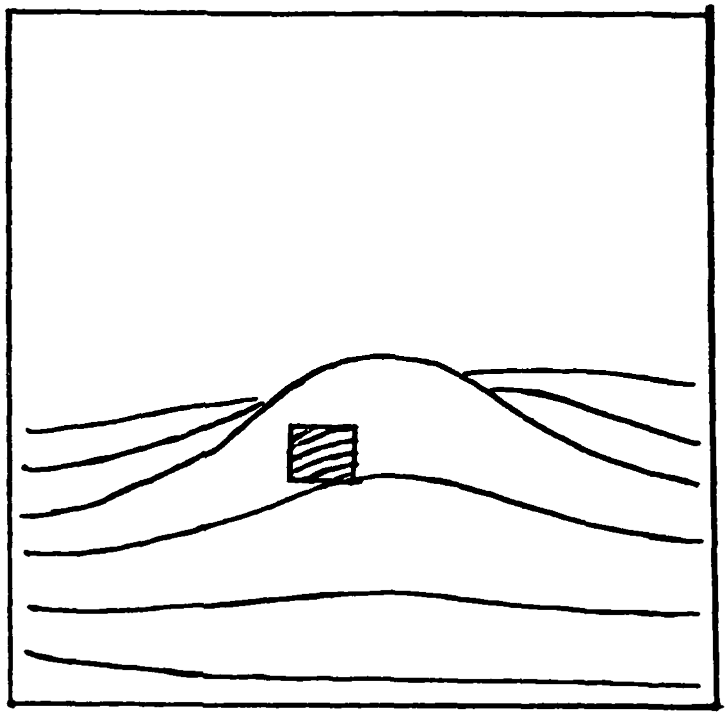


FIGURE 11. Acceptable Elements of a Raster Pixel (Any Point in Hatched Area).

This example will be used to examine the details of the algorithm used in the ETL software. The algorithm is implemented in the FORTRAN subroutine GRNDPT, to which the reader is referred (see appendix B). Note that the algorithm described is not restricted to the perspective projection. All that is required for its use with another projection is the substitution of the appropriate projection equation.

The analysis of the algorithm proceeds from the projection equations

$$X = X_o + \beta \ell ({}^{x}/y) \tag{51}$$

$$Y = Y_o + \beta({}^{\ell}/\cos\theta)\,(h - \gamma Z)/r \tag{52}$$

where the variables are as defined before. Recalling that the visibility problem has been simplified to one requiring only the determination of a visible point corresponding to a given pixel, the visibility problem can be solved by sampling the terrain along a radial. Since equation (10) can be rewritten as

$$X = -X_o + \beta \ell \cot\theta \tag{53}$$

and since θ is a constant for a radial elevation profile, the x coordinate will also be constant for the projection of a radial profile. This will correspond to a column of pixels for most raster display devices. Thus, a single radial profile of elevation data may be used to determine the visible portions of the terrain for an entire column of pixels. (see figure 3).

To illustrate the features of the algorithm, consider a general example. The image to be generated is to be composed on $m \times (2n + 1)$ pixels, where **m** is the number of pixels per column and the columns of pixels are numbered from **-n** to **+n** as illustrated in figure 12. A visible pixel that will be projected into the j^{th} pixel (from the bottom of the image) of the k^{th} column ($-n \leq k \leq n$) is sought. The first task, which is not executed unless the column of pixels is being considered for the first time, is the sampling of the terrain along a radial. This is accomplished by filling some array, say **Y**, with **p** successive elevation data values, $Y(R_I, \theta_K)$, such that

$$Y(R_I, \theta_K) = Y(X_{IK}, Y_{IK}) \tag{54}$$

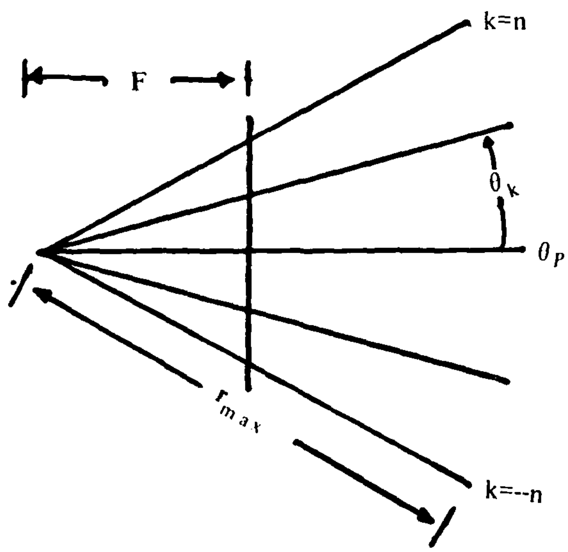


FIGURE 12. Radials in the Perspective Image.

Where

$$X_{IK} = R_I \cos\theta_K \tag{55}$$

$$Y_{IK} = R_I \sin\theta_K \tag{56}$$

$$R_I = (I - 1)\,\Delta R \tag{57}$$

$$\theta_K = \tan^{-1}(K\,\Delta Y / \ell) \tag{58}$$

$$1 \leqslant I \leqslant P \tag{59}$$

Here, y' is the incremental distance between pixel centers, f is the focal length of the imaging system, and r is the elevation sample interval. Note that

$$\Delta R = r_{max} / (P - 1) \tag{60}$$

where r_{max} is the maximum radius of interest and ΔR should be chosen such that the radial increment is less than one-half the wavelength of the highest frequency terrain feature of significance in the data base. A radial profile such as this, serving as a local terrain model, is illustrated in figure 13.

The visibility problem is solved implicity by processing from the bottom of a given column of pixels upward and outward along the radial elevation profile (see figure 13). The first time that the projection of two consecutive elevation data points bracket the center of a given pixel of interest, then terrain lying between those two data points will be projected onto the pixel. This search procedure, isolating a visible portion of the terrain bracketing the pixel of interest, is the first component of the correlation algorithm. The second component is a recursive procedure for isolating a point of the terrain just identified that will be projected with arbitrary accuracy onto the center of the pixel of interest. This is done by

1. Approximating the terrain by a linear model.
2. Solving for the point of the linear model that will be projected onto the center of the pixel.
3. Accessing the actual elevation data value at the predicted point.
4. Checking the projection of the actual point against the required tolerance and iterating the procedure again, if necessary.

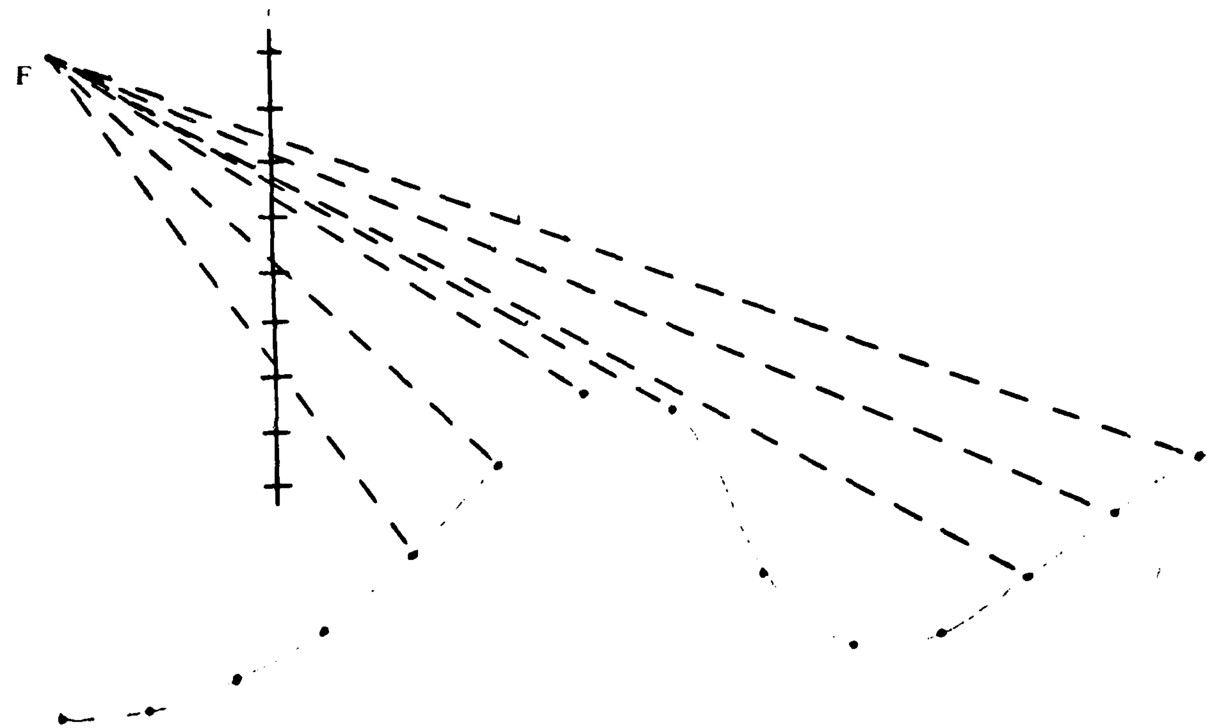


FIGURE 13. A Radial Terrain Profile With Sample Points Projected.

The first time that the correlation subroutine is called for a given column I, the counter governing the progression through the elevation radial is set to unity.

$$I = 1$$

The elevation values of the first and second entries of the elevation profile are retained in variables Z1 and Z2, since they will be needed in the second component of the correlation algorithm. The array positions are replaced with the image Y' coordinates of their projections (actually, Y(1) is replaced with a large negative number since the actual projection is −100). This is illustrated by the following code:

```
   Z1 = Y(1)
   Z2 = Y(2)
 Y(1) = -100
 Y(2) = PROJ(Y (2) )
```

where PROJ(Y (2)) symbolizes the operation

$$\text{PROJ}(Y(2)) = Y_0 + \beta\,(\ell/\cos\theta_K)\,(h - \gamma\,Y(2))/r \qquad (61)$$

the test of the first component of the algorithm is next executed. If the center of the current pixel of interest, YS = j x y, is less than Y (2), then the correlation logic described below is executed. If not, I is incremented, Z1 and Z2 are updated, Y(I + 1) is transformed, and the search procedure continues. If YS lies between Y (I) and Y (I + 1), then the search ends and the correlation logic described below is executed. If the current points don't bracket YS, then I is incremented again, unless the end of the elevation profile has been reached. In this case, a flag is set specifying that the rest of the current column of pixels should be imaged as 'sky', since no terrain within r_{max} will have a projection onto the current point or onto any higher pixel in the current column.

The qualitative description given above of the first component of the correlation algorithm may be symbolically coded as follows:

```
10   I = I + 1
     IF (I.NE.P)        GO TO 70
     NFLAG = 1
     RETURN

70   Z1 = Z2
     Z2 = Y (I + 1)
     Y (I + 1) = PROJ (Y (I + 1) )
     IF (Y (I + 1) .GE.YS.AND.Y (I).LE.YS)    Go to 20
     GO TO 10
```

Assume that the above code has been successfully executed and that two points of a radial elevation profile have been found with projections bracketing the Y' value of the current pixel of interest. The remaining component of the algorithm needs to be executed, that is, to find the coordinates of a point along the profile with a projection lying within the specified tolerance, TOL, of the pixel center YS. The iterative procedure for this will now be examined.

The iterative procedure used to find a terrain point between the known array positions Y(I) and Y(I + 1) satisfying the tolerance criteria is based on two observations:

1. The terrain between the I^{th} and the $I + 1^{st}$ elevation data values Z1 and Z2 will be monotone (increasing or decreasing) if the resolution **r** is adequate.
2. An analytic solution to the problem of finding the point corresponding to YS exists for the case that the terrain is linear.

In the first observation, it is implied that a series of linear approximations to the terrain between the two data points will converge to a solution. In the second observation, a quick, analytic method is presented for predicting the position of a data point that will map onto YS, based on the linear approximations. By then accessing the predicted data points and evaluating the projection of the actual point in terms of the tolerance criteria, an easily developed iterative procedure will yield a suitable point.

The key to this portion of the algorithm is the ability to determine a point on a linear surface, the perspective projection of which will fall onto a predetermined point on the image plane. The X' coordinate of the point has already been determined from the radial profiles. Thus, the Y' coordinate is of concern. As above, let the Y' coordinate of the screen point of interest take the value YS. If, as shown in figure 14, it is known that two elevation points, ZU and ZL, are visible at distances from the observer of RL and RU, respectively, then the terrain between the two points may be modeled by the linear function

$$Z(R) = \frac{(ZU - ZL)}{(RU - RL)} (R - RL) + ZL \qquad (62)$$

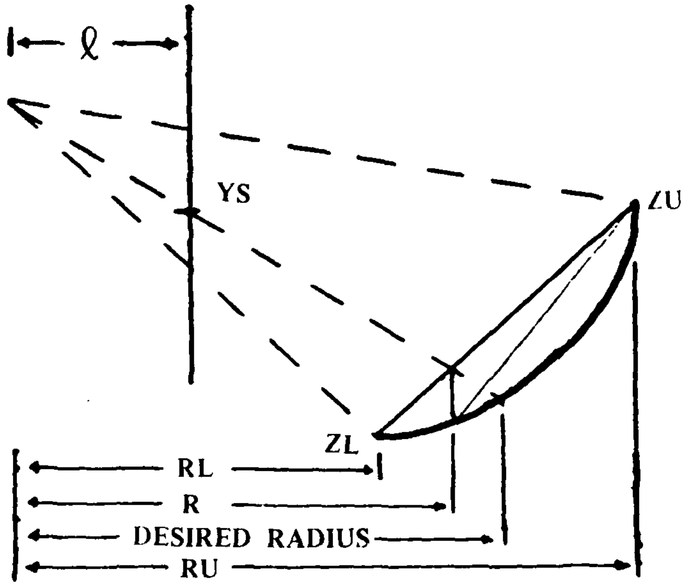


FIGURE 14. Concave Radial Profile.

After substituting this into the **Y'** equation of the perspective projection and solving for the **R** such that the projection of Z(R) is equal to YS, it is seen that

$$R = \frac{-hf + (ZL)\beta\gamma - [(ZU - ZL)\ell\beta\gamma\,(RL)/\Delta]}{[YS - Y\phi]\cos\theta - [(ZU - ZL)\ell\beta\gamma/\Delta]} \tag{63}$$

This predicted radius is then used to access the actual elevation at that point. The elevation is then projected, and if within the tolerance criteria, is a suitable point. This logic may be symbolically coded as

```
XI = XO + R*COS(THETA)
YI = YO + R*SIN(THETA)
CALL ALT(XI, YI, Z)
YTRY = PROJ(Z)
YERR = YTRY - YS
IF(ABS(YERR).LE.TOL)) GO TO 30
```

where ALT is a FORTRAN-callable subroutine that will return elevation values (z) at specified points (x, y).

Two cases exist if the projection of the predicted point is outside of the tolerance bounds. These, corresponding to locally convex and concave terrain, respectively, are illustrated in figures 14 and 15.

Consider the concave case first. In this case, YERR = YTRY – YS is less than zero; the convex case corresponds to a positive value of YERR.

```
IF(YERR.GT.0) GO TO 40
```

This situation is illustrated in figure 14. It is clear that a new, more accurate linear approximation to the terrain may be achieved by substituting the value of z derived from the first prediction for the value of ZL, and correspondingly replacing the value of RL with the predicted radius.

```
ZP = ZU - Z
ZL = Z
RL = R
DELTA = RU - RL
GO TO 50
```

The program will then predict a new radius, as outlined above, and in successive approximations, will approximate the desired point with an arbitrary degree of accuracy.

The convex case is similarly simple (see figure 15). In this case, the upper point and radius are modified before proceeding to the next iteration of the algorithm. In a manner similar to that of the concave case, the logic may be symbolized by

```
ZP = Z - ZL
ZU = Z
RU = R
DELTA = RU - RL
GO TO 50
```

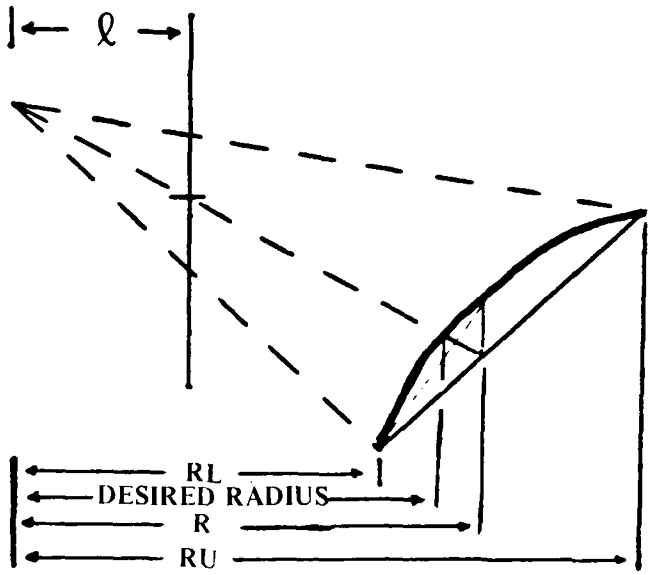


FIGURE 15. Convex Radial Profile.

Similar, successively more accurate approximations to the desired point will be made until a satisfactory point is found, i.e. until the projection of a predicted point lies within the tolerance interval of the point YS.

Once a satisfactory point is found, control returns to the main routine. To reduce redundant operation of the correlation routine, NFLAG, (the 'sky' flag noted above) one can pass the pointers to position in the elevation array and the last two elevation values to the main routine. Further, with the polynomial terrain data base, the orientation of the terrain may be determined at the same time that the predicted points are tested against the tolerance criteria. Consequently, the slope parameters described below are also passed to the main routine.

C. SLOPE DETERMINATION. In this section, the problem of determining the parameters necessary to calculate the brightness of the surface at an arbitrary point on that surface will be discussed. In particular, the brightness of the surface at points as determined by the algorithms of the previous section will be calculated. The actual calculation of the brightness of the terrain at our point of interest will not be discussed because it is conceptually and algorithmically better treated with the calculation of pixel density in the next section. From the previous algorithms, at a given point, the brightness of terrain obeying Lambert's law is a function of the cosine of the angle θ between the normal to the surface and the vector defining the incident radiation. Similarly, everything being equal, the brightness of terrain obeying the Lommel-Seeliger law is a function of $\cos\theta$, and the cosine of the angle i between the normal to the surface and the vector from the point under consideration to the observer. Thus, in this section the algorithmic determination of the normal to a surface for both discrete and polynomial terrain data bases will be discussed. In addition, the calculation of the cosine of the angle between this vector and some arbitrary vector as defined by a polar angle α and an azimuthal angle β will be presented.

1. Normal to a Surface. Having found a point on the surface of the terrain with a corresponding projection within the pixel of interest on the image plane, one's next task is to determine the orientation of the surface at that point. The most convenient way to define the orientation is by a unit vector that is normal to the surface at that point.

A two-dimensional surface can be defined by the equation

$$Z = Z(u, v) \tag{64}$$

where **u** and **v** are arbitrary curvilinear coordinates on the surface of the terrain. The normal-to-a-surface at a point is defined by the cross product of any two district vectors tangent to the surface at that point. This is clear since the vector defined by the cross product of two vectors is perpendicular to both of them. Two such vectors are defined by the partial derivatives of the function with respect to the two variables. Thus, the normal vector (not a unit vector) is given by

$$\vec{r} = \frac{\partial \vec{f}}{\partial u} \times \frac{\partial \vec{f}}{\partial v} \tag{65}$$

Generalized curvilinear coordinates are of little interest because most topographic data bases are defined by the usual orthogonal cartesian coordinate system. However, two algorithms are of interest at this point: (1) the generation of normals from a discrete elevation data base, and (2) the generation of normals directly from the coefficients of a polynomial terrain data base. Algorithms of both types were implemented in the FTL software.

a. Discrete Data Bases. First, consider an elevation data base composed of discrete elevation data points. No particular format will be assumed for these data points (see Yoeli*). Thus, the points may be found at the intersection of some orthogonal grid, maybe randomly distributed, or maybe composed of points selected on the basis of arbitrary criteria. Having dealt with the general case, one can simplify the results for the case of data points found at the nodes of a uniform orthogonal grid.

Let us consider three data points θ (x, y, z), a (x, y, z), and b (x, y, z), surrounding the point **P** at which we seek to find the normal. These three points define the local surface at this point. The boundaries of the surface can be defined by two vectors $\vec{a}$ and $\vec{b}$, respectively, correcting the points ϵ a and **b** to θ. Algebraically, these vectors are defined by

$$\vec{a} = a(x, y, z) - \theta(x, y, z) \tag{66}$$

$$\vec{a} = (a_x - \theta_x, a_y - \theta_y, a_z - \theta_z) \tag{67}$$

and

$$\vec{b} = b(x, y, z) - \theta(x, y, z) \tag{68}$$

$$\vec{b} = (b_x - \theta_x, b_y - \theta_y, b_z - \theta_z) \tag{69}$$

Since the vectors $\vec{a}$ and $\vec{b}$ define the local boundaries of the surface around **P**, and are thus coplanar and hence tangent to the surface, then the (non-unit) normal vector at **P** is given by $\vec{n} = \vec{a} \times \vec{b}$ with components

$$n_x = a_y b_z - a_z b_y \tag{70}$$

$$n_y = a_z b_x - a_x b_z \tag{71}$$

$$n_z = a_x b_y - a_y b_x \tag{72}$$

*See footnote 26.

The length of $|\vec{n}|$ of $\vec{n}$ is given by $\sqrt{(n_x^2 + n_y^2 + n_z^2)}$, and hence, $\hat{n}$, the unit normal vector is given by

$$n = n/ n \tag{73}$$

If the data base is composed of elevation data points found at the intersections of lines parallel to the X- and Y- axis and is uniformly spaced, then a simplification in the formula is possible, resulting in reduced computation times. If a lies on a vertical plane parallel to that defined by the Z- and X- axis and **b** lies on a vertical plane parallel to the Z- and Y- axes, then $\vec{a}$ is given by (a_x, o, a_z) and $\vec{b}$ is given by (o, b_y, b_z). Thus, the normal vector is given by

$$n_x = -a_z b_y \tag{74}$$

$$n_y = -a_x b_z \tag{75}$$

$$n_z = a_x b_y \tag{76}$$

As a result, reductions will occur in the computations necessary to compute the cosine of the angle between a normal vector and any other vector.

The above formulas are somewhat arbitrary, because the information content of the data base is not used in an unbiased manner. Thus, if a pixel center is taken to coincide with a given point of a regular grid data base, the four nearest points are all equidistant and, hence, are of equal information content. The above equations use only two of these four points and, hence, are biased (see figure 16).

The easiest way of removing this bias is by shifting the pixel center to the center of one square of the grid of the data base and then using the mean of the slopes determined along the sides of the square. Hence, the Z –components of equations (66) and (67) may be re-defined as

$$a_z = 1/2 (Z_2 - Z_1 + Z_4 - Z_3) \tag{77}$$

$$b_z = 1/2 (Z_3 - Z_1 + Z_4 - Z_2) \tag{78}$$

where the Z_i are as defined in figure 17. These formulas have been implemented in the ETL software.

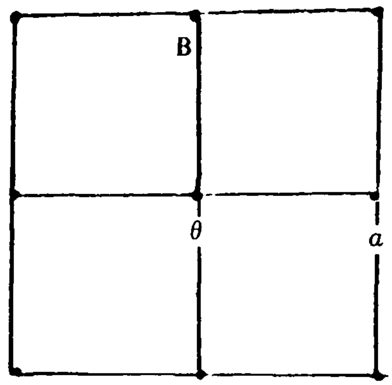


FIGURE 16. Biased Use of Elevation Data Points.

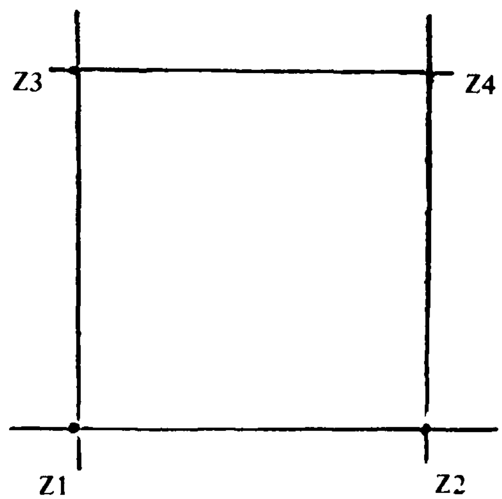


FIGURE 17. Unbiased Use of Elevation Data.

b. Polynomial Data Bases. The problem of analytically determining the normal to a topographic surface at a point will be discussed for a terrain modeled by a polynomial data base of the type developed for ETL.[29]

The elevations are defined as a function of position

$$Z(X, Y) = \sum_{i=1}^{4} Z_i(X, Y)\ W_i(X, Y) \tag{79}$$

where the $Z_i(X, Y)$ are the four partially overlapping polynomial functions locally defining the terrain around the point $P(X, Y)$, and the $W_i(X, Y)$ are the four corresponding weight functions used to define the terrain in the overlap area.

The normal to the surface at the point $P(X, Y)$ can be defined in terms of the two partial derivations, $\partial Z/\partial X$ and $\partial Z/\partial Y$, as outlined above. From straightforward applications,

$$\partial Z/\partial X = \sum_{i=1}^{4} (\partial Z_i(X, Y)/\partial X\ W_i(X, Y) + \partial W_i(X, Y)/\partial X\ Z_i(X, Y)) \tag{80}$$

$$\partial Z/\partial Y = \sum_{i=1}^{4} (\partial Z_i(X, Y)/\partial Y\ W_i(X, Y) + \partial W_i(X, Y)/\partial Y\ Z_i(X, Y)) \tag{81}$$

To use equations (80) and (81), one must have a vector representation in a form analagous to the vectors defined for the discrete data base. The best concept of this is to consider these tangent vectors as defining a plane tangent to the topographic surface at the point of interest. If $\vec{a}$ describes $\partial Z/\partial X$ and $\vec{b}$ describes $\partial Z/\partial Y$, then the vectors $\vec{a}$ and $\vec{b}$ can be defined by

$$\vec{a} = (1, 0, \partial Z/\partial X) \tag{82}$$

$$\vec{b} = (0, 1, \partial Z/\partial Y) \tag{83}$$

[29] J. Jancaitis and J. Junkins. *Mathematical Techniques for Automated Cartography*. U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, ETL–CR–73–4, February 1973, AD–758 300.

remembering that $\vec{a}$ and $\vec{b}$ lie in vertical planes parallel to the X- and Y-axes, respectively. The normal, $\vec{n}$, is defined as above with

$$n_x = -\partial Z/\partial X \tag{84}$$

$$n_y = -\partial Z/\partial Y \tag{85}$$

$$n_z = 1 \tag{86}$$

$$n = n/|n| = n/\sqrt{(n_x^2 + n_y^2 + n_z^2)} \tag{87}$$

2. **Calculation Of Angles To A Normal.** Next will be discussed the problem of determining the angle i between $\vec{n}$, the normal vector to our topographic surface at a point **P**, and determining some vector $\vec{S}$, with component S_x, S_y, and S_z. We shall define $\vec{S}$ in terms of a polar angle φ, and an azimuthal angle θ (see figure 18). Then

$$S_x = \sin\varphi \cos\theta \tag{88}$$

$$S_y = \sin\varphi \sin\theta \tag{89}$$

$$S_z = \cos\varphi \tag{90}$$

Note: As defined, $\vec{S}$ is a unit vector.

As noted above, our interest actually lies in the cosine of the angle between these two vectors. If the scalar product of two vectors is defined, then

$$\vec{S} \cdot \vec{n} = |\vec{s}|\ |\vec{n}| \cos i = S_x n_x + S_y n_y + S_z n_z \tag{91}$$

where i is the angle between the two vectors. If $\vec{S}$ is normalized, then, by equations (58) to (60)

$$\cos i = \vec{S} \cdot \vec{n}/(|\vec{s}|\ |\vec{n}|) \tag{92}$$

$$\cos i = \frac{S_x(a_y b_z - a_z b_z) + S_\partial(a_z b_x - a_x b_y) + S_z(a_x b_y - a_y b_x)}{(a_y b_z - a_z b_y)^2 + (a_z b_x - a_x b_y)^2 + (a_x b_y - a_y b_x)^2} \tag{93}$$

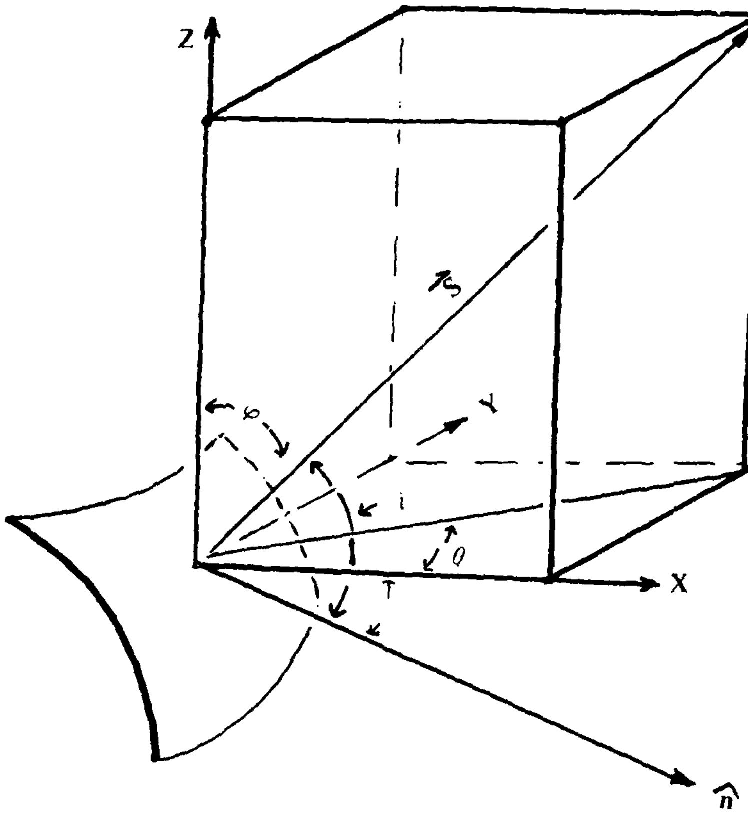


FIGURE 18. Angles to a Normal.

If, as outlined above, $\vec{a}$ and $\vec{b}$ fall into vertical planes parallel to the X- and Y-axes, respectively, then this formula is

$$\cos i = \frac{-S_x a_z b_y - S_y a_x b_z + S_z a_x b_y}{\sqrt{\{(a_z b_y)^2 + (a_x b_z)^2 + (a_x b_y)^2\}}} \qquad (94)$$

where $a_y = 0$ and $b_x = 0$ for this case.

Further, if the components of the normal vector are calculated from the polynomial data base as in equations (84) and (85), then

$$\cos i = \frac{-S_x \, \partial Z/\partial X - S_y \, \partial Z/\partial Y + S_z}{\sqrt{\{(\partial Z/\partial X)^2 + (\partial Z/\partial Y)^2 + 1\}}} \qquad (95)$$

It will be recalled that the Lommel-Seeliger law uses the viewing angle n_z, which is the angle between the normal to the surface and the unit ray $\vec{V}$ to the observer, as well as the angle of illumination i between the unit ray $\vec{V}$ to the sun $\vec{S}$ and the normal to the surface. The general case follows equation (93) with the components of $\vec{V}$, replacing those of $\vec{S}$, and ϵ replaced i. Similar substitutions are held for equations (94) and (95). For the orthonormal view, equations (94) and (95) become

$$\cos i = \frac{V_z a_x b_y}{\sqrt{\{a_z^2 b_y^2 + a_y^2 b_z^2 + a_x^2 b_y^2\}}} \qquad (96)$$

and

$$\cos i = \frac{V_z}{\sqrt{\{(\partial Z/\partial X)^2 + (\partial Z/\partial Y)^2 + 1\}}} \qquad (97)$$

where V_z is the Z-component of the unit ray from the point of interest to the observer.

D. DENSITY COMPUTATIONS. The current problem is to calculate the illumination of a pixel of the image plane, for either perspective or orthonormal projection of terrain obeying either Lambert's or the Lommel-Seeliger brightness law, for a given direction of illumination and a given observer location.

The position of a point on the terrain corresponding to the current pixel that we are processing and the normal to the surface at that point have already been determined. First, computation of lighting and viewing vectors as used in the ETL software will be defined. The vectors are then applied to the computation of the illumination of our pixel. Next, the density concept used the concept of density to define the properties of the pixel that was used to define the image produced on an output device will be introduced.

1. Definition and Calculation of Vectors of Observation and Illumination. First, the vector of illumination that describes the position of the sun relative to the terrain of interest will be defined. Algorithmically, the normal vector is needed in terms of its X-, Y-, and Z-components, in the (X, Y, Z) coordinate system of the object. An angular definition is sometimes easier to understand, specifying an altitude φ and an azimuth θ for the sun. Fortunately, a simple transformation is possible (see figure 19). The ETL data bases, as well as most others, can locally define the Y-coordinate of the terrain as pointing due north and the X-coordinate as pointing due east. Aeronautical convention fixes azimuth such that 0° is due north, or parallel to the Y-axis, and 90° is due east, or parallel to the X-axis. Altitude is defined such that 0° is parallel to the surface, and 90° points towards the zenith. Thus, if $\vec{S}$ represents the vector pointing towards the sun, the transformation is

$$S_x = \cos\varphi \cos\theta' \qquad (98)$$

$$S_y = \cos\varphi \sin\theta' \qquad (99)$$

$$S_z = \sin\varphi \qquad (100)$$

where we introduce the variable $\theta' = 90° - \theta$.

Next, consider the definition of the unit vector of observation, $\vec{V}$, pointing from the point of the surface that we are considering to the instrument of observation. This vector is important only for the Lommel-Seeliger law; it does not

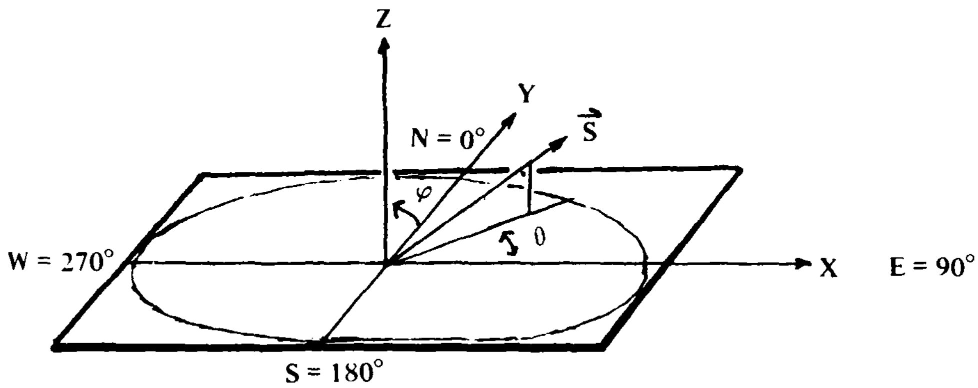


FIGURE 19. Definition of Solar Vector.

have a role in the brightness of a surface as described by Lambert's law. Further, for the case of an orthonormal image of terrain obeying the Lommel-Seeliger law, this vector is always vertical, in accordance with the definition of an orthonormal image.

All that needs to be considered then is the calculation of $\vec{V}$ for a perspective view, and this need only be done when a Lommel-Seeliger image is to be created. Consider a perspective image geometry as illustrated in figure 20. The focal point **P** is defined at some distance Z_0 above a point **Q** of the terrain. A vertical plane corresponding to the i^{th} column of pixels is defined by requiring that it be a vertical plane containing $\overline{PQ}$ and that it be positioned at an angle θ to the northerly vertical plane. From equation (58), θ can be defined in terms of a principle viewing direction θ_p specified by the user, and an angle θ_i relative to the principle direction. (figure 20). Thus,

$$\theta = \theta_p + \theta_i \tag{101}$$

$$\theta_p + \tan^{-1}\left(\frac{i \Delta X}{\varphi}\right) \tag{102}$$

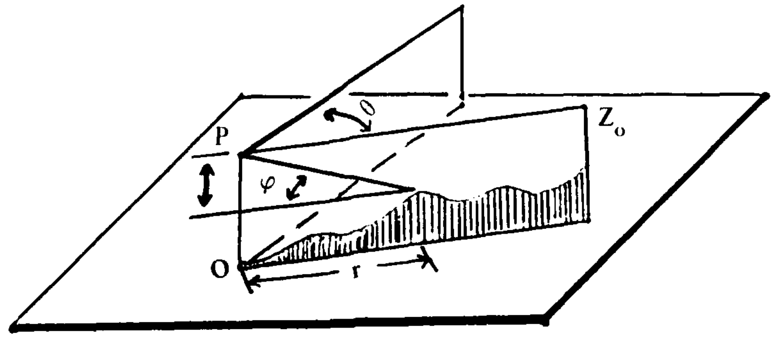


FIGURE 20. Definition of V.

where ΔX is the incremental distance between columns of pixels on the image plane, i defines the current column of interest, and f is the distance between **P** and the image plane along the principle viewing direction.

Assume that point **T** (r, θ, Z) of the terrain is found, corresponding to the pixel of interest, where θ is as above, **r** is the distance from **O** to the projection of **T** onto the horizontal plane containing **O**, and **Z** is the elevation of **T** above this plane. Then the angle of altitude φ is defined by

$$\varphi = \tan^{-1} \left(\frac{Z_o - Z}{r}\right) \tag{103}$$

Thus, the components of the perspective viewing vector $\vec{V}$ for a given pixel of the perspective projection as defined in terms of previously determined quantities is

$$V_x = -\cos\varphi \sin\theta \tag{104}$$

$$V_y = -\cos\varphi \cos\theta \tag{105}$$

$$V_z = \sin\varphi \tag{106}$$

Thus, the parameters necessary to compute the image illumination for the cases that are being considered have been defined.

2. **Image Illumination Calculation.** To calculate image illumination, one must use the light-scattering laws previously reviewed. In the ETL Software, the user may specify that the brightness of the terrain is to be modeled by either the Lommel-Seeliger law, $B = Eo\ b\sum(\alpha)/1 + \cos\theta/\cos i$, or by Lambert's law, $B = Eo \cos i$, where the variables are defined in appendix A.

First, normalize Eo; it is not a user defined parameter. Second, assume in the case of the Lommel-Seeliger law that $b\Sigma(\alpha)$ is constant for all α and that this factor drops out. Thus, in essence the light-scattering law is reduced to

$$B = [1 + \cos\theta / \cos i]^{-1} \tag{107}$$

for the Lommel-Seeliger law and to

$$B = \cos i \tag{108}$$

for Lambert's law.

As mentioned in section II, D, the concern is with the illumination of an image element in an optical system, not with the brightness of the illuminating point as observed at the optical system. As approximated within the ETL Software, the relationship is one of proportion.

$$< E > \approx \alpha B \tag{109}$$

where $< E >$ is the illumination of our pixel, and **B** is the brightness of some point of the image found by the object-image correlation algorithm (see section III, A). The calculation for **B** is dependent only on the relevant angles, which are found by evaluating the slope at the object point by means of the appropriate algorithm (see section III, B), the position of the observer (for the Lommel-Seeliger case) based on equations (104) to (106), and the position of the sun as defined by equations (98) to (100). The angles are then used to determine the cosines of the relevant angles by the appropriate formula of the sequence (91) to (97). The cosines are then used to determine the brightness by which the illumination of the pixel is approximated (after normalization).

However, the quantity that is to be used to define the image to be produced is not illumination, rather it is the density of the image. Before specifying the relationship, the meaning of density must be explained.

Consider a piece of photographic film, with illumination Eo, normal to the surface of the film. Let us assume that the film has been exposed and developed and, hence, that some light will be absorbed. The change in illumination across an infinitesimal layer of the fill will be

$$dE = -bE\,dt \qquad (110)$$

where **b** is the fraction of the light absorbed in passing a unit thickness, **E** is the illumination of the upper surface of the infinitesimal layer, and **dt** is the thickness of the layer. Integrating, we find that the illumination of the far side will be

$$E = Eoe^{-bt} \qquad (111)$$

where **t** is the thickness of the film. If the film is designed for viewing from the front, then assuming that all light is reflected at the bottom of the film, it undergoes similar absorbtion in the second pass through the film and obeys Lambert's law upon striking the upper surface (as in a matte print). The brightness of the film will be given by

$$B = Eoe^{-2bt} \qquad (112)$$

In any case, convention defines the image, not by the brightness or intensity, but by the density. The density, **D**, is proportional to the exponent of equations (111) or (112). Consider (111) and let

$$D' = \ln(E/Eo) = \ln(e^{-bt}) = -bt \qquad (113)$$

This is not the usual density, which is conventionally defined in terms of the $\log_{10}$ and must be positive. The relation is given by

$$D = \log(Eo/E) = \log_{10} e^{+bt} = \alpha bt = \alpha 2D' \qquad (114)$$

where $\alpha = 1/\ln_{10}$. The term Eo/E is known as the opacity.

Since the intensity Eo in the ETL algorithm is normalized, the density **D** is defined as

$$D = \log_{10}(1/E) \qquad (115)$$

where **E** is calculated as outlined above. Output devices have between 12 and 256 different discrete density values available, with the values linearly positioned between some minimum and maximum density values. Consequently, a density scaling factor is used in the ETL algorithms to take advantage of the latitude in densities offered by the output devices, storing each density value to an 8-bit byte.

Thus the analysis of the basic algorithms of the ETL shaded relief software is completed.

E. GRAPHIC PRODUCTION. The basic algorithms governing the pixel-by-pixel generation of the shaded relief images have been described. The result of these algorithms, for a given pixel, is a number. The process of converting this number into a pixel of corresponding density at a given position relative to the other pixels of the image remains to be discussed.

Three types of devices are available at ETL to generate gray-shade images: (1) line printers for low resolution "proofing" plots, (2) a VERSATEC raster plotter to generate moderate resolution digital half-tone images, and (3) an Electron Beam Recorder (EBR) to generate high resolution near-continuous tone images. The first two output devices require special algorithms to generate shaded relief images, and the third device directly accepts a file containing sequential rows. Since the software to generate line printer or VERSATEC graphics requires sequential processing similar to that of the EBR and since it may be derived to output a given image to any or all output devices, the product of the ETL shaded relief software is a disk file contouring the coded density data. Both files store 7 bits of data (densities 0–127) in 8-bit bytes. Each record represents one column of pixel, and successive records represent successive columns. For compability with the disk access routine DSKTRN used at ETL, the image files generated are of uniform size, 2048 bytes by 2048 records for the file LPPERDAT generated by the shaded perspective routine SHDPER, and 1024 bytes by 1024 records for the orthonormal shaded relief routine SSLPLP.

The line printer routine used was devised by P. Yoeli and uses multiple stikings to generate varying densities.[30] The look-up table used for this algorithm is shown in table 1. The VERSATEC halftoning software creates a digital halftone pattern, characterized by 33 halftone dots per inch.[31] Using the EBR is described in the literature.[32]

Since all the algorithms used are described in available literature, no additional discussion is necessary. It is clear, though, that the file generated by the ETL software can be used by any gray-shade graphics generation device with minimal trouble.

---

[30]P. Yoeli, "The mechanization of Analytic Hill Shading," *Carto. J.* 1967.

[31]R. Rosenthal, Private Communication, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, 3 July 1979.

[32]*Operation and Maintenance Manual for Cartographic EBR System*, Image Graphics, Inc., under contract No DAAG–53–75–C–0221, 1976, pp. 67–72.

TABLE 1. Line Printer Density Table.

CHARACTERS			DENSITY
1st Printing	2nd Printing	3rd Printing	
			0.00
.			0.07
–			0.10
I			0.12
/			0.15
V			0.19
#			0.22
@			0.26
U	.		0.30
A	/		0.35
A	V		0.40
A	@		0.46
A	V	1	0.52
A	U	4	0.60
A	@	W	0.70

F. SPECIAL PURPOSE ALGORITHMS. As noted in the introduction to this section, a number of special-purpose algorithms devised to address specific cartographic problems have been implemented in the ETL shaded relief software. The algorithms enable the solar azimuth to be varied, thus enhancing the impression of terrain relief and delineating the ridge lines.[33] Producing "relief contours" enables atmospheric haze to be simulated, thus providing the option of simulating an additional qualitative distance cue in the perspective images.

1. **Variable Sun Angle Algorithms.** A problem occurs when a single source of illumination is used to delineate similar terrain features in different orientations. This is as much a problem for aerial photography as it is for artifically generated orthonormal or perspective images. First, the problem will be analyzed, then the various alleviating procedures developed by cartographers in the past will be considered. Finally, the adaptations of procedures implemented in the ETL software, will be examined.

The problems inherent in using a single source of illumination may be seen in figure 21, which depicts an idealized ridge illuminated by side lighting. The ridge slop facing the sun is illuminated, and the far slope is in shadow. The form of the terrain is clear to an observer if he is aware of the direction of illumination. Now consider a situation such as in figure 21(b), in which the illumination is parallel to the ridge line. In such a case, both sides of the ridge are equally bright, and no information as to the terrain form is available to the observer. It is clear that in an area with varied topography and illuminated by a single source of light, the form of the terrain is poorly delineated, with some features exaggerated and some washed out.

As noted in the introduction, the problems associated with depicting terrain form are not new. For sometime, cartographers have been manually shading contour maps to improve interpretation of the terrain features. A simple solution might be in the careful choice of the direction of lighting, thus optimizing the delineation of the terrain. However, a problem exists that prohibits such a solution. This is the problem of interpretation, which requires a near-intuitive knowledge of

[33]P. Yoeli. "Die richtung des licht bei analytischen," *Kartographische Nachricten Guetersloh* 17(1967), pp. 537–544.

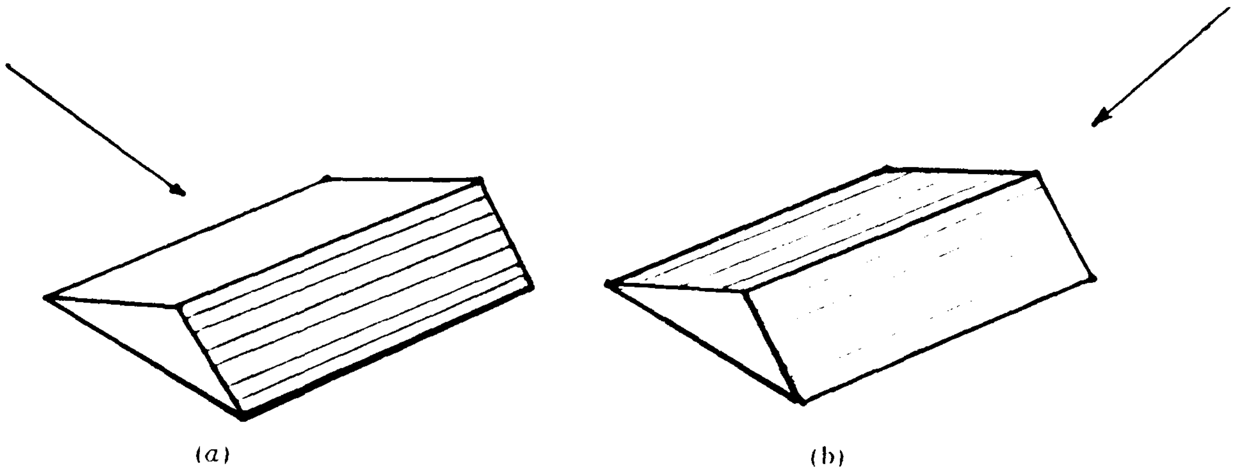


FIGURE 21. Sun Azimuth Importance.

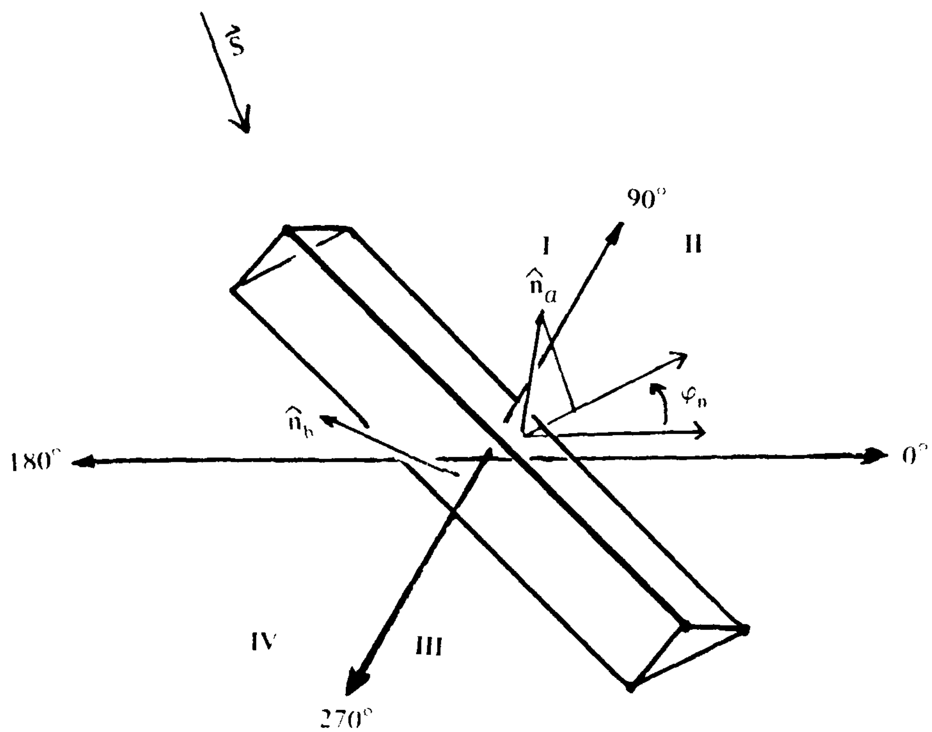


FIGURE 22. Variation of Solar Azimuth.

light direction. Shaded relief images are usually treated as though lighting is from the top or upper left. Because of this treatment, cartographers usually use north-west lighting when producing shaded relief overlays for contour maps, thus preventing the choice of the lighting direction.

To overcome the problem, cartographers have developed the procedures of implicitly varying the direction of the lighting to emphasize all major terrain features. The variation in lighting direction is around a principal direction defining north-west lighting. Although cartographers manually prepare shaded relief overlays of contour maps, substantial progress has been made in automating the process. Although quite sophisticated, the present algorithms are largely experimental.[34] The algorithms used at FTL are relatively simple, created largely to demonstrate the capability.

The FTL software is based on the variation of the azimuthal component of the illumination vector. Thus, the vertical component will not be considered in this discussion. North-west lighting will be used throughout, and algorithm will be based on the local orientation of the surface, as defined by the normal vector $\hat{\mathbf{n}}$. The azimuth of $\hat{\mathbf{n}}$ will be φ_n.

First, let us consider the case of ridges parallel to the principal direction of light **S** (see figure 23). The normals to the two faces of the ridge lie roughly in the second and fourth quadrants of the Cartesian plane. This is the case where the greatest variation will probably be needed, as this is an example in which the ridge would be washed out. To guide our development of an algorithm, we note that as the ridge is rotated from an orientation parallel to the Y-axis to an orientation parallel to the X-axis, face **b** gradually dims and face **a** gradually brightens. A corresponding change will occur if the illumination vector **S** is changed from a position parallel to the X-axis to a position parallel to the Y-axis. Thus, for normals in the second quadrant the "washout" effect can be prevented by defining the light vector components S_x, S_y by

$$(S_x = -1, S_y = 0) \qquad 0 \leq \varphi_n \leq 45^\circ \qquad (116)$$

$$(S_x = 0, S_y = 1) \qquad 45^\circ < \varphi_n \leq 90^\circ \qquad (117)$$

[34] K. Brassel, "A Model for Automatic Hill Shading," *Am. Cart.*, 1(1974), pp. 15–27.

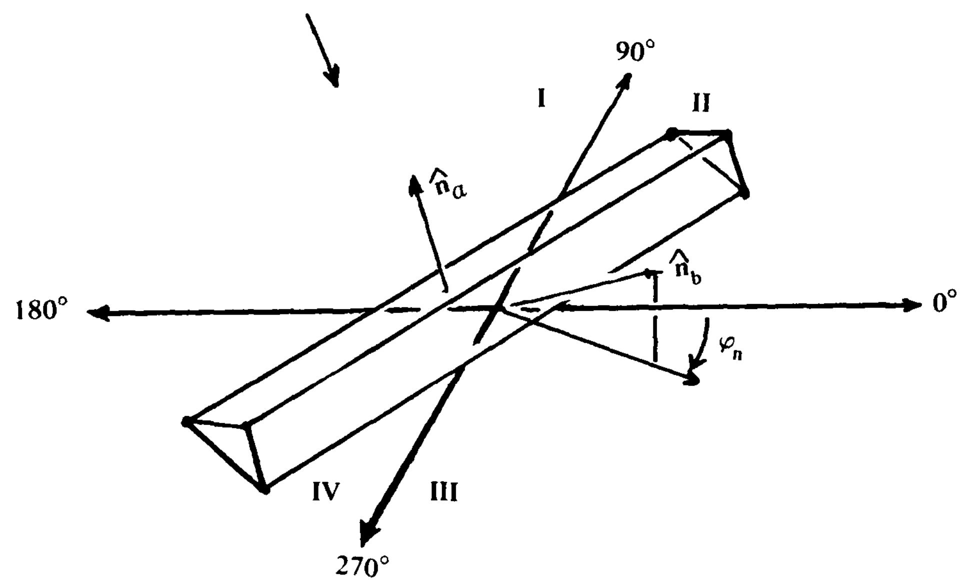


FIGURE 23. Variation of Solar Azimuth.

and by

$$(S_x = -1, S_y = 0) \text{ for } 180^\circ < \varphi_n \leqslant 225^\circ \quad (118)$$

$$(S_x = 0, S_y = 1) \text{ for } 225^\circ < \varphi_n < 270^\circ \quad (119)$$

for normals in the fourth quadrant.

For areas in between, one would like a smooth transition between the two lighting vectors used in equations (116 – 119). This can be defined by the first quadrant

$$(S_x = -\cos\varphi_n, S_y = \sin\varphi_n) \quad 90^\circ < \varphi_n \leqslant 180^\circ \quad (120)$$

and for the second quadrant

$$(S_x = \cos\varphi_n, S_y = \sin\varphi_n) \; 270^\circ < \varphi_n \leqslant 360^\circ \quad (121)$$

In practice, one would like more control over the variation in lighting direction. The ETL software enables the user to input the angular variation $Z\theta$ desired. The above formulas are altered accordingly. Thus, instead of the orthongonal lighting directions specified by equations (116 – 119),

$$[S_x = -\cos(135^\circ + \theta), S_y = \sin(135^\circ + \theta)], \; 325^\circ + \theta < \varphi_n \leqslant 145^\circ \quad (122)$$

$$[S_x = -\cos(135^\circ + \theta), S_y = \sin(135^\circ - \theta)], \; 145^\circ < \varphi_n \leqslant 135^\circ - \theta \quad (123)$$

$$[S_x = -\cos(135^\circ + \theta), S_y = \sin(135^\circ + \theta)], \; 135^\circ + \theta < \varphi_n \leqslant 225^\circ \quad (124)$$

and

$$[S_x = -\cos(135^\circ - \theta), S_y = \sin(135^\circ - \theta)], \; 225^\circ < \varphi_n \leqslant 325^\circ - \theta \quad (125)$$

Formulas (120 and 121) are unaltered, except that the ranges are modified in accordance with equation (122 – 125).

Determining the local illumination vector straight forward, and it calculated immediately after the normals are determined, as outlined. The local illumination vector is then used to determine the illumination of the local surface area and hence, the density of the image.

2. **Relief Contours.** Several attempts have been made to combine directly the quantitative properties of contours with the qualitative advantages offered by shaded relief images by creating an image of a contour terrace model. These attempts implicitly involve creating a "layer-cake" model of the terrain, with the elevation discontinuities corresponding to contour lines. This model is then illuminated, and shadows are cast by the tier structure, which is then photographed. In practice, actual models are rarely created. Instead, an ingenious manual method developed by Kitiro Tanaka is used.[35] Working from a contour map, one can trace the contours away from the assumed source of light, keeping the nib parallel to the light source. Thus, a variable width contour line results. For complete representation, a gray background is used, and contours facing the light are similarly inked in white. The result is a striking, if costly, representation of the terrain.

An alternative method of creating a map with the visual impact similar to that of Tanaka's method can be created using the software developed at ETL. The method uses the main orthonormal shaded relief routine, SSLPLP, but calculates slopes in a slightly different manner. Briefly, the relief contour option of program SSLPLP involves creating a square grid of quantized elevation data. The quantization interval is a user input parameter and corresponds to the contour interval of the image. The quantized elevation values are then used to calculate slope for each pixel center. The slopes are used to calculate image densities at each pixel position as in the usual method for creating a shaded relief overlay.

The calculation of the slopes is straightforward. Each pixel has a uniform size and is assumed to be square. Elevation data values are generated at the ground locations corresponding to the four corners of the pixel. Letting Z_{ul} be the Z–value of the upper left corner of the pixel, Z_{ur} be the elevation value of the upper right corner, and Z_{ll} and Z_{lr} be the corresponding lower elevation values, we have

$$\Delta Z / \Delta X = \tfrac{1}{2}(Z_{ur} - Z_{ul} + Z_{lr} - Z_{ll}) / \Delta X \qquad (126)$$

$$\Delta Z / \Delta Y = \tfrac{1}{2}(Z_{ul} - Z_{ll} + Z_{ur} - Z_{lr}) / \Delta Y \qquad (127)$$

where $\Delta X = \Delta Y =$ pixel size.

---

[35] K. Tanaka, "The Orthographic Relief Method of Representing Topography on Maps," *Geogr. Rev.*, 40(1950), pp. 444–456.

Since elevation values are quantized, unless a corresponding contour line passes through a given pixel, all elevations for that pixel will thus be assigned the neutral background density corresponding to flat terrain. If a single contour line passes through a pixel, then the mean plane fit to the four points corresponding to the pixel can assume any of 24 different orientations. Since the resolution of the image created is variable, more than one contour line may pass through a pixel. Thus the number of orientations is correspondingly increased.

It should be noted that ea n point of a contour line will be assigned to som pixel and that the width of the contour line cannot be less than the size of the pixel. To maximize contrast, the ETL software alters the vertical exaggeration such that illumination of slopes facing the source of illumination is maximized. Since contour lines parallel to the direction of illumination will be of low contrast relative to the background, it is usually convenient to utilize the variable sun angle algorithm described in this report.

3. **Simulation of Atmospheric Haze.** A simple model to simulate one aspect of atmospheric haze has recently been proposed.[36] This model treats only the attenuation properties of such haze, predicting for uniform haze an exponential decay of the apparent luminous intensity of a surface element with increasing distance. This is certainly one aspect of such haze, but only in the case of highly absorbent haze (such as, perhaps, industrial smog) or when the haze is not actually imaged but shades the ground as do clouds. The most important prediction of this model is that, for thick haze, distant objects will appear very dark. This is not the case with most actual haze. Consider a moderately thick ground fog. Distant "objects" in such a fog approach a uniform, non-zero brightness. To model more accurately the effects of such haze, ETL personnel have developed a new model of atmospheric haze.

---

[36]W. Dungan, Jr. "A Terrain and Cloud Computer Image Generating Model," *Computer Graphics*, 13(1979), No. 2, p. 143.

Careful consideration of the physics of light scattering results in a more appropriate (though still approximate) model of haze. Consider a thin slice of the atmosphere of thickness **dr**, measured along a radial to an observer. Fog or haze in this atmospheric section can be modeled as a collection of small randomly distributed particles with a cross section σ, and brightness **b**, with a density of **n** such particles per unit volume. The attenuation of a light beam traversing this section of the atmosphere along the radial to the observer will be

$$dE = -En\sigma dr \tag{128}$$

where **E** is the luminous intensity of the beam.

If the sun or other source of illumination shines upon the slice, then there will be an additional term to the expression for the attenuation of a beam traversing the ground haze. This term, expressing the fraction of the incident sunlight that is scattered into the beam, will be proprotional to $bE_s n\sigma$, where E_s is the luminous intensity of the source of illumination. Detailed analysis of the situation requires that the scattering be calculated in terms of illumination of an imaging element, considering the area of the light sensitive element, the size of the light acceptance cone of it, and the distance of a given scattering section of the atmosphere from the element. However, the net result of these considerations is a constant factor. It is thus convenient to lump this constant together with the brightness, **b**, in a new constant, **C**.

The differential equation governing the propagation of light through a ground fog, along a path roughly parallel to the surface of the earth is

$$dE = (-En\sigma + CEn\sigma)\, dr \tag{129}$$

Integration yields the solution

$$E(r) = \exp(-rn\sigma)\ [CE_s \{\exp(rn\sigma)\} + K] \tag{130}$$

where **K** is a constant of integration. Two boundary conditions must be satisfied by this constant of integration if the result is to be physically meaningful. These boundary conditions are

1. $E(r) \to E_0$ as $n\sigma \to 0$, where E_0 is the brightness of some object in the distance; and

2. $E(r) \to CE_s$ as $n\sigma \to \infty$ or as $r \to \infty$.

The first of these conditions corresponds to the physical situation of clean air, in which distant objects should be unobscured. The second boundary condition requires that the model of haze results in a uniform, finite brightness as the thickness of the haze increases. Both of these boundary conditions are satisfied by choosing

$$K = E_o - CE_s \tag{131}$$

Thus, the model of haze implemented at ETL takes the form

$$E(r) = CE_s + (E_o - CE_s)\exp(-rn\sigma) \tag{132}$$

By applying the Tyndall effect to this formula, the effect of atmospheric haze on the appearance of distant objects can be simulated. The Tyndall effect notes that very small objects, such as suspended dust, will scatter blue light much more than they will red. Thus, red light will pass through such a medium without appreciable change in intensity, and most blue light will be scattered. In color-shaded relief displays, the bluish-purple cast of distant mountains may be simulated by generating unattenuated shaded relief images for all colors, except blue or purple. In these image components, the above model for haze will be used. The result should be a moderately bright, relatively uniform, bluish tint of low contrast over distant objects.

From the above model of atmospheric haze, the effects of such haze can be more accurately predicted than with previous models. Although only slightly more complicated in form, this model should provide much more qualitative information to the viewer of a shaded relief image. As such, the model should ease the problem of interpreting such images. In particular, the bluish-purple cast of distant mountains can be accurately modeled, thus providing subtle distance clues otherwise unavailable. This will enable the accurate generation of color-shaded relief images.

The algorithms discussed in this report have been coded in two FORTRAN routines. One set of routines, named SSLPLP, generates orthonormal shaded relief images; and the other, named SHDPER, generates perspective shaded relief images. To test the algorithms and software that have been developed, the two routines use an elevation/slope computation subroutine, ALSLP, that is "hardwired" to a polynomial terrain data base of Cache, Okla. In this section, the results of these tests are discussed.

IV. IMPLEMENTATION AND RESULTS

A. PERSPECTIVE SHADED RELIEF ALGORITHMS. In this section, the results will be described of the implementation of the true-perspective algorithms in the shaded relief routines SHDPER. Running on a PDP–11/45 minicomputer, one can create a shaded relief perspective image SHDPER in a multi-user environment in approximately 5 milliseconds per pixel of clock time. The image is subject to enormous variation, dependent on viewing parameters, machine utiliztion, and control parameters. The user of the SHDPER routines can

1. Specify viewer location.
2. Specify imaging geometry.
3. Specify image resolution.
4. Specify the terrain viewed.
5. Specify a vertical exaggeration.
6. Define the position of the sun.
7. Enable local variation solar azimuth within user specified boundaries.
8. Choose between the Lommel-Seeliger law and Lambert's law to define terrain brightness.
9. Define the density scaling.
10. Specify program control parameters affecting processing efficiency.
11. Simulate the effects of atmospheric haze.

The program generates

1. Diagnostic output.
2. A DSKTRN-callable disk file of coded pixel densities.
3. Low resolution line printer plots.

Sample outputs demonstrating these capabilities are shown in figures C1 through C15. These images are each composed of 700,000 pixel (700 pixels by 1000 columns). The disk files containing the image density data were processed using ETL-modified VERSAPLOT Software. To prevent the generation of overly dark, esthetically displeasing, halftone images, only ½ of the available density latitude of the VERSAPLOT software has been used. This was done at the expense of some loss of tone continuity.

The images contained in appendix C are largely self-explanatory. The first image may be considered standard. In succeeding images, the effect of varying a single parameter at a time is demonstrated. The effect of changing the observer's altitude, the focal length of the imagery system, solar elevation, and solar azimuth are successive displayed. The effects of a very light ground haze are next portrayed. In figure C11, the effects of degrading resolution and displayed, which is a 128 by 179 pixel image that has been magnified by bilinear interpolation. Figures C12 and C13 are views of two different areas using the same image parameters that were used for the first image. Figure C14 is a view of the same area seen in the preceeding one, but this image is degraded by a moderate haze. The final image, figure C15, shows the parameters of the first image, but with the local variation of solar azimuth to enhance terrain features. No perspective image was included that illustrated the Lommel-Seeliger law. No satisfactory parameters have been found for such an image.

No attempt was made in coding to optimize the speed and efficiency of SHDPER. Tests on the ETL PDP–11/45, using the parameters of figure C1, indicate that approximately 4.8 milliseconds of clock time are required to process each pixel. This value may be strongly influenced by the parameters governing image generation. In particular, the dependence of execution time upon the sampling density has been investigated for the first image. The relationship is roughly linear, and, for a D (depth of view) of 7.4 map sheet inches (18.8 cm), the total execution time for this image may be estimated by

$$T \text{ (clockminutes)} = 40 + .031 \text{ NPTS} \tag{133}$$

where NPTS is the number of sample elevation data points generated along the radial elevation profile. This formula should not be considered to be generally applicable, and may not be valid for NPTS $\leqslant$ 200.

B. ORTHONORMAL SHADED RELIEF ALGORITHMS. In this section, the implementation of the orthonormal shaded relief in the program SSLPLP is discussed. From these routines, an orthonormal shaded relief image is produced that is defined by various input paramters from a polynomial terrain model. When the routines are run on the ETL PDP−11/45 computer, one can produce an image in approximately 3.3 milliseconds per pixel. However, the image is subject to variations as noted in the introduction to the SHDPER routines.

The user of the SSLPLP routines can

1. Specify the area to imaged.
2. Specify the resolution of the image.
3. Specify the vertical scaling factor for the terrain.
4. Specify the shading law to be used to model the terrain.
5. Specify the position of the "sun".
6. Define a variation in solar position to highlight terrain features.

From the routines one can generate

1. Diagnostic output.
2. A DSKTRN − Callable disk file of coded pixel densities.
3. Low resolution line printer plots.

For sample outputs demonstrating these capabilities, see figures C10 through C21. As with the perspective images, the examples were generated from disk files containing coded image density data by the ETL-modified VERSAPLOT software. The exception of figure C20 must be noted, since this image was generated by means of a half-tone algorithm developed by R. Rosenthal of ETL.[37]

---

[37] Personal Communication, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia.

The orthonormal images contained in figures C10 through C22 should be easy to interpret. Figure C16, which might be considered a "standard," is an orthonormal view of much of the terrain seen in the first perspective image (the perspective view is from the left of the orthonormal image, looking across the orthonormal view): the image contours 600 by 900 pixels. The second orthonormal image uses the same variation of solar azimuth used in the corresponding perspective view (figure C15). Figure C18 is the same image, upon which contour lines generated by the program SIMCON[38] have been overlaid by ETL-developed software.[39] The next two images illustrate the relief contour option of SSLPLP, with and without local variation of solar azimuth (figures C19 and C20). These two figures display a portion of the area of the previous orthonormal images. The final orthonormal image illustrates the use of the Lommel–Seeliger law (figure C21). The image is dark and of very low contrast.

As with the perspective software, no attempt was made in coding to optimize the speed and efficiency of SSLPLP. From test runs, the execution time is approximately 3.3 milliseconds per pixel for images some 540,000 pixels. This time shouldn't vary appreciably with any variation of parameters, but no extensive tests of this have been made.*

---

*The algorithms decribed in this report have been substantially improved since the first draft of this report. A modified GRNDPT routine has reduced execution times to an average of 40 percent of the above figures. A further modification, exploiting the redundant processing of previous approaches, has reduced execution times by an additional factor DF ~2.5.

[38] J. Jancaitis, *Modeling and Contouring Surfaces Subject to Constraints*, University of Virginia, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, ETL–CR–74–19, January 1975, AD–A010 406, pp. 163–165.

[39] R. Rosenthal, *User's Guide to MERGE*, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, to be published.

V. DISCUSSION

The problems associated with generating shaded relief images have been systemtically investigated. Also, the theory of the formation of gray shade images by optical systems have been investigated, as has the theory of the scattering of light by solid surfaces with mathematical models having been successfully developed. The theoretical conclusions of this investigation have been used to develop a series of algorithms for the efficient generation of a wide variety of shaded relief images. Special algorithms designed to address the specific cartographic problems of enhancing terrain features, modeling haze, and generating relief contours have also been developed. These algorithms have been implemented in two sets of FORTRAN routines: SHDPER, to generate shaded perspective images; and SSLPLP, to generate orthonormal shaded relief images and relief contours. These programs have been tested using a polynomial terrain data base of the Cache, Okla. The images that have been generated demonstrate the feasibility, economy, and promise that shaded relief images hold for future cartographic applications and for tailored specific user need (see tables 2 – 6).

VI. CONCLUSIONS

It is concluded that: 1. This effort has resulted in versatile and efficient shaded relief software that can now serve as the basis for future studies into terrain data user special requirements and needs. 2. More work is needed in two main areas, improved presentation and product use studies. 3. The improved presentation should include color CRT and incorporate non-hypsometric terrain data.

TABLE 2. SHDPER FTN

Function/Source Code Correlation

Description of Program Function	**Compilation Line Number**
Overhead	1 - 26
Interactive Parameter Entry	27 - 44
Parameter Computation	45 - 84
Loop Over Columns of Image	85 - 153
Calculation of Radial Profile (CALL PTS)	86 - 99
Calculation of Densities of Current Column of Pixels	100 - 145
Determination (CALL GRNDPT, CALL ALT) Object/Image Point Correlation/Slope	110 - 115
Density Computation	116 - 145
Write to Disk File	146 - 153
Write to Line Printer (If appropriate)	154 - 162
Overhead	163 - 17[illegible]

TABLE 3. ALSLP.FTN

Function/Source Code Correlation

Program Function	Compilation Line Number
Overhead	1 – 13
Read Coefficients if first call	14 – 32
Retrieve needed coefficients	33 – 53
Calculate Z	54 – 67
Calculate $\partial Z_i/\partial X$, $\partial Z_i/\partial Y$	68 – 76
Calculate W_i	77 – 83
Calculate $\partial W_i/\partial X$, $\partial W_i/\partial Y$	84 – 92
Calculate Z, $\partial Z/\partial X$, $\partial Z/\partial Y$	92 – 98
Overhead	100 – 111

TABLE 4. Outline of GRNDPT Source Code Function

Function	Compilation Line Numbers
Initilize Flags, Overhead	1 – 13
$Y(I) < Y_s$ (Y(I + 1)?	14 – 23
Calculate 1st iteration parameters	24 – 40
Calculate projection of algorithmically determined point	41 – 44
Set parameters for next iteration, if needed	45 – 59
Return	62 – 63

TABLE 5. The Angular Width of the Image as a Function of DIS

DIS (")	Total Angular Width of Image
4.	122°
6.	101°
8.	84°
10.	72°
12.	62°
14.	55°
16.	49°
18.	44°
20.	40°
24.	34°
30.	27°

TABLE 6. SSLPLP.FTN

Function/Source Code Correlation

Description of Function	Compilation Line Number
Overhead	1 - 25
Interactive Parameter Entry	26 - 55
Parameter Computation	56 - 97
Loop Over Rows of Image	98 - 148
Calculation of Slopes for row (CALL SLPS)	101 - 110
Calculation of Densities: Lambert's Law	105 - 120
Calculation of Densities: Lommel-Seeliger Law	122 - 139
Write Densities to Disk File	144 - 147
Write to Line Printer (If appropriate) (CALL LP)	149 - 158
Overhead	159 - 169

APPENDIX A.

1. **Theoretical Details.** Lambert's law is a scattering law that is an idealization of empirical evidence. As noted in this report, no good derviation of Lambert's law exists that is based on assumptions about the microscopic structure of materials obeying it. This appendix presents an inadequate derivation of Lambert's law. This derivation has two components, just as Lambert's law may be thought of as having two components:

1. The brightness of a surface obeying Lambert's law is independent of the direction of view.
2. The brightness of such a surface is proportional to the cosine of the angle between the vector of illumination and the normal to the surface.

The first component of Lambert's law accurately describes the functional dependence of brightness of self-luminous objects as well as that of some reflecting bodies. Consequently, the derivation given below first postulates a model for the microscopic structure of a self-luminous medium, and then shows that the surface of a body described by this model will be of constant brightness, regardless of the orientation of the surface to an observer.

The derivation of the second component of Lambert's law is not satisfactory. In essence, this portion of the derivation is an attempt to justify the third assumption given below about self-luminous bodies for illuminated reflecting bodies. This attempt is not successful in a rigorous mathematical sense, but is included for completeness of this appendix. It must be noted that this attempt is the author's: better "derivations" may exist, but they are not known to the author at the time of writing.

First, the variables involved in the derivation will be defined. As can be seen in figure A1, the system modeled might be used to measure the brightness of the surface **S**. These measurements would be based on the determination of the total flux of radiant energy incident upon the light-sensitive area, φ, of a photodetector. This detector has a light acceptance cone of width **dw**, the center of which is inclined at an angle θ to the normal to the surface **S**. The distance of the detector from the surface is **R**, measured along a ray within the detector acceptance cone, and the corresponding distance to some volume element **dV** is **r**, located a distance $z = (r - R)\cos\theta$ below the surface.

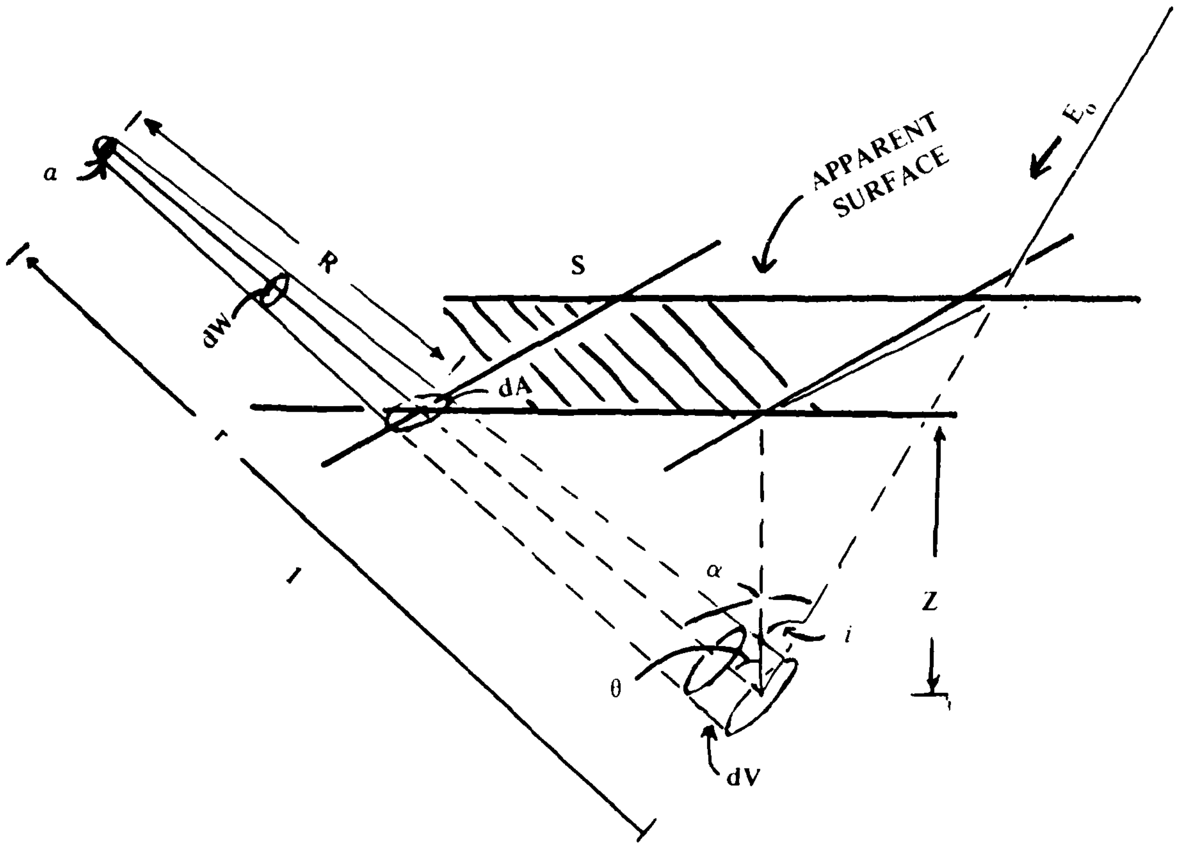


FIGURE A1. Definition of Variables for Lambert and Lommel–Seeliger Derivation.

A derivation such as this must be motivated by empirical evidence. If self-luminous bodies obey the first aspect of Lambert's law, then the body with surface **S** will be first modeled as a homogeneous, self-luminous medium. These assumptions can be easily modified to treat non-radiant bodies that are accurately modeled by Lambert's law, as will be discussed. The medium presently being considered is characterized by three features:

1. Each infinitesimal volume element radiates equally in all directions.
2. A ray of light has a mean free path of length before being absorbed or re-scattered, and thus lost to a given beam.
3. All volume elements are equally luminous, with luminosity per volume I_0.

It is a well known result of statistical mechanics that a beam of light characterized by flux **F** will, in such a medium, be attenuated by a factor of **dF** in traversing a distance **dx**, such that

$$dF = -F(1/\tau)\,dx \tag{134}$$

Thus, a beam of initial intensity F_0 will be of intensity $F(x)$ after traversing a distance x, such that

$$F = \int_0^x dF = F_0\, e^{-x/\tau} \tag{135}$$

Consider now a volume element a distance z beneath the surface. A light ray emitted from this element towards the detector will traverse a path of length $x = (r - R) = z \sec\theta$ before emerging from the surface. Now, $F_0 = I_0\, dV\, d\Omega$, where in this case, I_0 is the luminosity per unit volume, **dV** is the differential volume element, and **dΩ** is the solid angle subtended by the detector at the volume element. Hence the differential flux incident upon the detector from this volume element will be

$$dF = I_0\, dV\, d\Omega\, e^{-(r - R)/\tau} \tag{136}$$

To find the total flux of radiant energy upon the detector, one must integrate all the volume elements within the detector acceptance cone **dW**. Hence,

$$F = \int dF = \int_{r=R}^{\infty} I_o \, e^{-(r-R)/\tau} \, dV \, d\Omega \tag{137}$$

Now, $d\Omega = a/r^2$, and $dV = r^2 \, dW \, dr$. Hence,

$$F = \int_{r=R}^{\infty} I_o \, (a/r^2) r^2 \; dW \, e^{-(r-R)/\tau} \, dr$$

$$= \int_{r=R}^{\infty} I_o \, a \, dW \, e^{-(r-R)/\tau} \, dr$$

$$= I_o \, a \, dW \, \tau \tag{138}$$

Thus, the flux received by a detector is independent of the inclination of the detector to the surface (i.e. independent of θ), if all else is held constant. The apparent surface is a function of angle

$$dA = R^2 \, dW/\cos(\theta) \tag{139}$$

The solid angle that the light subtends is approximately constant:

$$d\Omega = a/r^2 \simeq a/R^2 \tag{140}$$

This is a valid approximation, if $R \gg \tau$, since all light emerging from the surface effectively emerged from a radius $r \lesssim (R + 10\gamma) \approx R$. This follows from the exponential decay of the emerging flux: less than 1% of the emerging light traverses a distance of more than 10τ. Having made this approximation, one can keep the brightness of the surface constant:

$$B = \frac{F}{dA \cos\theta \, d\Omega}$$

$$= F/[R^2 \, (dW/\cos\theta) \cos\theta \, (a/R^2)]$$

$$= F/(a \, dW)$$

$$= I_o \tau \tag{141}$$

The brightness of a surface obeying lambert's law is defined as being independent of the direction from which it is viewed, which is the case of the surface of the body just modeled. The first aspect of Lambert's law has thus been derived from microscopic considerations for a self-luminous body, in a form relating the brightness of the surface of the body to the flux incident upon a photodetector.

To complete a derivation of Lambert's law, one must justify the third assumption for reflecting bodies. This can only be done by making an additional assumption about the microscopic structure of such bodies. In particular, reflecting bodies that obey Lambert's law are composed of miscroscopic particles, each of which is

1. Non-absorbing.
2. Defuses light uniformly in all directions.

It must be shown that these assumptions lead to the following conclusions: In a semi-infinite body which may be modeled by these assumptions, (1) the intensity of the light with the body is everywhere constant, and (2) the intensity of the light is proportional to the cosine of the angle incidence of the incoming light.

First, examine a differential volume element **dV** a depth **Z** below the surface (see figure A1). By showing that the net flux upon this volume element is independent of **Z**, one assumption can be justified. Considerations of flux out (which, since no absorbtion is assumed, must equal flux in), will define the value of the constant.

The flux upon the volume element can be defined as an integral with two components; the flux of the incident light reaching volume element from the surface, and the flux scattered from the rest of the body. This incident light is assured to be of intensity E_0 above the surface and incident at an angle to the normal to the surface. By analysis similar to that above, the fraction of the incident light reaching **dV** will be

$$E_{1\ onto} = \pi E_0 \Delta^2 e^{-Z \sec i/\tau} \qquad (142)$$

where it has been assumed that the volume element is spherical and has a radius Δ ($dV = 4\pi/3\ \Delta^3$; $dA = \pi \Delta^2$). Of this, a fraction $\Delta/3\tau$ will be scattered. Thus, F_{out} of this fraction is

$$I_{1\,out} = \frac{\pi}{3\tau} E_o \Delta^3 e^{-Z\sec i/\tau} \tag{143}$$

The contribution from scattered light from the rest of the body must now be considered. Here the differential contribution for a volume element **dV'** of luminous intensity $\mathbf{I}_o$ (assured constant, we must find $\mathbf{I}_o$ in terms of $\mathbf{E}_o$ and i) at a distance **r** will be

$$dF_{2\,out} = I_o \frac{\pi \Delta^2}{r^2} e^{-r/\tau} dV' \tag{144}$$

where $\pi \Delta^2/r$ represents the solid angle subtended by **dV** at a. As above, a fraction of $\Delta/3\tau$ of this flux will be scattered. Thus,

$$dF_{out} = I_o \frac{\pi \Delta^3}{3\tau r^2} = e^{-r/\tau} dV' \tag{145}$$

This must be integrated over the volume of the body. Since the surface delimiates the body, the limits of integration are:

$$F_{2\,out} \quad \int_{r=0}^{\infty} \int_{\varphi=0}^{2\pi} \int_{\theta_2}^{\pi/2} I_o \frac{\pi \Delta^3}{3r^2 \tau} e^{-r/\tau} dV'$$

$$+ \int_{0}^{Z\sec\varphi} \int_{0}^{2\pi} \int_{0}^{\pi/2} I_o \frac{\pi \Delta^3}{3r^2 \tau} e^{-r/\tau} dV' \tag{146}$$

where, in spherical polar coordinates,

$$dV' = r^2 \sin\theta \, dr \, d\theta \, d\varphi \tag{147}$$

The first term can be readily integrated by standard techniques. The second cannot be evaluated in closed form; hence, approximations must be resorted to if it is to be evaluated. This alone precludes the success of the approach, since in the absence of a closed form representation, it cannot be shown that the result will contain a term negating the Z dependence of F_1. Strict limits can be placed on the Z dependence of both F_1 and the non-integrable term of F_2, but these are not adequate. They are inconsistent with the assumptions necessary for the derivation of the first "aspect" of Lambert's law. Consequently, this "derivation" must be considered unsuccessful.

2. **The Lommel-Seeliger Law.** The Lommel-Seeliger law is a scattering law derived from realistic physical assumptions about the microscopic structure of a scattering medium. From this law, accurate predictions can be made about the light-scattering behavior of many natural surfaces. In particular, a scattering surface is assumed to be apparent and that it is actually the surface of a material body composed of many small, uniform scattering bodies. It is also assumed that the interaction of light with the body is amenable to statistical analysis.

The present derivation of the Lommel-Seeliger law is an extension of the treatment given by Hapke.[40] Since the derivation assumptions about the nature of the scattering body are. Explicit, the approach is representative of a large class of similar theoretical laws and is included as a detailed examination of one of these laws. As noted in the beginning of this section, the Lommel-Seeliger law was selected for the simplicity of its final mathamatical form and theoretical basis and for its ability to predict accurately the light-scattering properties of a variety of surfaces.

The following variables are used in the discussion that follows (see figure A1).

Definition of Variables

n	=	mean number of scattering objects per unit volume;
σ	=	average cross section of a scattering object;
γ	=	mean free path of a beam of light rays in the medium. We shall assume that $\gamma = 1/n\sigma$ is always a good approximation.
b	=	total reflectivity of a scattering object. Hence, $(1 - b)$ is equivalent to the fraction of light incident on a scattering object which is absorbed.
i	=	the angle of the incident light with respect to the normal to the apparent surface;
α	=	angle between the rays of incidence and reflection;
θ	=	angle of the ray reflected towards the detector with the normal to the apparent surface;
$d\Omega$	=	solid angle subtended by the detector at the surface;
a	=	light-sensitive area of the detector;
R	=	distance of the detector from the apparent surface, measured along the path of the reflected ray $(d\Omega \simeq a/R^2)$;
dW	=	solid angle of the acceptance cone of the detector;
E_0	=	incident intensity (flux of radiant energy per unit area normal to the direction of incidence); and
$\Sigma(\alpha)$	=	scattering law of an individual object. Of the light reflected, $\Sigma(\alpha)$ is that fraction of the incident light from a unit solid angle about the source reflected into the unit solid angle about the direction of observation. $\Sigma(\alpha)$ is normalized so that $\int_{4\pi} \Sigma(\alpha)\, d\Omega = 1$.

[40] B. Hapke, *J. Geophys. Res.* 68(1963), p. 4575.

Consider the fraction of radiation incident upon the apparent surface reaching a volume element $dV = r^2\, dW$ at some distance $Z = (r - R)\cos\theta$ below the apparent surface. If the radiation is incident at an angle i to the normal to the apparent surface, then the distance traversed by the ray will be

$$Z' = Z/\cos i = Z\, \mathrm{Sec}\, i \tag{148}$$

The fractional change in the flux incident upon an infinitesimal volume element of the scattering material of thickness **dZ'** (measured along the path of the incident ray), owing to scattering or absorption within the differential volume element, will be

$$dE = -E'\, n\sigma\, dZ' \tag{149}$$

Hence,

$$E(Z') = E_o\, e^{-n\sigma Z'} \tag{150}$$

and

$$E(Z) = E_o\, e^{-n\sigma Z\, \mathrm{Sec}\, i} = E_o\, e^{-(Z\, \mathrm{Sec}\, i)/\tau} \tag{151}$$

The light reaching the detector is the light within the detector acceptance cone, **dW**. This light may be interpreted as being reflected from the apparent surface area $dA = R^2 dW/\cos\theta$.

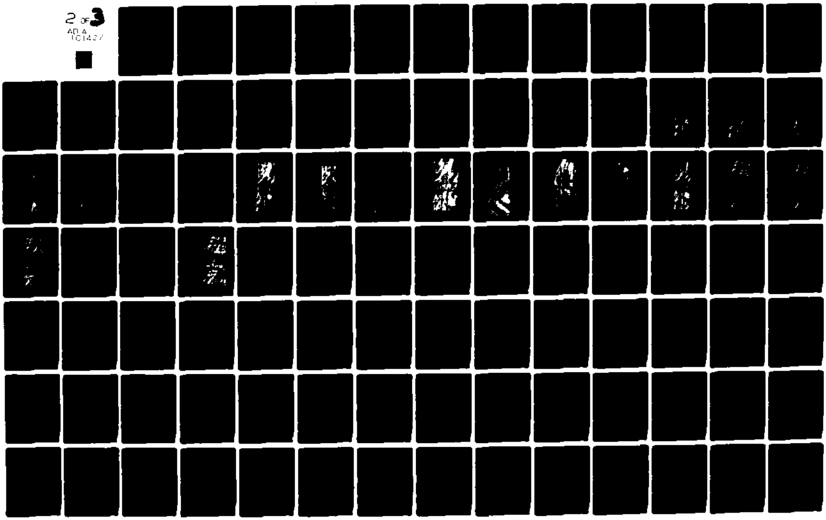
Consider the volume element **dV** a distance **Z** below the apparent surface. As above, the light intensity at the **dV** will be

$$E(Z) = E_o\, e^{-(Z\, \mathrm{Sec}\, i)/\tau} \tag{152}$$

Now, determine the differential flux incident upon the detector owing to scattering within this volume element. First, not all light incident upon the volume element will be reflected. Instead, only the light incident upon the surface area $dS = n\sigma\, dV$ of the scatterers within the volume will be available for reflection. Of this, only a fraction **b** will be reflected. The detector is located at an angle α to the radiation incident on the scatters; $\Sigma(\alpha)$ of the total reflected radiation will fall within a unit solid angle around the detector. However, the detector doesn't subtend a unit solid angle. It subtends a fraction $d\Omega = a/r^2$ of a unit solid angle. The product of these factors represents the flux incident upon the detector:

AD-A101 422 ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA F/G 8/2
SHADED RELIEF IMAGES FOR CARTOGRAPHIC APPLICATIONS.(U)
APR 81 C C TAYLOR
UNCLASSIFIED ETL-0259 NL

2 OF 3
AD A
101422



$$dF = E_0 (a/r^2)\ b \sum(\alpha)\, n\sigma\, e^{(Z\ Sec i)\ \tau}\ dV \qquad (153)$$

Assume that the light reflected towards the detector will be further attenuated by a factor of $\exp[(-Z\ Sec\theta)\ \gamma]$ before emerging from the apparent surface, since it must again pass through the scattering body. Hence, the flux of radiant energy reflected from the volume element **dV** and emerging from the surface will be the product of equation (153) and the new attenuation factor. Thus,

$$dF = E_0 (a/r^2)\ b \sum(\alpha)\, n\sigma\, e^{(Z\ Sec i)\tau}\, e^{(Z\ Sec\theta)\tau}\ dV$$

$$= E_0\, n\sigma a\, b \sum(\alpha) \exp[-(r-R)(1+\cos\theta/\cos i)\ \tau]\ dr\, dW \qquad (154)$$

where the identities $Z = (r - R)\cos\theta$ and $dV = r^2\, dW\, dr$ has been used.

Thus, the total flux of radiant energy falling upon the detector within the detector acceptance cone **dW** will be found by integrating the differential flux from each $dV = r^2\ dW\ dr$ within the acceptance cone **dW**; i.e. by integrating over all **r** greater than **R**:

$$F = \int_{r=R}^{\infty} dF$$

$$= \int_{r=R}^{\infty} E_0\, n\sigma\, ab \sum(\alpha)\, dW\, e^{-[(r-R)(1+\cos\theta/\cos i)\tau]}\ dr$$

$$= \left(\frac{\tau}{1+\cos\theta/\cos i}\right)(E_0\, n\sigma\, a\, b \sum(\alpha)\, dW)$$

$$= \frac{E_0\, a\, b \sum(\alpha)\, dW}{1+\cos\theta/\cos i} \qquad (155)$$

As mentioned before, this report is not interested in the total incident flux in photometric applications, such as photometry, but is interested in the photometric brightness. Recalling the basic photometric identity and invoking the geometric law of conservation of energy, one has

$$B = F/(dA \cos\theta \, d\Omega)$$

$$= F/[dA \cos\theta \, (a/R^2)]$$

$$= \frac{F_0 \, a \, b \sum(\alpha) \, dW}{1 + \cos\theta/\cos i} \quad \frac{R^2}{a} \quad \frac{1}{dA \cos\theta}$$

$$= \frac{F_0 \, a \, b \sum(\alpha)}{1 + \cos\theta/\cos i} \quad \frac{dA \cos\theta}{R^2} \quad \frac{R^2}{a} \quad \frac{1}{dA \cos\theta}$$

$$= \frac{F_0 \, b \sum(\alpha)}{1 + \cos\theta/\cos i} \tag{156}$$

This is the form of the Lommel-Seeliger scattering law of interest to us.

3. **Image Element Illumination.** Let us repeat the problem addressed by this report: that is the determination of the densities of each of an array of image elements of a clearly defined optical system for the specified lighting conditions of a given terrain model.

The first problem to be considered is what each image element is to represent. Ideally, each element will reproduce the average illumination of that image element. Mathematically, this is a continuous process, defined by the integral of the illumination at each point of the image element, divided by the finite area of this element. Thus, the average illumination is given by

$$\langle F \rangle = \frac{\int_A dF}{A} \tag{157}$$

where **A** is the area of the image element, and **dF** is the illumination of a differential image area element, **dS'**. Recalling equation (41) and remembering that there is a 1:1 mapping of the differential surface area of the object into the corresponding element **dS'**, one sees that equation (43) can be written as

$$
\langle E \rangle \approx \frac{\displaystyle\int_{S'} \frac{a\, B \cos\theta \cos\varphi \, dS}{Mo^2 \cos\alpha \cos\beta}}{+A} \tag{158}
$$

where S' is the corresponding area of the object.

It is this continuous integral that we must approximate by a discrete sum. If we let E_i represent the illumination of the i^{th} point of the image element under consideration, then

$$
\langle E \rangle \approx \frac{\sum_{i=1}^{n} E_i}{n} \tag{159}
$$

If the i^{th} point of the image element is defined in terms of a local (x, y) coordinate system by $(X_i\ Y_i)$, and the corresponding point of the objects surface by (η_i, ξ_i) in terms of some (η, ξ) coordinate system on the surface of the object, then $B = B(\eta,\xi)$ and particular points can be defined as B_i. Similarly, θ and φ can be labeled by θ_i and φ_i. Then

$$
\langle E \rangle \approx \frac{\displaystyle\sum_{\epsilon=1}^{n} \frac{B_i\, a \cos\varphi_i \cos\theta_i}{Mo^2 \cos\alpha \cos\theta}}{n} \tag{160}
$$

For realistic imaging systems, one may assume that θ will not vary appreciably over the image element. A more dubious assumption, although well-justified for image elements of sufficiently small areas, is that B_i and φ_i will also vary by a negligible amount over the area of interest. Thus, we have

$$
\langle E \rangle \approx \frac{a\, B_i \cos\varphi_i \cos\theta_i}{Mo^2 \cos\alpha \cos\beta} \tag{161}
$$

for some typical B_i and corresponding θ_i and φ_i within the area of interest. Ideally, this point should be at or near the center of our image element, but in light of the approximations introduced thus far, this is not essential.

Finally, the approximation is introduced so that the field of view of the imaging system is small, and that for all elements of the image, θ_i XO. However, it is an approximation only for the perspective projection; it is exact for the orthonormal projection. Having introduced this approximation and noting that

$$\cos\varphi \approx \cos\alpha \cos\beta \tag{162}$$

one has the formula for the "average" illumination $< E_j >$ of the j^{th} area element of the image, which contains a typical point with a corresponding point on the surface of the object characterized by a brightness B_j.

$$< E_j > = \frac{a B_j}{Mo^2} = \alpha B_j \tag{163}$$

This approximation is introduced for two reasons. First, the number of calculations will be significantly reduced that need to be made to determine the densities of a synthetic gray shade perspective image. Second, the final perspective should be composed of image elements with densities that are well defined functions of the terrain and lighting and that are not dependent on the arbitrary viewing parameter. In essence, the approximations introduced remove the vignetting effects that the rigorous equation accurately models. These effects would probably detract from the qualitative appearance of the image.

APPENDIX B.

1. **Software Guide.** This appendix is a user's guide to the ETL shaded relief software developed during the course of the work described in this report. The software was developed in FORTRAN on a PDP–11/45 minicomputer under an RSX–11/D operating system. The software is currently running on the ETL PDP–11/45 under an RSX–11/M operating system. Familiarity with the body of this report is desirable before reading this appendix, and limited familiarity with FORTRAN and the RSX–11/M operating system is assumed.

This appendix present the information necessary to use the three sets of routines developed. First, the perspective shaded relief routine SHDPER.TSK procedures that are necessary to create the executable file from the FORTRAN source files will be examined, explaining the input parameters and detailing the format of the polynomial elevation data base, the output line printer graphic, and the coded-density disk file. The orthonormal shaded relief routine, SSLPLP.TSK will be examined. Finally, the program PDSKTP will be reviewed. The program reads the coded density data from the disk file created by either SSLPLP or a SHDPER and writes the data to a nine-track tape to generate high resolution images, using either the DICOMED or EBR gray shade recording devices available at ETL.

2. **Perspective Routines.** In this section, the use of the true-perspective shaded relief routines SHDPER will be described. Running on a PDP–11/45 minicomputer, SHDPER can created a shaded relief perspective image in a multi-user environment in a time on the order of 5 milliseconds per pixel real time. This figure is subject to enormous variation, dependent on viewing parameters, machine use, and control parameters. The user of the SHDPER routines can

1. Specify viewer location.
2. Specify imaging geometry.
3. Specify image resolution.
4. Specify the terrain viewed.
5. Specify a vertical exaggeration.
6. Define the position of the sun.
7. Enable a variable sun position within user-specified boundaries.
8. Choose between the Lommel-Seeliger law and Lambert's law to define terrain brightness.
9. Define the density scaling.
10. Specify program control parameters affecting processing efficiency.
11. Simulate the effects of atmospheric haze.

The program generates

1. Diagnostic output.
2. A DSKTRN–callable disk file of coded pixel densities.
3. Low resolution line printer plots.

a. Building the Task: SHDPER.TSK. In this section, the procedures are examined for creating the file SHDPER.TSK to be executed on an RSX–11/M operating system. The user should know how to create files of the source code, as listed in appendix B, as well as the FORTRAN–callable I/0 routine DSKTRN.

(1) FORTRAN Source Code File. In this section, the FORTRAN source code files needed to create the task SHDPER. TSK are examined. First, the algorithmic function of each of these files will be reviewed, and where appropriate, sufficient information for specific minor modifications will be presented.

(a) SHDPER.FTN. The program SHDPER.FTN is the main routine of the shaded perspective routines. An understanding of the basic structure of the program can be quickly gained by examining the listing and comparing it with the functional outline of the FORTRAN code presented in table 2.

There are two modifications of the file that the users might wish to make: (1) to alter the size of the maximum image that can be produced, and (2) to change the disk file created to another device.

To make the first modification, let M represent the desired maximum number of pixels per column of the image, and let N represent the maximum number of columns of pixels in the image. Thus, M' and N' are the corresponding least multiple of 256 greater than or equal to M/2 and N, respectively. Hence, the byte anag SHADE should be re-deminsioned

LOGICAL*1 SHADE (M)

and the two calls to DSKFIL should be altered as

```
100 CALL DSKFIL (2, -1,' DB:, LPPER.DAT',DUM, M , N )

    CALL DSKFIL (2, -2,' DB1:, LPPER.DAT', DUM, M , N )
```

The dimensions of the Y-array may be altered with that of SHADE. The disk file name, or the device it is to be written on, can be changed by altering the appropriate strings within the apostraphes of the two calls to DSKFIL.

(b) PTS. FTN. The PTS.FTN file contains the subroutine PTS, which determines the coordinates at which elevation data points are to be accessed. The subroutine is compatible with the SHDPER calling routine and the altitude- and slope-computing subroutine ALT found in file ALSLP.FTN. Because it is simple, no outline of function will be given here.

The calling statement is

CALL PTS (X1, Y1, X2, Y2, NPTS, B, DLN, IASCD)

where

(X1, Y1)	are the mapsheet coordinates of the observer's location
(X2, Y2)	are the coordinates of the other end point of the radial of interest.
NPTS	is the number of elevation data points to be generated along this profile and returned to the calling routine.
B	is the array in which the elevation data points are to be stored.
DLN	is, effectively, a dummy parameter for this application.
IASCD	should be set to 0 to minimize execution time.

One COMMON block should be defined in the calling routine (as it is in SHDPER):

COMMON/BOUNDS/XMAXB,/YMAXB, XMINB, YMINB, ELEVB

where

(XMINB, YMINB)	are the mapsheet coordinates of the origin of the area modeled by the polynomial terrain data base.
(XMAXB, YMAXB)	are the maximum mapsheet coordinates of the area modeled by the polynomial data base.
ELEVB	is the minimum elevation of the polynomial data base.

These variables are appropriately defined in the overhead portion of SHDPER. FTN.

(c) ALSLP.FTN. This file contains subroutine ALT. This subroutine accepts the **X** and **Y** mapsheet coordinates of the current point of interest and an integer flag IASCD. When IASCD = 0, only the elevation of the point of interest will be returned. When IASCD is set to 1 in the calling routine, both elevation and the partial derivatives ($\partial Z/\partial X$ and $\partial Z/\partial Y$) at the point of interest will be returned.

The subroutine uses a polynomial terrain data base and currently is structured to read the coefficients of the upper one--third the Cache, Okla mapsheet into the arrays COEF1 (90, 40) and COEF2 (90, 40, 3). By bringing these coefficients into core, execution time is considerably reduced.

With this implementation, only data for points within the region modeled by the coefficients brought into core can be evaluated. The subroutine avoids inadvertent attempts to access non-modeled terrain areas by checking to see that the point of interest is within the region modeled. Points outside of this region will have their z values set to a nominal elevation, and the partial derivatives will be set to 0.

In table 3, the correlation is outlined between the function and the FORTRAN compilation line numbers used in the listing in appendix B.

The calling statement is

CALL ALT (X, Y, Z, SN, AZ, BZ, IASCD)

where

(X, Y)	are the coordinates of the point of interest.
Z	will be the elevation of that data point.
SN	is a dummy parameter.
AZ, BZ	are respectively $\partial Z/\partial X$ and $\partial Z/\partial Y$; IASCD should not be equal to zero if **AZ** and **BZ** are desired.

Two COMMON Blocks must be defined in the main routine.

COMMON/BOUNDS/XMAXB, YMAXB, YMINB, ELEVB

as in

ALSLP, FTN and COMMON/NEW/MRDR

where MRDR should be next to 1 in the main routine.

(d) GRNDPT.FTN. This file contains the image–object point correlation subroutine GRNDPT. The subroutine accepts the number of the current pixel of interest in the column of pixels being processed at the time of the call to the subroutine. On the basis of an array of elevation data points along the radial corresponding to the current column of pixels and the parameters defining the perspective view being generated, the subroutine finds a point along the radial, whose projection lies within a specified distance of the center of the pixel of interest.

The algorithm used (described in section IV, A) is iterative and makes several assumptions about the nature of the terrain modeled. Foremost among these is the assumption of continuity, which is violated at the "edge" of the area modeled by the coefficients in core. Hence, those views of the terrain in which the "cliff" would be visible are not amenable to analysis by the SP (or other) interative routine. The subroutine GRNDPT detects this condition, issues diagnostic error messages, and sets NFLAG to 10, which halts the execution of SHDPER.

In table 4, the correlation is outlined between the function of the source code and the line numbers of the FORTRAN compilation of GRNDPT in appendix C. In the table the notation found in section IV, A, is used.

The calling statement is

GRNDPT (J, Y, NFLAG, XI, YI, I, YMAX, IASCD, AZ, BZ, Z)

The argument and return values are as follows:

J	is the index of the pixel for which we are seeking a corresponding point on the ground. J runs from 1 at the bottom of the screen to NPIX at the top, where NPIX is the number of pixels in each column.
Y	is the array containing the screen Y–values of the projections of the points comprising the radial of elevation data.
NFLAG	is a flag returned by GRNDPT to the calling routine. NFLAG is 1, if a successful correlation has been made, and it is set to 10 if an error condition has been detected.
(XI, YI)	are the coordinates of the point on the ground whose projection lies within the pixel of interest.
I	is an index pointing to the last Y array position with a value less than the last pixel center.
YMAX	is the maximum Y value in the Y array in the range Y(1) to Y(I). Any Y array value greater than YMAX is assumed to be visible.
IASCD	is the flag directing ALT to perform either altitude and slope, or just altitude computations. It should be set to 0 before calling GRNDPT.

Two COMMON Blocks must be defined in the calling routine:

COMMON/AARGH/Y, Y0, THETAP, D, NPTS, DX, H,

YSF, I, SCL, TOL, THETA, Y1ST

and

COMMON/HGRAA/Z1, Z2, CSTP, SSTP, CST, SST

where

X, Y0	are the SHDPER user specified "coordinates of position" (see SHDPER input Parameters below).
THETAP	is the absolute azimuth in radians (aeronautical convention) of the current radial.
THETA	is the azimuth, in radians (aeronautical convention) relative to the principal direction of view.
Y1ST	is the elevation of the point (X, Y0).
DX	is the increment between pixel centers (Defined as 14.5/(number of radials in SHDPER).
H	is the elevation of the observer above ground level (in meters).
VSF	is the vertical scaling factor.
TOL	is the user specified tolerance.
SCL	is the ratio of inches on the image plane to meters on the ground.
D,DIS,NPTS	are as specified in SHDPER Input Parameters.

(e) SXSY.FTN. This file contains the variable sun angle subroutine SXSY. The subroutine varies the azimuthal position of the sun to highlight terrain features, within user-specified tolerances. Since the subroutine is well documented internally with respect to the functions outlined in section III, F, no table of source code function is given.

The calling sequence is

CALL SXSY (SX, SY, AZ, BZ)

where

SX, SY	will be the X– and Y– components of the illumination vector to be used for the current pixel.
AZ, BZ	are the partial derivations $\partial Z/\partial X$ and $\partial Z/\partial Y$, defining the normal at the point of the surface of interest.

One COMMON Block must be defined:

COMMON/SEXY/DELTAG, COSEL, IVSUN, ANG, C1, C2, S1, S2

where DELTAG should be set to 1.

COSEL	is the cosine of the elevation of the sun.
IVSUN	is a flag specifying that the sun angle is to be varied (IVSUN = 1) or is not to be varied (IVSUN = 0).

(f) LPPER.DAT. The file LPPER.FTN contains the line printer output algorithm LP. The subroutine accepts a byte array of arbitrary length and outputs the coded gray shade information of the first 100 array positions. The image format is summarized in section III, E of this report. Each byte should contain a value between 0 and 128. The correlation between density and input value can be found by examining the listing of LPPER.FTN and correlating it with table 1.

The calling statement is

CALL LPER (SHADE)

where SHADE is a byte (LOGICAL * 1) array of arbitrary length.

ANG	is the angle of view (in radians, following the mathematical convention).
C1, C2, S1, S2	are the values of S_x in equations (108, 109) and the values of S_y in equations (108, 109).

(g) LPPER.FTN. This file contains the line printer output subroutine LP. The subroutine is a straightforward implement of the Yoeli density table found in table 1. The calling statement is

CALL LP(SHADE)

where SHADE is a byte array containing the coded image densities.

(2) Compilation and Task Building. Before the SHDPER shaded relief perspective routines can be run, the source code files must be compiled, and the object files created by compilation must be combined into an executable file by the task builder.

Compilation is straightforward. If the user is logged on to the system, then the Main Console Routine (MCR) will issue a prompt:

MCR >

The user should then enter

FOR object file name, LP: = source file name, [CR]

If no line printer listing of the compilation is desired, the ",LP:" may be deleted. An example of this process is

MCR > FOR SHDPER, LP: = SHDPER [CR]

The output file containing the object code will be placed, by default, in the file SHDPER.OBJ.

It should be noted that all of this assumes that the User Identification Code (UIC) of the user is that whose library contains the needed source code files as outlined in this appendix. If not, recourse to the RSX-11/M manuals should be made. In any case, all source code files should be compiled.

At this point, the executable task SHDPER.TSK can be built. This is done by using the indirect file SHDPER.TKB, (listed in this appendix). After the last source file has been compiled, one enters, in response to the MCR prompt

MCR > TKB SHDPER [CR]

This assumes that the FORTRAN --callable data access routine is stored under the UIC (300, 300). If this is not the case, then the command file SHDPER.CMD should be appropriately modified by the user. If the user is unsure as to how to do this, he should review the RSX–11/M operator's manuals. It is also possible that the user may desire alternative device assignments. These may be altered by changing the command file before task building or at the time of the running of the program (see the following section). In table 5, a summary of device assignments and program characteristics is shown.

b. Running SHDPER.TSK. Before discussing input parameters, it is appropriate to discuss the commands involved in initiating the execution of the SHDPER routine, as well as possible error conditions that might occur. Two procedures are available for initiating the execution of the task. Both assume that the task file SHDPER.TSK has been created, that the user is privileged (i.e. has logged on to a privileged UIC and has SET his UIC to that containing the task file), that the data base CACHE1.DAT exists on DB0: (and is unlocked), and that both disk drives (DB0: and DB1:) have been properly mounted.

The first means of initiating execution is, in fact, a subset of the second. In response to an MCR prompt, the user enters

MCR > RUN SHDPER [CR]

Execution should now commence, as outlined in the next section. It is possible, however, for error messages to occur as a result of any of several faulty conditions. The most frequent is related to the files. Either the input data file or the output data file may be locked. It is possible to check this (as well as correct it) by means of the Peripheral Interchange Program (PIP). In response to an MCR prompt, the user should enter

MCR > PIP:CACHE 1 DAT, DB1: = LPPER.DAT/UN

If the files are not locked, PIP will respond with a message to that effect. If this is the case with both files, check to make sure that the files exist on the appropriate devices and are of the right dimensions. If they don't exist, PIP will respond with a message to that effect after the above command. If the files were locked, then no message will be issued; an MCR prompt will be issued, and the user should again attempt to run SHDPER.

It should be noted that the active task created by the RUN *command*, as used above, will be named by the name of the terminal from which the task is excuted. Thus, if the user initiated operation on TT0:, then to abort the task during execution, the user must first bring up MCR by entering a < Control > [CR] command and then entering, in response to the MCR prompt

MCR > ABO TT0 [CR]

Alternatively, the task can be INSTALLED before running, and an alternative name given to the active task. In response to an MCR prompt, the user enters

MCR > INS SHDPER/TASK = SHDPER [CR]

In this example, the task SHDPER is installed under the name of SHDPER. Any other name (6 characters) could be substituted for the second occurrence of "SHDPER". Having installed the task, one may alter the running priority by using the ALT command, or logical unit numbers can be reassigned to other physical devices by means of the REA command. See the RSX-11/M Operating Procedures Manual for details. The installed task is then run as above. The task should be removed after execution by the REM command:

MCR > REM SHDPER [CR]

c. Input Parameters. It is now assumed that the program is executing properly. In this section, the prompts issued by the program in sequential order will be reviewed, and the meaning of each of the input parameters will be explained. All READ statements in the program SHDPER.FTN are list directed. Thus, no attention needs to be paid to considerations of format.

Once the program has begun successful execution, it will issue a prompt

ENTER COORDINATES OF POSITION

The user should enter the X and Y mapsheet coordinate (X, Y 0) of the position from which he wishes to create a perspective image. These coordinates do not have to be within the area modeled by the coefficients in core, but if this is not the case, care should be taken to avoid circumstances such as those described in the discussion of the subroutine GRNDPT in A.1.d. An example of an entry in response to this prompt is

7, 0, 14. [CR]

The shaded perspective routine will now issue another prompt:

ENTER ALTITUDE, AZIMUTH, # RADIALS, PTS/RADIAL,

DIS, # PIX/COLUMN, D

The first value entered should be the altitude in feet of the observer above ground level at the coordinates just entered. This defines the program variable HT. Low values for the altitude (0 to 1000 feet) will result in images with most of the image portraying terrain in the foreground; higher altitudes (1000 – 10,000 feet) will result in substantial coverage of distant objects.

The *azimuth to* be entered (AZ1), will specify the direction in which the imaging system will face. The program accepts any value for this parameter. It assumes that the value entered will be in degrees and will follow the aeronautical convention, such that 0° implies facing North, 90° implies facing East, etc.

The number of radial entered (NRAPS) specifies the number of columns in the resulting image. The actual number of columns in the image created will be (because of the algorithm used)

#Columns = 2 times INT(#radials specified/2) + 1

The number of columns are important in determining the vertical dimensions of the image. The software assumes a 14.5-inch-wide display screen, and a nominal spacing between pixels of DX = 14.5/(#columns–1). This **DX**, when mutiplied by the number of pixel per column, yields the effective vertical dimension of the image. The number of radials must be less than or equal to 1023, and the value entered should be positive.

The number of points per radial entered (NPTS) will determine the number of elevation data points to be generated along each radial profile. The NPTS number should greatly affect the time of execution; however, no study has been done of the actual impact of the value on execution time. It should be greater than 10 times the parameter **D** (see below) and must be less than 1024.

The distance of the image plane from the focal point is defined by DIS. Since the width of the image plane is internally fixed at 14.5 inch, the parameter also defines the angular width of the image. In table 5, a brief list of total angular widths are presented for various values of this parameter. Values of DIS below 10 result in fish–eyed views of the terrain, and values of DIS above 20 have narrow fields of view and are telescopic in nature.

The number of pixels per column (NPIX) defines the height of the image, as noted above. This number must be less than 100 if line printer output is desired, and it must be less than 1024.

Finally, the mapsheet length of the radial of elevation data points generated is defined by D. Thus, only terrain within a radius D of the position of the observer will be imaged. Although D may assume any value, the value chosen should reflect the size of the area modeled.

The values for these seven parameters should be entered sequentially, seperated by commas. For example, one might enter

5000., 0., 200., 10., 100., 10., 100., 9.6 [CR]

The program will next issue the prompt

ENTER VERTICAL SCALING FACTOR, TOLERANCE

The first parameter to be entered, the vertical scaling factor VSF, is the vertical exaggeration to be applied to the terrain. It corresponds to the γ of equation (9) and should take a value between 1 and 5, although any value is acceptable.

The second parameter to be defined, the tolerance TOL, is defined in section III. B. There is no mathematically valid reason for defining TOL less than 0.5. Significantly greater values may under some circumstances result in images of ragged appearance. Thus, 0.5 is the suggested value.

One might enter the following in response to the prompt being considered

3, 0.5

The next prompt asks the user to

ENTER 0 FOR LAMBERTS LAW, 1 FOR

LOMMEL-SEELIGER LAW; DENSITY SCALING FACTOR, ATNTN COEF

The first entry requested defines the light-scattering law to be used in subsequent processing and sets a flag (LAW) in accordance with the entry.

The density scaling factor DSCL defines the factor by which the normalized densities that are calculated by the program are to be multiplied before packing each in an 8-bit byte. Any positive value is acceptable. For the upper one-third of the CACHE, Okla. mapsheet as modeled at ETL, with the sun at altitude of 45°, with Lambert's law and with a vertical scaling factor of three, a good value is 200. A wide latitude is available, and it will be needed for various images. Thus, it is always a good idea to "proof" a proposed image with a line printer plot before generating a high resolution image.

The next input requested defines the attenuation coefficient. This parameter effectively enables the user to define the amount of haze present over the terrain viewed. Since haze is assumed to be of constant density, the contribution of the haze to image illumination increases roughly exponentially with distance. This parameter should be about half the value of D. An entry of 0. results in no haze.

A typical entry in response to this prompt might be

0, 200., 0. [CR]

The final input parameter prompt of SHDPSR asks the user to

ENTER ELEVATION, AZIMUTH OF SUN,

VARIATION OF SOLAR AZIMUTH.

This prompt enables the user to define the illumination of the terrain.

The first two inputs requested define the (principal) position of the sun. The first value requested, the elevation of the sum (ELEV1), is the altitude of the sun relative to a flat base plane. The parameter can assume values between 0° and 90°, such that zero degrees defines the sun to be on the horizon and 90° defines the sun to be directly overhead.

The second input parameter defines the azimuth of the vertical plane in which the elevation of the sun is measured. Any positive value is acceptable. The program assumes that the input will be in degrees and that the user will follow aeronautical conventions. Thus, an entry of 0 would define the sun as due north; and entry of 90 would define it as due east.

The third parameter enables the user to vary the sun angle, as outlined in section III, F of this report. The input requested is the variation in solar azimuth to be used in the algorithm, which defines variable RVIEW1. The program assumes an input in degrees, with any value between 0 and 90 being acceptable. Any value below 1° results in no variation of solar azimuth and values of 90° enables variation through one quadrant.

At the present time, no detailed guidance can be given to the user with regards to the selection of these parameters. Optimum values are highly dependent on terrain, as well as on the other image parameters. A typical entry might be

45., 300., 0. [CR]

After processing the image that the user has just defined, the user will be required to

ENTER 1 FOR ANOTHER IMAGE

An entry of 1 enables the user to redefine parameters as outlined above. Any other entry halts execution.

3. **Orthonormal Shaded Relief Routines.** In this section the orthonormal shaded relief routines SSLPLP will be described. The routines produce an orthonormal shaded relief image defined by various input parameters from a polynomial terrain model. When being run on the ETL PDP-11/45 computer, the routines can produce an image in approximately 3 milliseconds per pixel, subject to variations as noted in the introduction to the SHDPER routines.

The user of the SSLPLP routines can

1. Specify the area to be imaged.
2. Specify the resolution of the image.
3. Specify the vertical scaling factor for the terrain.
4. Specify the shading law to be used to model the terrain.
5. Specify the position of the "sun".
6. Define a variation in solar position to highlight terrain features.

The routines generate

1. Diagnostic output.
2. A DSKTRN–Callable disk file of coded pixel densities.
3. Low resolution line printer plots.

a. Building the Task: SSLPLP.TSK. In this section, the procedures for creating the file SSLPLP.TSK are reviewed. It is assumed that the user has already created files of the source code (as listed in appendix B) as well as the FORTRAN callable disk I/0 routine DSKTRN.

(1) FORTRAN Source Code Files. Before describing the actual procedure to create the SSLPLP task file, the function of each of the FORTRAN source files will be reviewed. Since most of the files have already been described in the SHDPER portion of this appendix, the previous discussion will not be repeated, only referenced.

(a) SSLPLP.FTN. This is the main routine of the orthonormal imaging routines. An understanding of the basic structure of the program can be quickly gained by comparing the listing of SSLPLP.FTN with the functional outline of the FORTRAN code presented in table 6.

SLPS.FTN. This file contains the source code for the subroutine SLPS, which generates an array of x and y slopes at equally spaced points along a profile of the terrain. The subroutine is called by

```
CALL SLPS (S1, Y1, X2, Y2, NPTS, AZA, BZA, DLN)
```

where

(X1, Y1)	are the mapsheet coordinates of the first point of the profile.
(X2, Y2)	are the mapsheet coordinates of the last point of the profile.
NPTS	is the number of points along the profile at which the slopes are to be evaluated.
AZA	is the array that will be filled with the x-slopes $\partial Z/\partial X$.
BZA	is the array that will be filled with the y-slopes $\partial Z/\partial Y$.
DLN	is the distance in mapsheet inches between successive points along the profile. It is computed within SLPS.

One COMMON Block must be defined in the calling routine:

```
COMMON/BOUNDS/XMAXB, YMAXB, XMNB, YMINB, FLEVB
```

The common variables must be defined in the calling routine. They are:

XMAXB, YMAXB	The mapsheet coordinates of the upper right corner of the area modeled by the polynomial data base.
XMINB, YMINB	The mapsheet coordinates of the lower left corner of the area modeled by the polynomial data base.
ELEVB	The minimum elevation in meters of the area modeled.

The subroutine is very similar to the SHDPER subroutine PTS in structure.

(b) ALSLP.FTN, LPPER.FTN, SXSY.FTN. The files ALSLP.FTN, LPPER.FTN, and SXSY.FTN contain the remaining subroutines called by SSLPLP.FTN. All have been described in this appendix.

(2) Task Building. Before the SSLPLP orthonormal shaded relief routines can be run, the source code must be compiled, and the object files created by the compiler must be combined into an executable task file by the task builder. Source files should be compiled as outlined in this appendix.

When the source files have been appropriately compiled, the task file can be created. This is done by entering

MCR > TKB @SSLPLP [CR]

As with SHDPER, the indirect command file SSLPLP.CMD assumes that the FORTRAN--callable disk I/0 routine DSKTRN is stored in the UIC [300, 300]. If this is not the case, appropriate changes should be made in the command file.

b. Running SSLPLP.TSK. Execution of the program should begin. If an error condition occurs, it can be corrected as outlined in the discussion of the initiation of SHDPER in this appendix.

The prompts issued by the program will be reviewed in sequential order, and the meaning of each of the input parameters will be detailed. As with SHDPER, all READ statements in SSLPLP are list directed, and no attention needs to be paid to the format of the input parameters.

Once the program has begun successful execution, it will issue the prompt

ENTER MAPSHEET COORDINATES OF LOWER

LEFT-HAND AREA OF INTEREST

The user should enter the X and Y mapsheet coordinates of the lower left corner (north-up) of the area of interest. This point does not have to be within the area modeled by the coefficients in core, though the unmodeled area will appear as a featureless plane. An entry in response to this prompt might be

5., 15. [CR]

The program will next prompt the user to

ENTER HEIGHT AND WIDTH OF AREA OF INTEREST

Height is the y-extent of the rectangular region to be modeled, and width is the corresponding x-extent. Both are measured in terms of mapsheet inches. A user might enter

4., 4. [CR]

The program next asks the user to

ENTER # PIXELS/HORIZONTAL LINE

The SSLPLP calculates one profile of elevation data points at a time. Each profile has a constant Y-coordinate, with successive profiles running from the upper (greater Y) boundary of the rectangle to the lower boundary. This prompt is asking the user to define the number of pixels in each of the profiles, which defines the pixel size and, thus, the number of profiles. It should be noted that a value of NPIX (the number of pixels per profile) of 100 or less results in the immediate production of line printer output, as well as of the disk file. Values greater than 100 will create only a disk file. A typical entry might be

100 [CR]

The program next asks the user to

ENTER VERTICAL SCALING FACTOR

The number entered will be used to scale the Z-coordinate of the terrain, resulting in an appearance of greater relief. Values between 1 and 10 are reasonable, although the values of other parameters will have an impact on the appearance of the final image. No absolute guidance can be given. This and other parameters must be optimized together, not one at a time. A typical entry might be

3. [CR]

The user next determines the light-scattering law that he will use to model the reflectance of the terrain. The user is asked to

ENTER FOR LAMBERTS LAW, 1 FOR THE LOMMEL-SEELIGER LAW

The prompt is self explanatory.

The final image defining prompt asks the user to

ENTER ANGLE OF ELEVATION OF LIGHT,
DIRECTION OF LIGHT, AZIMUTHAL VARIATION

This prompt plays precisely the same function that the corresponding prompt plays in the SHDPER routines. The user is referred to that section for more information about this prompt.

At this point, the program executes, and the user is asked, upon completion of the image just specified, to respond to

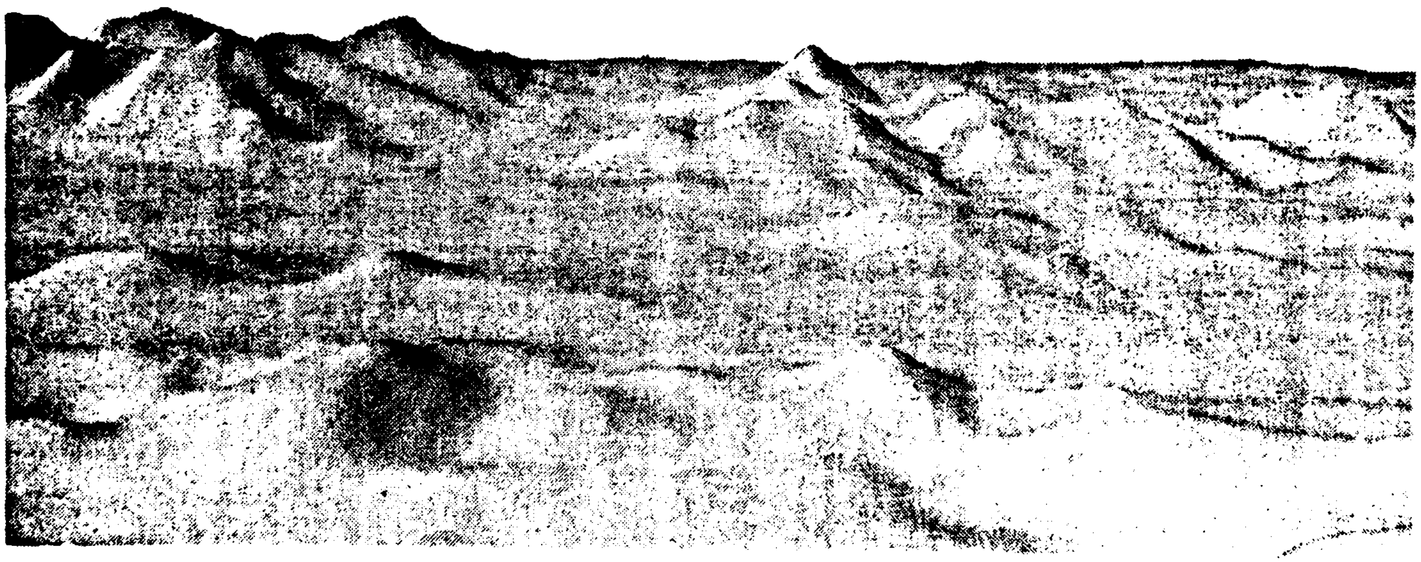
ANOTHER IMAGE: ENTER #1

The user should enter

1. [CR]

in order to continue execution. Any other entry will halt execution.

APPENDIX C.

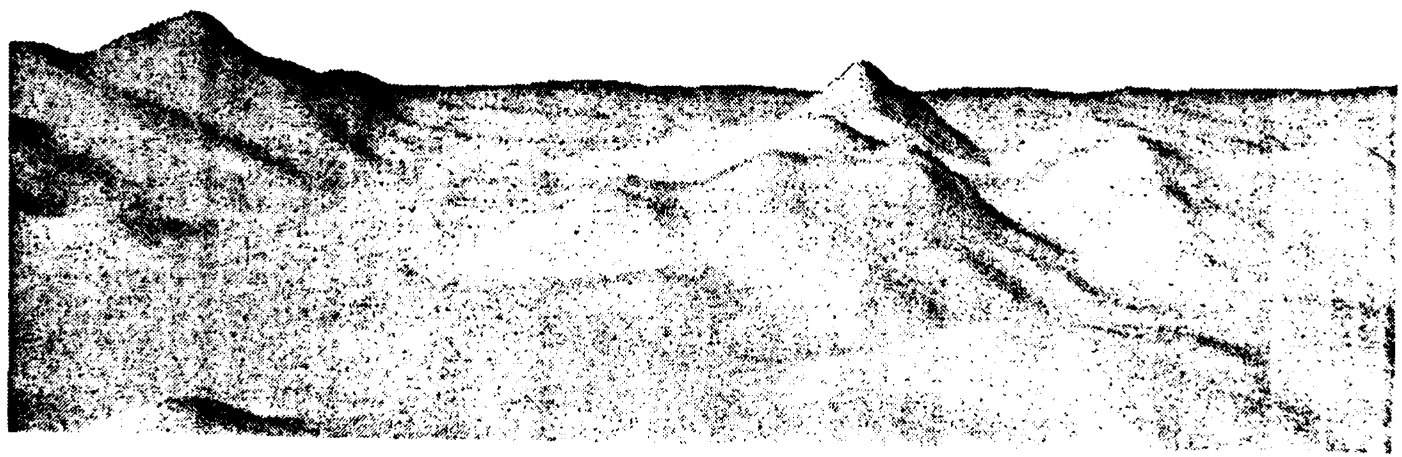


X, Y = 8., 22.	NO HAZE	DIS = 12.
H = 5000.	AZ = 180° (S)	700 X 1000 Pixels
USF = 3.	TOL = 0.5	SUN = 45., 45.
LAMBERT'S LAW	D = 7.4	NO SUN ANGLE VARIATION

FIGURE C1. Nominal Perspective View: Cache, OK.

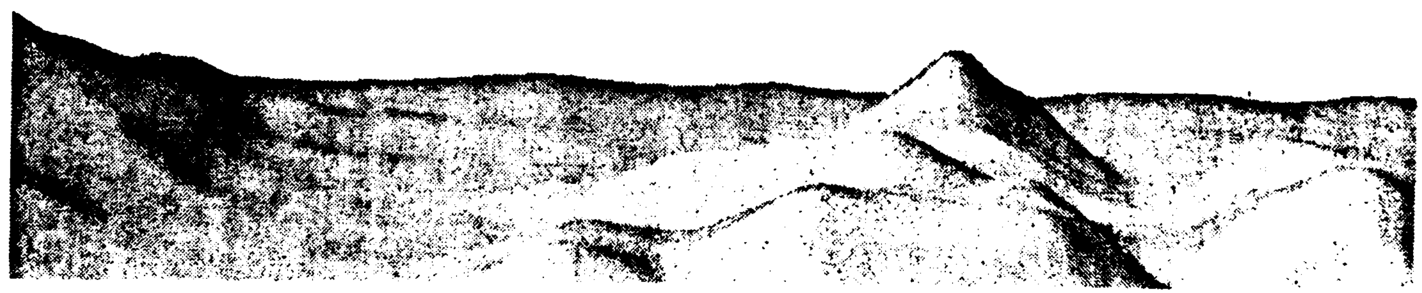
H = 5500

FIGURE C2. Telescopic Perspective View: Cache, OK.



DIS = 2.0

FIGURE C3. Perspective View: Cache, OK.

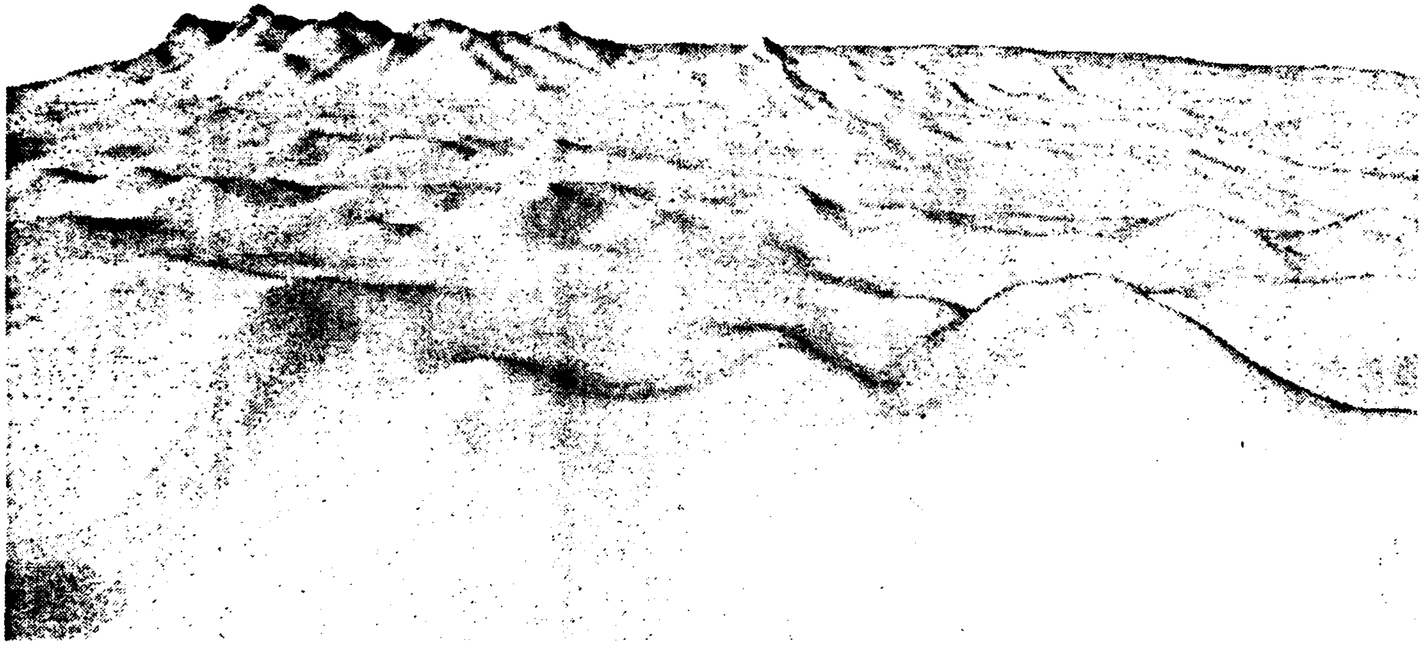


DIS = 30.

FIGURE C4. Telescopic Perspective View: Cache, OK.

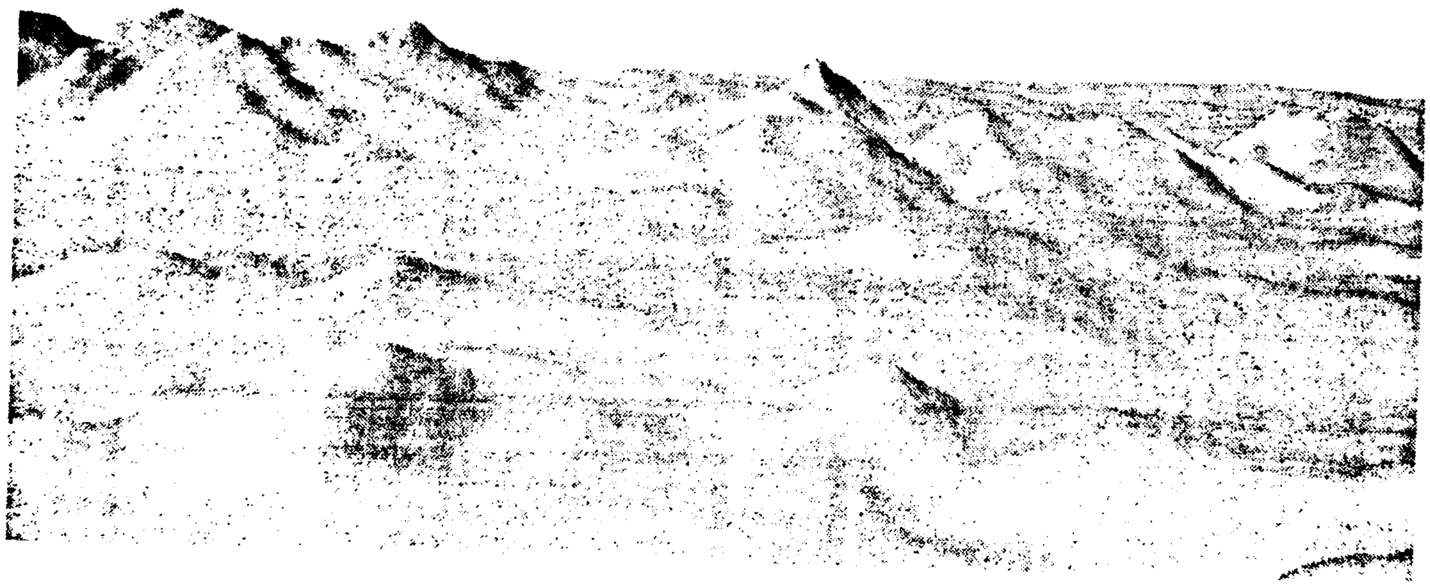
DIS 8

FIGURE C5. Wide Angle View: Cache, OK.



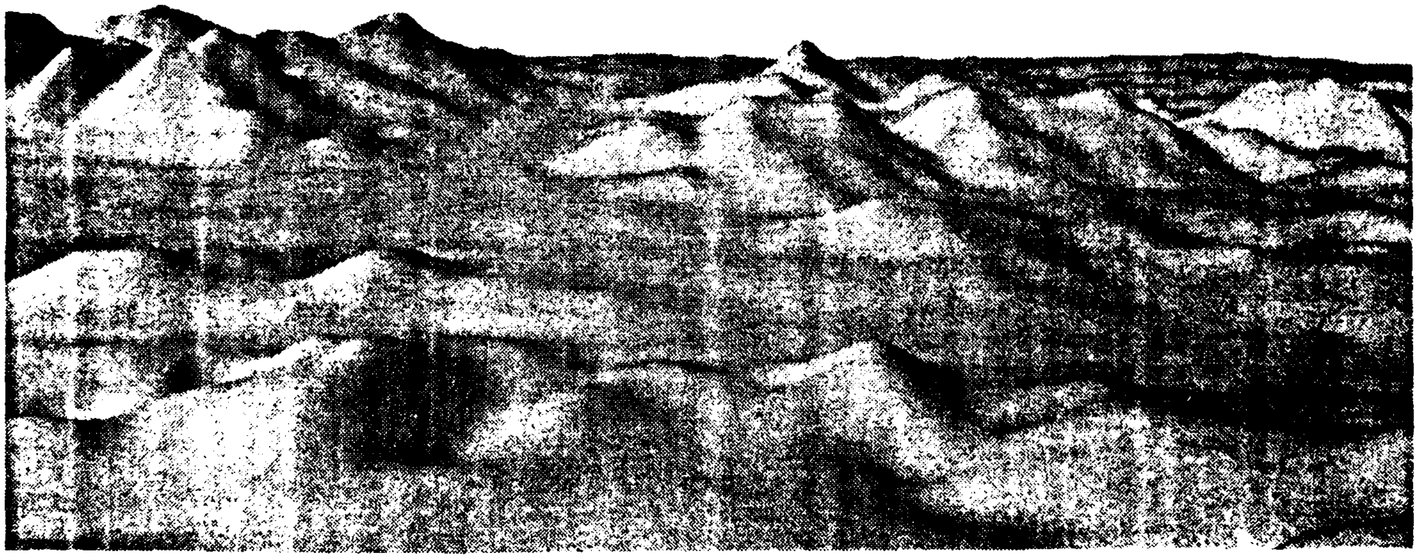
DIS = 6.

FIGURE C6. Fisheye View: Cache, OK.



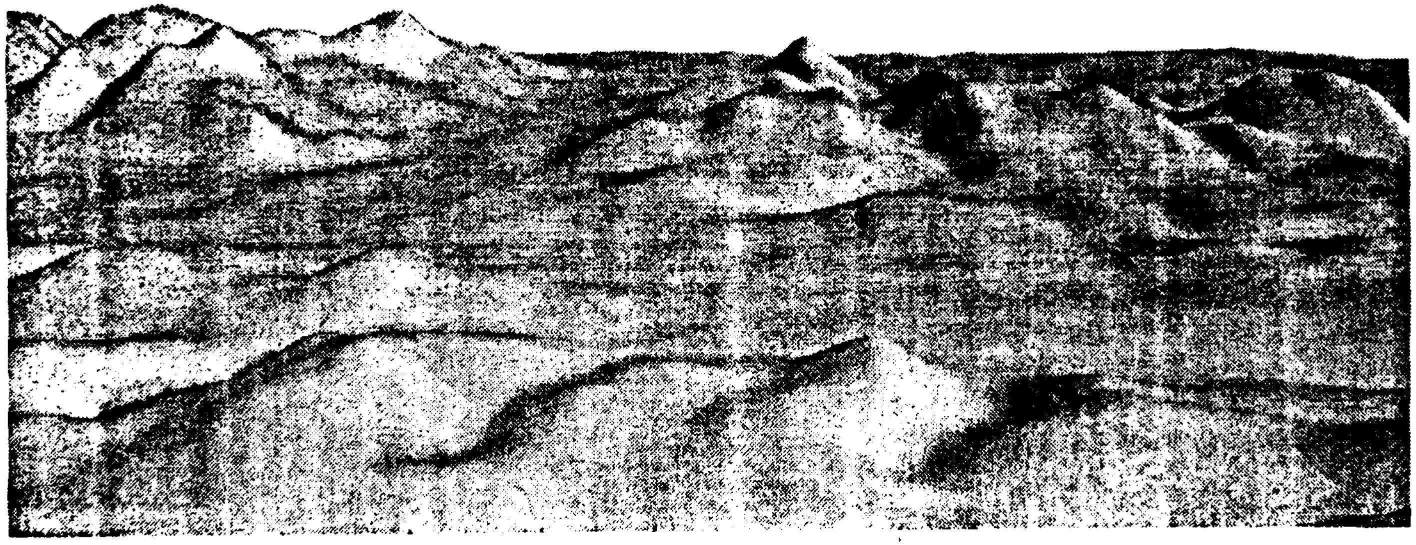
EL. AN = (60., 45.)

FIGURE C7. Perspective View. High Sun: Cache, OK.



EL, AW = (30, 45)

FIGURE C8. Perspective View - Low Sun - Cache, OK.

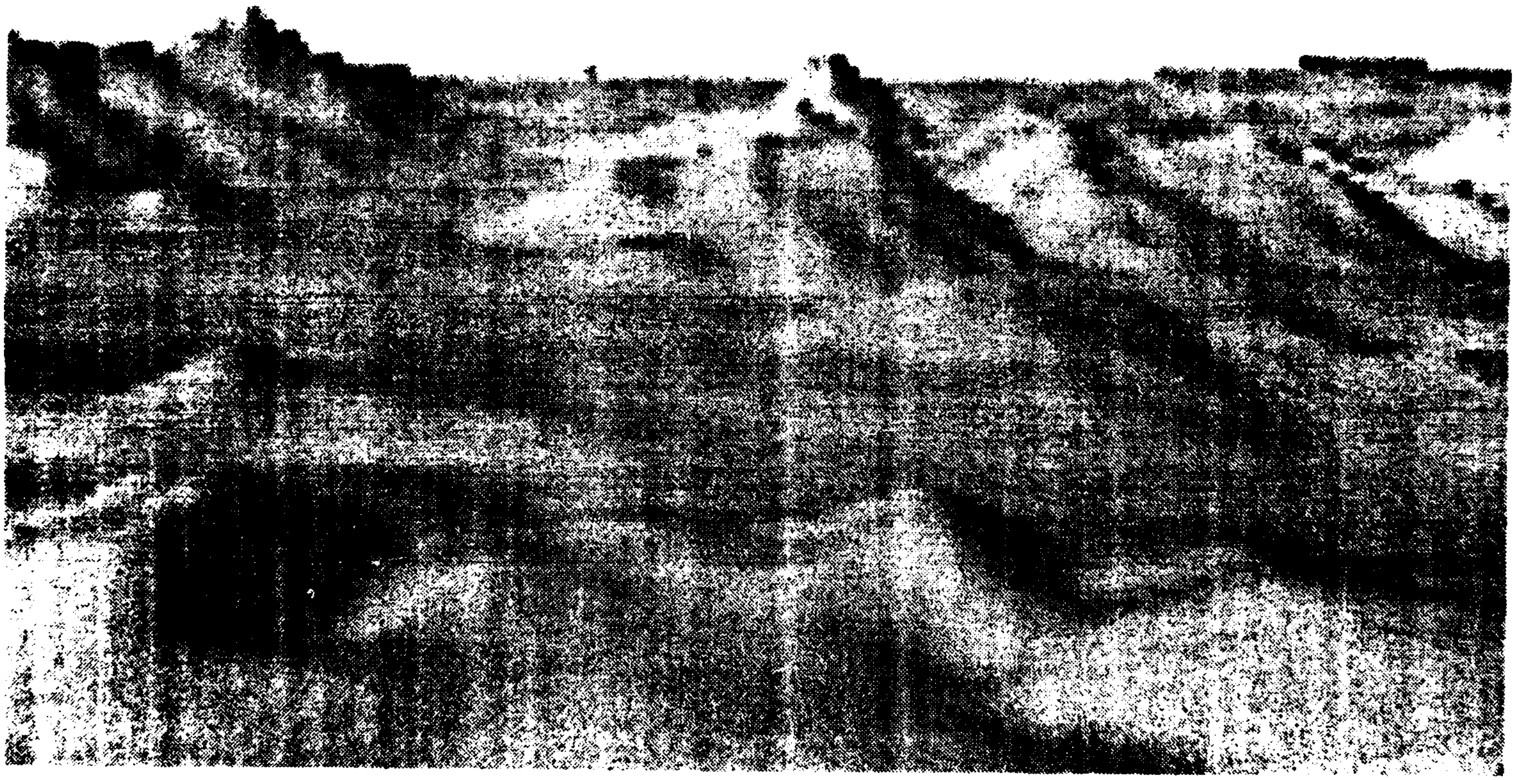


EL, AN = (45., 315.)

FIGURE C9. Perspective View, NW Sun Cache, OK

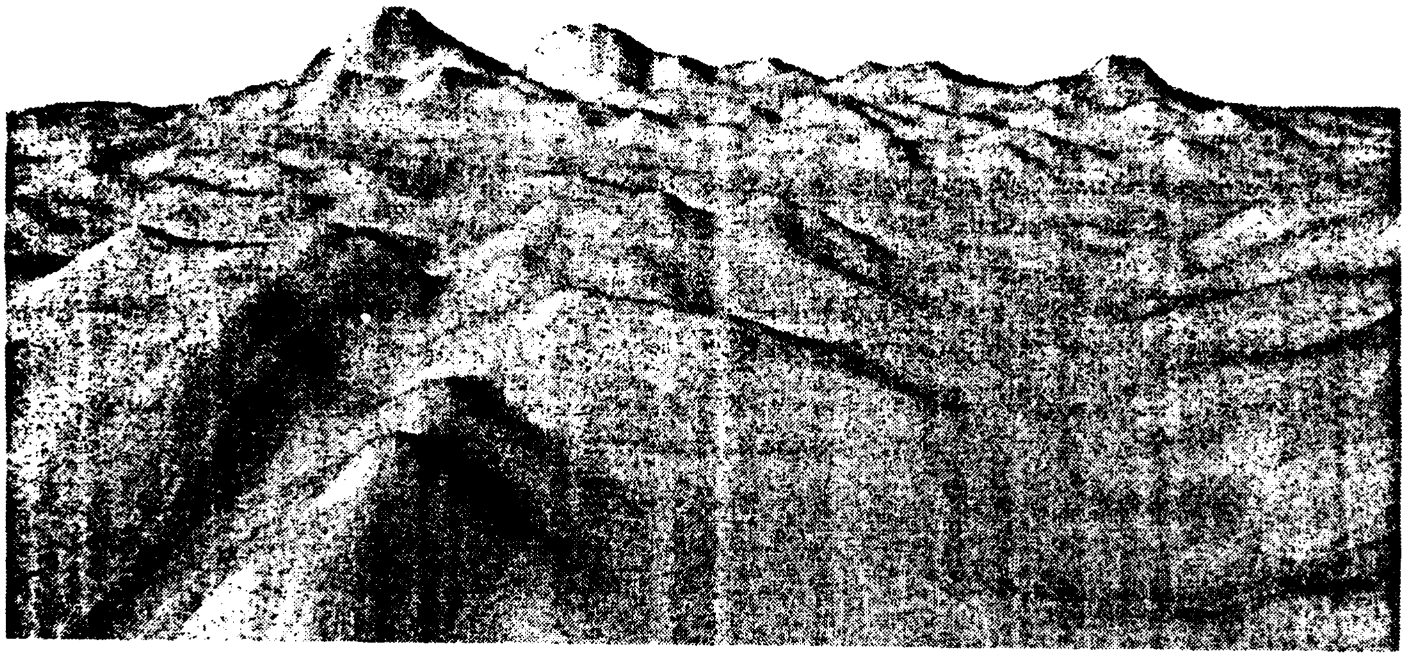
AINTN = 10.

FIGURE C10. Perspective View With Haze: Cache, OK.



NPIX = 128
NRAD = 179

FIGURE C11. Perspective View With 1:64 Resolution: Cache, OK.
(Magnified With Bilinear Interpolation)



X, Y = 8., 15. DIS = 12.
EL = 5000. OTHER PARAMETERS AS ABOVE
SUN = 45., 315.

FIGURE C12. Perspective View (N): Cache, OK.

X, Y = 5., 17.   AZ = 90.
H = 2000.   EL, AW = 50., 45.

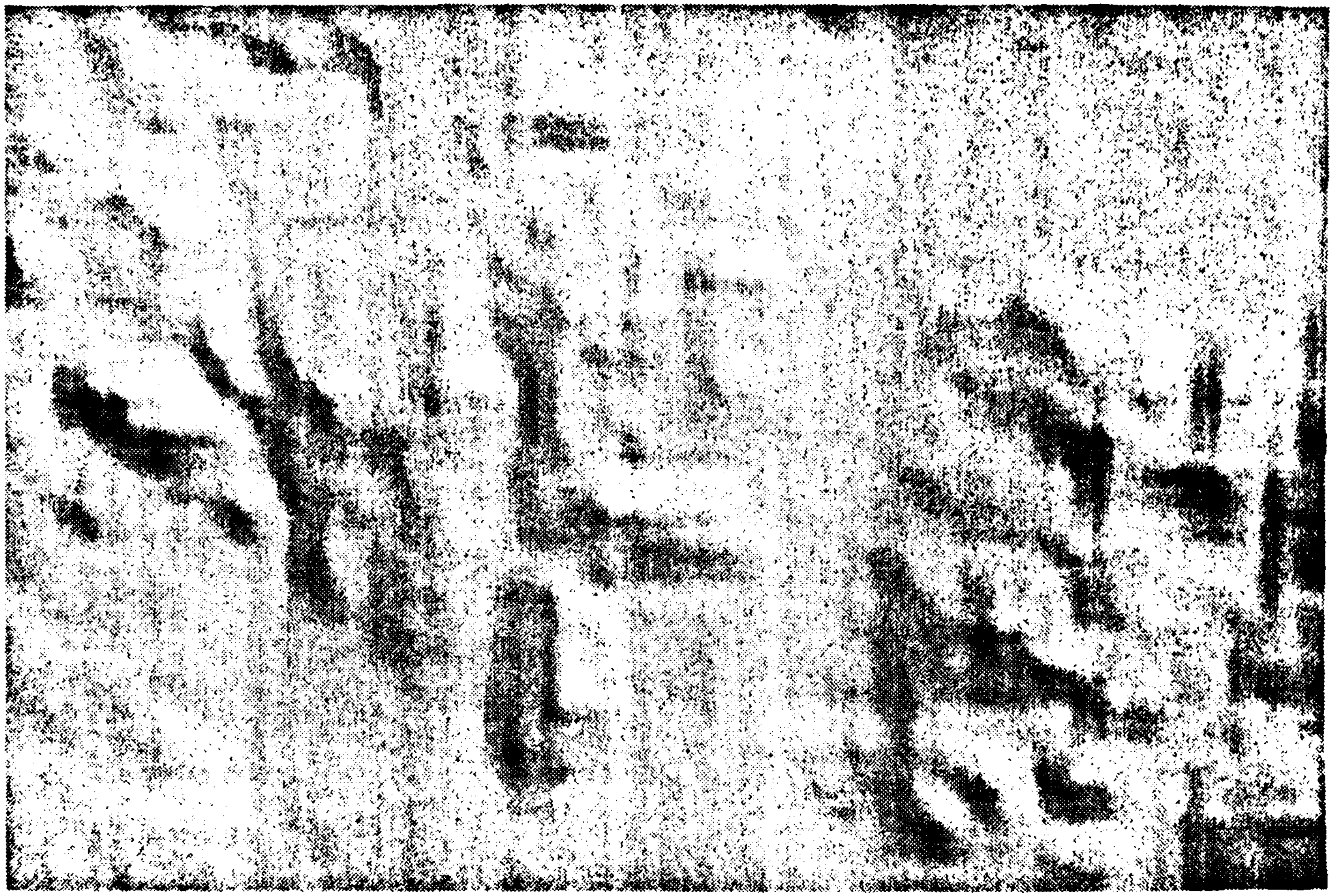
FIGURE C13. Perspective View: Elk Mountain.

X, Y =	5., 17.	EL, AW =	90., 45.
H =	2000.	ATNTN =	3.
AZ =	90.		

FIGURE C14. Perspective View: Elk Mountain With Haze.

10° SUN ANGLE VARIATION

FIGURE C15. Perspective View With Ridge Enhancement: Cache, OK.



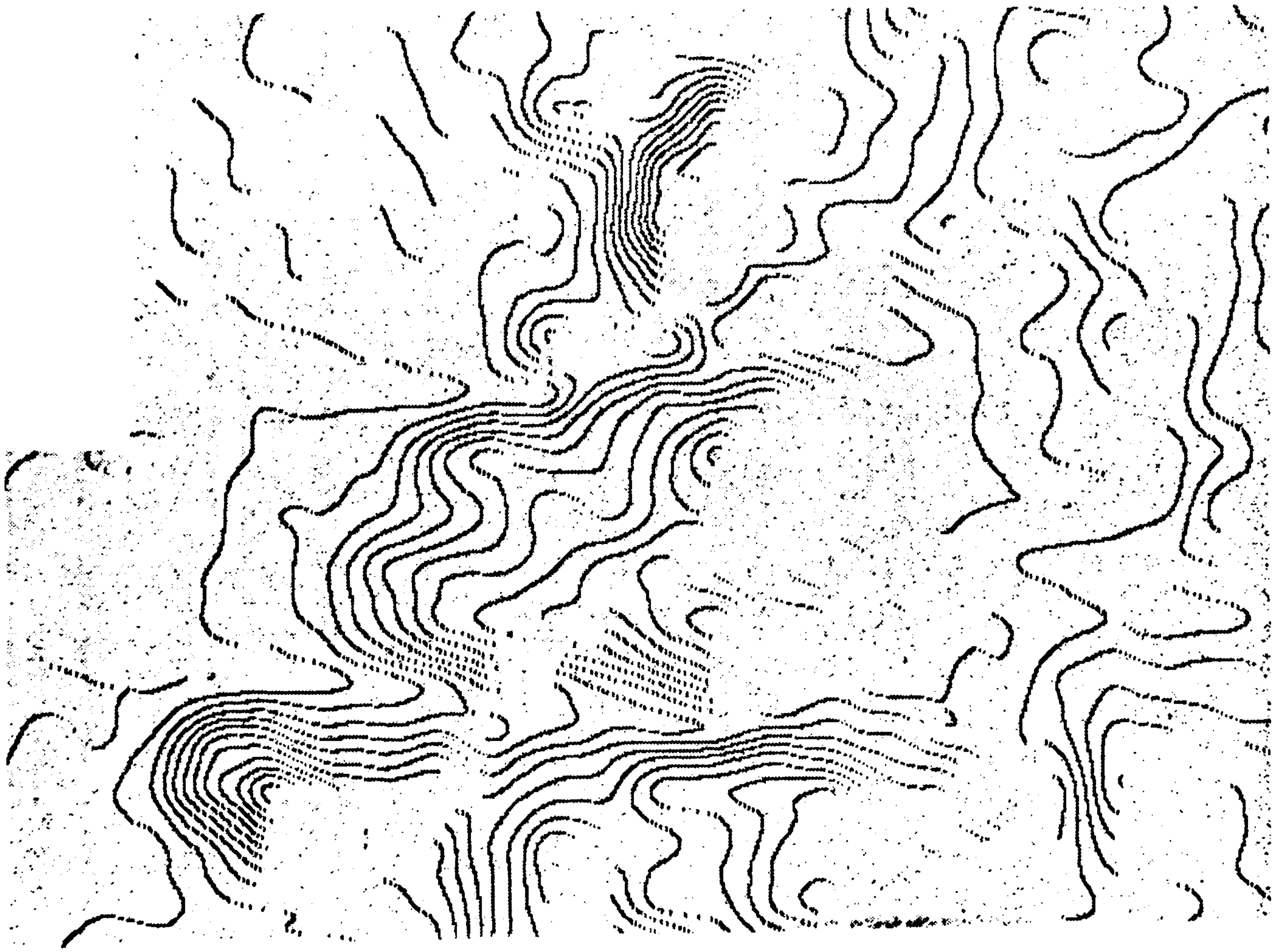
Sun from upper left, no variation of sun angle: 45° elevation Lambert's law; 900 x 600 image; 3x vertival exaggeration.

FIGURE C16. Orthonormal Shaded Relief: Cache, OK.

Parameters same as for (C16), both with 10° local sun variation.

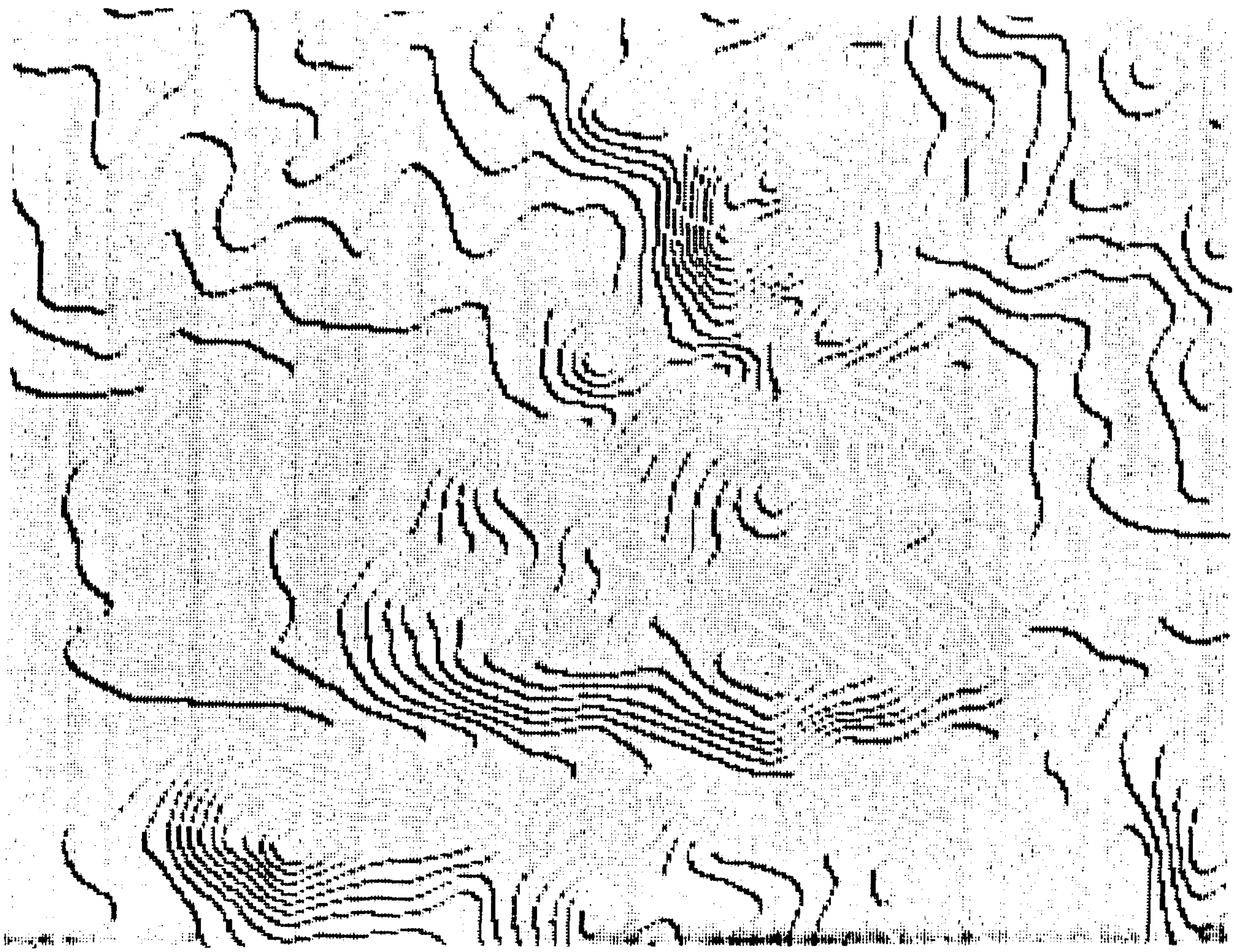
FIGURE C17. Orthonormal Shaded Relief: Cache, OK.

FIGURE C18. Orthonormal Shaded Relief Image With Variable Sun Azimuth Merged With SIMCON Contours. (20–meters Interval)



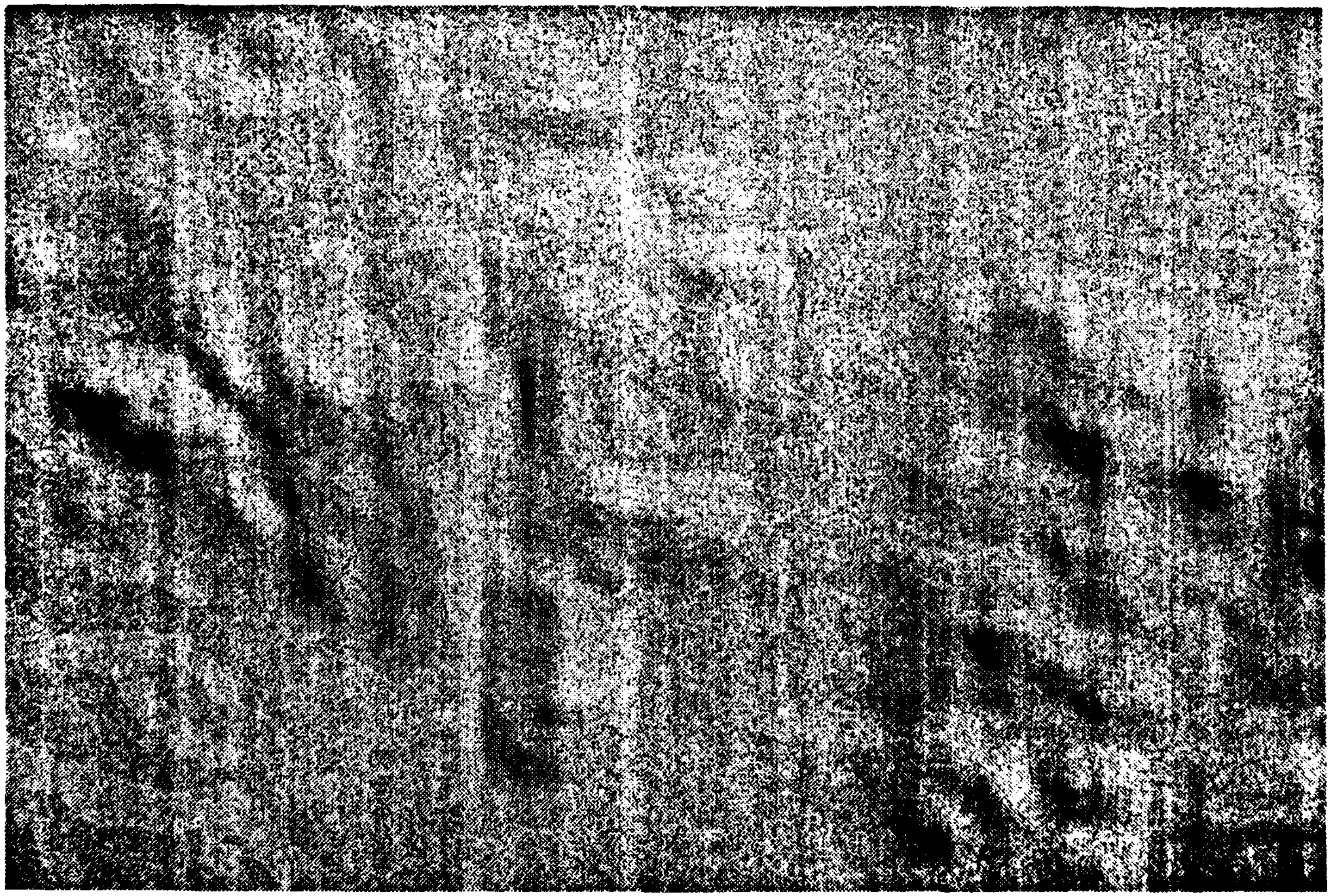
NE Sun, 10° Sun Angle Variation 10–meter Contour Interval.

FIGURE C19. Relief Contours: Cache, OK.



NW Sun, no solar azimuth variation.
(This image is different scale, and uses different half-toning routine from previous figure).

FIGURE C20. Relief Contours: Cache, OK.



Same Parameters as C16, but with Lommel–Seeliger law.

FIGURE C21. Orthonormal View: Cache, OK.

APPENDIX D.

```
SHDPER/PR:0/FP/CP/NOTR=SHDPER,[300,300]DSKTRN
SXSY,LPPER,ALSLP
PTS,RGSTR,GRNDPT
/
UNITS=21
ASG=DB2:6,TI:5,TI:7
ASG=DB0:14,DB0:13
ASG=DB2:12
//
```

```
            C THIS PROGRAM WILL PRODUCE AN EBR-COMPATIBLE
            C  9-TRACK ODD-PARITY MAGNETIC TAPE FILE
            C  CONTAINING THE CODED DENSITIES OF A PER-
            C  SPECTIVE VIEW OF THE CACHE MAP SHEET,
            C  BASED ON USER SPECIFIED INPUT PARAMETERS.
            C
            C       PROGRAM BY CYRUS C. TAYLOR
            C       AUTOMATED CARTOGRAPHY BRANCH
            C       USAETL, FORT BELVOIR, VA
            C       18 JULY 1978
            C
            C
0001                LOGICAL*1 SHADE(2048)
0002                DIMENSION Y(2048)
0003                DIMENSION ATIM(2)
0004                COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
0005                COMMON /REG/NRD1,NRD2,NRD3,NRD4,NRG1,NRG2,DXX
0006                COMMON /NEW/MRDR
0007                COMMON /SEXY/DELTAG,COSEL,IVSUN,ANG,C1,C2,S1,S2
0008                COMMON /AARGH/X,Y0,THETAP,D,NPTS,DX,H,VSF,
                1   SCL,TOL,DELR,DIS,THETA,Y1ST
            C FOLLOWING COMMON ADDED 16 AUG 79
            C FOR NEW GRNDPT ROUTINE. C. TAYLOR
0009                COMMON /HGRAA/Z1,Z2,CSTP,SSTP,CST,SST
            C
0010                DATA XMAXB/17.696/,XMINB/.17717/,YMAXB/21.59/,
                1   YMINB/.17717/,ELEVB/390./,IUNIT/0/,KODE/-1/,MODE/0/,
                1   IPAR/0/,IDEN/0/,ICTL/128/,KOUNT/128/
0011                DATA PI/3.1415926536/,SCL/.0007874016/
                1   MRDR/1/
0012                DATA IIMG/0/,IPFST/0/,IRFST/0/
            C       OPEN(UNIT=4,NAME='PL3DEM.DAT',TYPE='OLD',ACCESS=
            C   1   'DIRECT',READONLY,SHARED,ASSOCIATEVARIABLE=IV)
            C       OPEN(UNIT=9,NAME='FOR005.DAT',TYPE='OLD',
            C   1   ACCESS='DIRECT',READONLY,SHARED,ASSOCIATEVARIABLE=IW)
            C
            C       CALL TAPTRN(IUNIT,KODE,KOUNT,IER,NTRAN,IPOS,ISTAT,
            C   1   SHADE,MODE,IDONE)
            C       IF(IERR.EQ.0)GO TO 100
            C       WRITE(5,1)IERR
            C 1     FORMAT(' TAPTRN ERROR: ',I3)
            C       STOP
0013          100   CALL DSKFIL(4,-1,'DB0','CACHE1.DAT',DUM,1024,100
                1   ,IFLAG,IERR)
0014                IF(IERR.EQ.0)GO TO 102
0015                WRITE(7,2)IERR,IFLAG
0016          2     FORMAT(' OPEN FAILURE DB0:CACHE1 ',2I3)
0017                STOP
            C 101   CALL DSKFIL(3,-1,'DB0','CACHE2.DAT',DUM,1024,122,
            C   1   IFLAG,IERR)
            C       IF(IERR.EQ.0)GO TO 102
            C       WRITE(5,3)IERR,IFLAG
            C 3     FORMAT(' OPEN FAILURE DB0:CACHE2 ',2I3)
            C       STOP
0018          102   KOUNT=8*IPAR+IDEN+ICTL
0019                CALL DSKFIL(2,-1,'DB2','LPPER.DAT',DUM,1024,2048,
```

```
                1   IFLAG,IERR)
0020                IF(IERR.EQ.0)GO TO 114
0021                WRITE(7,115)IERR,IFLAG
0022        115     FORMAT(1X,'DB2:LPPER OPEN FAILURE',2I3)
0023                STOP
0024        114     CONTINUE
0025        127     FORMAT(1X,2A4)
          C         CALL TAPTRN(IUNIT,6,KOUNT,IERR,NTRAN,IPOS,ISTAT,
          C     1   SHADE,MODE,IDONE)
          C         IF(IERR.EQ.0)GO TO 103
          C         WRITE(5,1)IERR
          C         STOP
          C
0026        103     WRITE(7,4)
0027          4     FORMAT(' ENTER COORDINATES OF POSITION')
0028                READ(5,*)X,Y0
0029                WRITE(7,5)
0030          5     FORMAT(' ENTER ALTITUDE,AZIMUTH,#RADIALS,PTS/RADIAL',
                1   /,'  DIS,#PIX/COLUMN,D')
0031                READ(5,*)HT,AZ1,NRADS,NPTS,DIS,NPIX,D
0032                WRITE(7,6)
0033          6     FORMAT(' ENTER  VERTICAL',
                1   ' SCALING FACTOR, TOLERANCE')
0034                READ(5,*),VSF,TOL
0035                WRITE(7,7)
0036          7     FORMAT(' ENTER 0 FOR LAMBERTS LAW, 1 FOR LOMMEL-',/,
                1   '  SEELIGER LAW; DENSITY SCALING FACTOR;ATNTN COEF')
0037                READ(5,*)LAW,DSCL,ATNTN1
0038                WRITE(7,8)
0039          8     FORMAT(' ENTER ELEVATION,AZIMUTH OF SUN;',/,'
                1   VAR AZIMUTHIATION OF SOLAR')
0040                READ(5,*)EL1,AN1,RVIEW1
          C FOLLOWING 3 LINES OF CODE ADDED 1 AUG 79 BY
          C C. TAYLOR TO IMPROVE EFFICIENCY IN
          C 'SNAPSHOT' PROCESSEING, ETC.
0041                WRITE(7,125)
0042        125     FORMAT(1X,'ENTER Y-OFFSET')
0043                READ(5,*)YFSET
0044                CALL TIME(ATIM)
0045                WRITE(7,127)(ATIM(IJKL),IJKL=1,2)
0046                IVSUN=0
0047                IF(RVIEW1.LT.1.)GO TO 104
0048                IVSUN=1
0049         104    AN=90.-AN1
0050                IF(NRADS.GE.1024)GO TO 126
0051                IPFST=1024*(IIMG-(2*(IIMG/2)))
0052                IRFST=1024*(IIMG/2)
0053        126     CONTINUE
0054                AZA=(PI/2.)-(AZ1*PI)/180.
0055                RVIEW=RVIEW1/360.
0056                RVIEW=(PI-2.*RVIEW)/2.
0057                EL=(EL1*PI)/180.
0058                AN=(AN*PI)/180.
0059                ANG=((135.*PI)/180.)-AN
0060                DX=14./FLOAT(NRADS-1)
          C FOLLOWING LINE ADDED 1 AUG 79 BY C. TAYLOR
```

```
0061               OFST=YFSET/DX
0062               DXX=DX
0063               MRAD=NRADS/2
0064               H=HT*12./50000.
0065               SZ=SIN(EL)
0066               SY=SIN(AN)*COS(EL)
0067               SX=COS(AN)*COS(EL)
0068               COSEL=COS(EL)
0069               DELR=D/FLOAT(NPTS-1)
0070               DELTA=D/FLOAT(NPIX)
0071               DELTAG=(DELTA*50000.)*(2.54/100.)
0072               DELTAP=(DELTA/(((5.E-4)*100./2.54)*9.))
0073               C1=COS(-RVIEW+(3.*PI/4.))
0074               S1=SIN(RVIEW+(3.*PI/4.))
0075               S2=SIN(RVIEW-(PI/4.))
0076               C2=COS(-RVIEW-(PI/4.))
0077               DELSQR=DELTAG**2
0078               ATNTN=ATNTN1
0079               ITST=0
0080               CALL IRGSTR
0081               WRITE(6,10)X,Y0,HT,AZ1,VSF,LAW,ATNTN1
0082         10    FORMAT(' X,Y=(',2F9.4,') HT=',F9.4,' AZ=',F9.4,' VSF=',
                 1 F9.4,' LAW=',I5,'ATNTN1=',F8.5)
0083               WRITE(6,11)NRADS,NPTS,NPIX,MRDR,TOL,RVIEW1
0084         11    FORMAT(' NRADS=',I5,' NPTS=',I5,' NPIX=', I5,' MRDR=',I5,'
                 1 TOL=',F9.4,'RVIEW=',F8.3)
0085               WRITE(6,12)EL1,AN1,DSCL,IVSUN,D,DIS
0086         12    FORMAT(' EL,AN=(',2F9.4,') DSCL=',F9.4,' IVSUN=', I5
                 1  ,' D=',F9.4,' DIS=',F9.4,/)
             C DO LOOP OVER THE RADIALS
             C
0087               DO 105 NRAD=MRAD,-MRAD,-1
             C
             C FIRST PT OF RADIAL IS X,Y; LAST POINT FOUND BY
             C  TRIGONOMETRIC CONSIDERATIONS. NOTE THAT IN THIS PROGRAM
             C  RADIALS ARE SEPERATED BY A CONSTANT DISPLACEMENT
             C  (DX) AT THE IMAGE PLANE, RATHER THAN BY A CONSTANT ANGLE
             C  AS IN THE TEKTRONIX PERSPECTIVE ROUTINES.
             C
             C MULTIPLE IMAGE FILES MUST READ PREVIOUS IMAGE
             C IF THE PREVIOUS IMAGE IS TO BE PRESERVED.
             C FOLLOWING 8 LINES ADDED C. TAYLOR 10 AUG 79
0088               MRCNT=MRAD-NRAD+1+IRFST
0089               IF(IIMG.EQ.0)GO TO 128
0090               CALL DSKTRN(2,1,IERR,MRCNT,NRES,SHADE,0,IDONE)
0091               IF(IERR.EQ.0)GO TO 128
0092               WRITE(7,118)IERR,INRAD,MRAD,MRCNT
0093               STOP
0094        128    CONTINUE
             C END OF ADDITION 10 AUG 79 CCT
0095               THETA=ATAN((NRAD*DX)/DIS)
0096               IASCD=0
0097               THETAP=AZA+THETA
0098               CST=COS(THETA)
0099               SST=SIN(THETA)
0100               CSTP=COS(THETAP)
```

```
0101            SSTP=SIN(THETAP)
0102            XEND=D*(CSTP)
0103            XEND=X+XEND
0104            YEND=Y0+D*(SSTP)
0105            CALL PTS(X,Y0,XEND,YEND,NPTS,Y,DLN,IASCD)
        C FOLLOWING LINE ADDED FOR COMPATIBILITY WITH
        C NEW GRNDPT ROUTINE. C. TAYLOR. 16 AUG 1979.
0106            IASCD=1
        C FOLLOWING COMMENTS ARE NO LONGER APPROPRIATE.
        C THESE COMPUTATIONS ARE NOW DONE IN GROUND
        C CORRELATION ROUTINE.  C. TAYLOR 16 AUG 79.
        C       WRITE(6,13)(Y(IGF),IGF=1,NPTS)
        C
        C  NOW HAVE THE RADIAL REPRESENTED BY A PROFILE OF
        C  ELEVATION DATA PTS.  WE NEXT TRANSFORM THESE TO
        C  SCREEN COORDINATES, SO THAT WE MAY COMPARE PIXEL
        C  Y-VALUES WITH PROJECTED GROUND PTS, EVENTUALLY
        C  INTERPOLATING TO FIND GROUND POSITIONS CORRESPONDING
        C  TO PIXEL POSITIONS.
        C
        C FOLLOWING 3 LINES COMMENTED OOUT FOR NEW
        C GRNPDT ROUTINE. 16 AUG 1979. C. TAYLOR
        D       DO 106 IPT=NPTS,2,-1
        D       YS=H+(Y(1)-Y(IPT))*VSF*SCL
        D 106   Y(IPT)=7.-YS*DIS/((IPT-1)*DELR*CST)
0107            Y1ST=Y(1)
        C FOLLOWING LINE COMMENTED OUT 16 AUG 79. CCT.
        D       Y(1)=-100.
        C
        C THE CURRENT RADIAL HAS NOW BEEN CONVERTED TO SCREEN
        C  Y-VALUES.  NOW LOOP OVER THE PIXEL POSITIONS OF
        C  THE SCREEN, DETERMINING CORRESPONDING GROUND
        C  POSITIONS, IF ANY, AND COMPUTE IMAGE DENSITY AT
        C  THOSE POINTS.
        C
0108            NFLAG=0
0109            YMAX=0.
0110            I=1
        C       WRITE(6,13)(Y(IGF),IGF=1,NPTS)
0111     13     FORMAT(10(1X,10F10.4,/),//)
0112            DO 107 J=1,NPIX
        C FOLLOWING LINE ADDED 1 AUG 79.C.TAYLOR.
0113            IF(J.LT.OFST)GO TO 108
        C FOLLOWING LINE COMMENTED OUT FOR NEW GRNPT
        C ROUTINE 17 AUG 79. C. TAYLOR.
        D       IASCD=0
0114            CALL GRNDPT(J,Y,NFLAG,X1,Y1,I,YMAX,IASCD
             1  ,AZ,BZ,Z,ITST)
        C       IF(J.NE.10.AND.J.NE.11)GO TO 112
        C       WRITE(5,15)X,Y0,X1,Y1
        C15     FORMAT(' GRNDPT VALUES:',4F10.4)
        C112    CONTINUE
0115            IF(NFLAG.NE.0.AND.NFLAG.NE.10)GO TO 108
        D       IF(NFLAG.EQ.10)GO TO 111
        C FOLLOWING LINE COMMETED OUT 16
        C AUG 79 FOR NEW GRNDPT ROUTINE. C. TAYLOR
```

```
D          IASCD=1
C FOLLOWING LINE SUPERFLUOUS WITH NEW GRNPDT
C ROUTINE. 16 AUG 79. C. TAYLOR
           CALL ALT(X1,Y1,Z,SN,AZ,BZ,IASCD)
C FOLLOWING TWO LINES COMMENTED OUT C. TAYLOR 10 AUG 79
C          AZ1=AZ
C          BZ1=BZ
           AZ=AZ*VSF*DELTAP
           BZ=BZ*VSF*DELTAP
           CALL SXSY(SX,SY,AZ,BZ)
           BEG=-SX*AZ*DELTAG-SY*BZ*DELTAG+DELSQR*SZ
           DISI=SQRT((AZ*DELTAG)**2+(BZ*DELTAG)**2+(DELSQR**2))
           COSI=BEG/DISI
           IF(COSI.LT.1.E-6)COSI=.000001
           IF(LAW.EQ.1)GO TO 109
           SHAD=COSI
C          WRITE(5,113)AZ1,BZ1,AZ,BZ,VSF,DELTAP,SX,SY,SZ,DELTAP,DELTAG
  113      FORMAT(2(1X,6F10.5,/),//)
C THIS LINE ADDED IN NEW FOG FCTR CALCS
C 24 AUG 79 C. TAYLOR
           IF(ABS(ATNTN).LT.1.E-3)GO TO 138
   109     CONTINUE
C
C LOMMEL SEELIGER NOT AS EASY AS IN NORMAL VIEW...
C
           RIP=SQRT((X1-X)**2+(Y1-Y0)**2)
           YS=H+(Y(1)-Z)*VSF*SCL
           RRR=SQRT(RIP**2+YS**2)
C FOLLOWING LINE ADDED FOR SMOG FACTOR
C C. TAYLOR 24 AUG 79
           IF(LAW.NE.1) GO TO 110
           ALPHA=(ATAN(YS/RIP))
           VX=(CSTP)*COS(ALPHA)
           VY=(SIN(THETAP))*(COS(ALPHA))
           VZ=SIN(ALPHA)
           COSE=((-DELTAG*(AZ*VX+BZ*VY)+VZ*DELSQR))/DISI
           SHAD=1./(1.+(COSE/COSI))
           IF(SHAD.LT.1.E-6)SHAD=0.000001
           IF(ABS(ATNTN).LT.1.E-3)GO TO 138
  110      CONTINUE
           FCTR=ABS(RRR/ATNTN)
           SHAD=1.+(SHAD-1.)*EXP(-FCTR)
  138      SHAD=(ALOG10(1./SHAD))*DSCL
           IF(SHAD.LT.1.)SHAD=1.
           IF(SHAD.GT.127.)SHAD=127.
           SHADE(J+IPFST)=SHAD
           GO TO 107
   108     SHADE(J+IPFST)=0.
   107     CONTINUE
C
C NOW OUTPUT PROFILE
C
C
C
C FOLLOWING 2 LINES MOD'FD 10 AUG 79 C. TAYLOR
           NROD=MRCNT-IRFST
```

```
0152              CALL RGSTR(NROD,SHADE)
0153              CALL DSKTRN(2,2,IERR,MRCNT,NRES,SHADE,0,IDONE)
0154              IF(IERR.EQ.0)GO TO 105
0155              WRITE(7,118)IERR,NRAD,MRAD,MRCNT
0156       118    FORMAT(1X,'DSKTRN FAILURE',I3,3I6)
         C        WRITE(5,14)NRAD,MRAD,MRCNT
         C 14     FORMAT(' NRAD=',3I6)
0157              STOP
         C
         C
         C
0158        105   CONTINUE
0159              IF(NPIX.GT.100)GO TO 136
0160              DO 116 NRAD=1,(2*MRAD+1)
         C FOLLOWING LINE ADDED 10 AUG 79 FOR MULTIPLE
         C IMAGE PROCESSING. C. TAYLOR
0161              NROD=NRAD+IRFST
0162              CALL DSKTRN(2,1,IERR,NROD,NRES,SHADE,0,IDONE)
0163              IF(IERR.EQ.0)GO TO 120
0164              WRITE(7,118)IERR
0165              STOP
         C NOTE THAT PROGRAM IS NOT!!!!!!!!!! MODIFIED TO
         C CORRECTLY PRINT MULTIPLE IMAGE FILES ON LP0:
0166       120    CALL LP(SHADE)
0167       116    CONTINUE
0168       136    CONTINUE
         C
         C BECAUSE OF EBR SOFTWARE PROBLEMS, SEPERATE
         C  IMAGES BY 5 RECORDS OF BALNKS, WITH AN
         C  EOF AFTER ALL IMAGES
         C
         C
         C
         C
         C
         C
         C
         C
         C
0169     111      CONTINUE
         C...     4 LINES COMMENTED OUT R. ROSENTHAL 6/79
         C UNCOMMENTED C. TAYLOR, WITH MODS TO ALLOW
         C  MULTIPLE IMAGES TO BE GENERATED OVERNIGHT
         C  NEXT 3  LINES ADDED SAME TIME C. TAYLOR  9 AUG 79
0170              IIMG=IIMG+1
0171              CALL TIME(ATIM)
0172              WRITE(7,127)(ATIM(IJKL),IJKL=1,2)
0173              WRITE(7,139)ITST
0174       139    FORMAT(1X,'#ALT CALLS FROM GRNDPT=',I8)
0175              IF(IIMG.GT.3)GO TO 137
0176              WRITE(7,9)
0177       9      FORMAT(' ENTER 1 FOR ANOTHER IMAGE')
0178              READ(5,*)IANS
0179              IF(IANS.EQ.1)GO TO 103
0180       137    CONTINUE
0181              CALL DSKFIL(2,-2,'DB2','LPPER.DAT',DUM,1024,2048,
```

```
                1  IFLAG,IERR)
0182               CALL DSKFIL(4,-2,'DB0','CACHE1.DAT',DUM,1024,100,
                1  IFLAG,IERR)
          C        CALL DSKFIL(3,-2,'DB0','CACHE2.DAT',DUM,1024,122,
          C     1  IFLAG,IERR)
          C
          C        CLOSE(UNIT=4,DISPOSE='SAVE')
          C        CLOSE(UNIT=9,DISPOSE='SAVE')
          C
          C
          C
          C
          C
          C
          C
0183               STOP
0184               END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	005036	1295	RW,I,CON,LCL
2	$PDATA	000126	43	RW,D,CON,LCL
3	$IDATA	001574	446	RW,D,CON,LCL
4	$VARS	024366	5243	RW,D,CON,LCL
5	$TEMPS	000012	5	RW,D,CON,LCL
6	BOUNDS	000024	10	RW,D,OVR,GBL
7	REG	000020	8	RW,D,OVR,GBL
8	NEW	000002	1	RW,D,OVR,GBL
9	SEXY	000036	15	RW,D,OVR,GBL
10	AARGH	000066	27	RW,D,OVR,GBL
11	HGRAA	000030	12	RW,D,OVR,GBL

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALPHA	R*4	4-024332	AN	R*4	4-024120	ANG	R*4	9-000012	AN1	R*4	4-024102	ATNIN	R*4	4-024176
AININI	R*4	4-024072	AZ	R*4	4-024256	AZA	R*4	4-024124	AZ1	R*4	4-024054	BEG	R*4	4-024276
BZ	R*4	4-024262	COSE	R*4	4-024352	COSEL	R*4	9-000004	COSI	R*4	4-024306	CST	R*4	11-000020
CSTP	R*4	11-000010	C1	R*4	9-000016	C2	R*4	9-000022	D	R*4	10-000014	DELR	R*4	10-000046
DELSQR	R*4	4-024172	DELTA	R*4	4-024162	DELTAG	R*4	9-000000	DELTAP	R*4	4-024166	DIS	R*4	10-000052
DIST	R*4	4-024302	DEN	R*4	4-024230	DSCL	R*4	4-024066	DUM	R*4	4-024040	DX	R*4	10-000022
DXX	R*4	7-000014	EL	R*4	4-024134	ELEVB	R*4	6-000020	EL1	R*4	4-024076	FCTR	R*4	4-024356
H	R*4	10-000026	HI	R*4	4-024050	I	I*2	4-024242	IANS	I*2	4-024364	IASCD	I*2	4-024216
ICTL	I*2	4-024022	IDEN	I*2	4-024020	IDONE	I*2	4-024212	IERR	I*2	4-024046	IFLAG	I*2	4-024044
IIMG	I*2	4-024032	IJKL	I*2	4-024116	INRAD	I*2	4-024214	IPAR	I*2	4-024016	IPFST	I*2	4-024034
IRFST	I*2	4-024036	ITST	I*2	4-024202	IUNIT	I*2	4-024010	IVSUN	I*2	9-000010	J	I*2	4-024244
KODE	I*2	4-024012	KOUNT	I*2	4-024024	LAW	I*2	4-024064	MODE	I*2	4-024014	MRAD	I*2	4-024144
NRCNT	I*2	4-024206	MRDR	I*2	8-000000	NFLAG	I*2	4-024234	NPIX	I*2	4-024062	NPIS	I*2	10-000020
NRAD	I*2	4-024204	NRADS	I*2	4-024060	NRD1	I*2	7-000000	NRD2	I*2	7-000002	NRD3	I*2	7-000004
NRD4	I*2	7-000006	NRES	I*2	4-024210	NRG1	I*2	7-000010	NRG2	I*2	7-000012	NROL	I*2	4-024362
OFST	R*4	4-024140	P1	R*4	4-024026	RIP	R*4	4-024316	RRR	R*4	4-024326	RVIEW	R*4	4-024130
RVIEW1	R*4	4-024106	SCL	R*4	10-000036	SHAD	R*4	4-024312	SN	R*4	4-024272	SST	R*4	11-000024
SSTP	R*4	11-000014	SX	R*4	4-024156	SY	R*4	4-024152	SZ	R*4	4-024146	S1	R*4	9-000026
S2	R*4	9-000032	THETA	R*4	10-000056	THETAP	R*4	10-000010	TOL	R*4	10-000042	VSP	R*4	10-000032
VX	R*4	4-024336	V1	R*4	4-024342	V2	R*4	4-024346	X	R*4	10-000000	XEND	R*4	4-024220
X1	R*4	4-024246	XMAXB	R*4	6-000000	XMINB	R*4	6-000010	YEND	R*4	4-024224	YFSET	R*4	4-024112
Y1	R*4	4-024252	YMAX	R*4	4-024236	YMAXB	R*4	6-000004	YMINB	R*4	6-000014	YS	R*4	4-024322
Y0	R*4	10-000004	YTST	R*4	10-000062	Z	R*4	4-024266	Z1	R*4	11-000000	Z2	R*4	11-000004

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
ATIM	R*4	4-024000	000010	4	(2)
SHADE	L*1	4-000000	004000	1024	(2048)
Y	R*4	4-004000	020000	4096	(2048)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
2'	3-000000	4'	3-000104	5'	3-000146	6'	3-000252	7'	3-000332
8'	3-000472	9'	3-001202	10'	3-000620	11'	3-000720	12'	3-001022
13'	**	100	**	102	1-000112	103	1-000236	104	1-001062
105	1-004342	107	1-004164	108	1-004142	109	1-003402	110	1-003756
111	**	113'	**	114	1-000230	115'	3-000040	116	**
118'	3-001114	120	1-004520	125'	3-000576	126	1-001162	127'	3-000076
128	1-002544	136	1-004552	137	1-005002	138	1-004034	139'	3-001144

FUNCTIONS AND SUBROUTINES REFERENCED

```
ACT     DSKFIL  DSKIR.  GRNDPT  IRGSTR  LP      PTS     RGSTR   SXSY    TIME    $ALOG10 $ATAN   $COS    $EXP    $SIN    $SQRT
```

TOTAL SPACE ALLOCATED = 033602 7105

,LP,LST=SHDPER

```
C THIS SUBROUTINE IS DESIGNED TO VARY THE AZIMUTH OF THE
C  'SUN' TO STRENGTHEN THE DETAIL IN SHADED RELIEF IMAGES
C  CREATED BY PROGRAMS SHADE.FTN AND SHADLP.FTN.
C REFERENCE: P. YOELI, 'SOME REMARKS ON THE COMPLETION
C  OF THE ANALYTICAL HILL SADING PROJECT'
C  (UNPUBLISHED?-AUTO CARTO FILES)
C
C         PROGRAM BY CYRUS C. TAYLOR
C         AUTOMATED CARTOGRAPHY BRANCH
C         USAETL, FORT BELVOIR, VA
C         11 JULY 1978
C
C
          SUBROUTINE SXSY(SX,SY,AZ,BZ)
          COMMON /SEXY/DELTAG,COSEL,IVSUN,ANG,C1,C2,S1,S2
          REAL*4 NX,NY
          IF(IVSUN.EQ.0)GO TO 110
C
          NX=-AZ*DELTAG
          NY=-BZ*DELTAG
          DISFI=SQRT((NX**2)+(NY**2))
          IF(DISFI.LT.1.E-6)GO TO 110
          CO=NX/DISFI
          SO=NY/DISFI
          C=CO*COS(ANG)-SO*SIN(ANG)
          S=SO*COS(ANG)+CO*SIN(ANG)
C IN EFFECT, OUR X-Y COORDINATE SYSTEM HAS BEEN
C  ROTATED THROUGH AN ANGLE OF ANG RADIANS.  NOTE
C  THAT THE ANGULAR COORDINATE SYSTEM USED IN THIS
C  ROUTINE AND IN SHADE OR SHADLP IS NOT THE
C  AERONAUTICAL ANGULAR COORDINATE SYSTEM, BUT THE
C  TRADITIONAL ALGEBRAIC COORDINATE SYSTEM.  SINCE
C  OUR X-Y COORDINATE SYSTEM HAS BEEN ROTATED,
C  SX AND SY MUST BE ROTATED BACK BEFORE RETURNING
C  TO THE MAIN ROUTINE.
C SECTOR I CASE
          IF(C.GT.(C1). OR .S.LT.(S1))GO TO 10
          SX=-C*COSEL
          SY=-S*COSEL
          GO TO 100
C SECTOR III CASE
   10     IF(C.LT.(C2). OR .S.GT.(S2))GO TO 20
          SX=C*COSEL
          SY=S*COSEL
          GO TO 100
C SECTOR II CASE: 2 SUBCASES
   20     IF((C.LT.0.AND.S.LT.0).OR.(C.LT.0.AND.S.LT.S1)
     1     .OR.(S.LT.0.AND.C.LT.C2))GO TO 30
          IF(C.GT.(0.7071))GO TO 25
          SX=0.
          SY=-COSEL
          GO TO 100
   25     SX=COSEL
          SY=0.
          GO TO 100
C SECTOR IV CASE: 2 SUBCASES
```

```
0029      30    IF(C.LT.(-0.7071))GO TO 35
0030            SX=0.
0031            SY=-COSEL
0032            GO TO 100
0033      35    SX=COSEL
0034            SY=0.
0035      100   CONTINUE
        C  WE MUST NOW ROTATE OUR LOCAL X-Y COORDINATE
        C  SYSTEM BACK, SO THAT SX AND SY WILL BE CONSISTANT
        C  WITH THE MAIN ROUTINES.
0036            Y=SY*COS(ANG)-SX*SIN(ANG)
0037            X=SX*COS(ANG)+SY*SIN(ANG)
0038            SX=-X
0039            SY=-Y
0040     110  CONTINUE
0041            RETURN
0042            END
```

```
FORTRAN IV-PLUS V02-51E          13:23:08    21-APR-81          PAGE 3
SXSY.FTN         /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME        SIZE               ATTRIBUTES

  1      $CODE1  001000   256           RW,I,CON,LCL
  2      $PDATA  000014     6           RW,D,CON,LCL
  4      $VARS   000044    18           RW,D,CON,LCL
  5      $TEMPS  000004     2           RW,D,CON,LCL
  6      SEXY    000036    15           RW,D,OVR,GBL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 SXSY         1-000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 ANG    R*4   6-000012   A2     R*4   F-000006*  B2     R*4   F-000010*  C      R*4   4-000024   COSEL  R*4   6-000004
 C0     R*4   4-000014   C1     R*4   6-000016   C2     R*4   6-000022   DELTAG R*4   6-000000   DISFI  R*4   4-000010
 IVSUM  I*2   6-000010   NA     R*4   4-000000   N1     R*4   4-000004   S      R*4   4-000030   SX     R*4   F-000002*
 SY     R*4   F-000004*  S0     R*4   4-000020   S1     R*4   6-000026   S2     R*4   6-000032   X      R*4   4-000040
 Y      R*4   4-000034

LABELS

 LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

 10      1-000320        20      1-000412        25      1-000542        30      1-000570        35      1-000634
 100     1-000660        110     1-000776

FUNCTIONS AND SUBROUTINES REFERENCED

 $COS    $SIN    $SQRT

TOTAL SPACE ALLOCATED = 001122   297

,LP.LST=SXSY
```

```
        C  FILE NAME - LPPER.FTN
        C
0001              SUBROUTINE LP(SHADE)
0002              LOGICAL*1 SHADE(1),CHAR(12)
0003              LOGICAL*1 PRNT(100)
0004              DATA CHAR/' ', '.', '-', '1', '/', 'V', '*', '@', 'U',
             1           'A','4','W'/
        C         WRITE(5,51)(SHADE(M),M=1,32)
0005       51     FORMAT(1X,32O3,/)
0006              DO 10 I=1,3
0007              DO 50 K=1,100
0008              IF(SHADE(K).LT.0.)SHADE(K)=127.
0009              PRNT(K)=CHAR(1)
0010        50    CONTINUE
0011              DO 20 J=1,100
0012              IF(SHADE(J).GT.6)GO TO 21
0013              IF(I.NE.1)GO TO 20
0014              PRNT(J)=CHAR(1)
0015              GO TO 20
0016       21     IF(SHADE(J).GT.9)GO TO 22
0017              IF(I.NE.1)GO TO 20
0018              PRNT(J)=CHAR(2)
0019              GO TO 20
0020        22    IF(SHADE(J).GT.13)GO TO 43
0021              IF(I.NE.1)GO TO 20
0022              PRNT(J)=CHAR(3)
0023              GO TO 20
0024      43      IF(SHADE(J).GT.15)GO TO 23
0025              IF(I.NE.1)GO TO 20
0026              PRNT(J)=CHAR(4)
0027              GO TO 20
0028       23     IF(SHADE(J).GT.19)GO TO 24
0029              IF(I.NE.1)GO TO 20
0030              PRNT(J)=CHAR(5)
0031              GO TO 20
0032        24    IF(SHADE(J).GT.24)GO TO 25
0033              IF(I.NE.1)GO TO 20
0034              PRNT(J)=CHAR(6)
0035              GO TO 20
0036        25    IF(SHADE(J).GT.28)GO TO 26
0037              IF(I.NE.1)GO TO 20
0038              PRNT(J)=CHAR(7)
0039              GO TO 20
0040       26     IF(SHADE(J).GT.33)GO TO 27
0041              IF(I.NE.1)GO TO 20
0042              PRNT(J)=CHAR(8)
0043              GO TO 20
0044        27    IF(SHADE(J).GT.38)GO TO 28
0045              IF(I.EQ.3)GO TO 20
0046              IF(I.EQ.1)PRNT(J)=CHAR(9)
0047              IF(I.EQ.2)PRNT(J)=CHAR(2)
0048              GO TO 20
0049       28    IF(SHADE(J).GT.44)GO TO 29
0050              IF(I.EQ.3)GO TO 20
0051              IF(I.EQ.1)PRNT(J)=CHAR(10)
0052              IF(I.EQ.2)PRNT(J)=CHAR(5)
```

```
          GO TO 20
    29    IF(SHADE(J).GT.51)GO TO 30
          IF(I.EQ.3)GO TO 20
          IF(I.EQ.1)PRNT(J)=CHAR(10)
          IF(I.EQ.2)PRNT(J)=CHAR(5)
          GO TO 20
     30   IF(SHADE(J).GT.58)GO TO 31
          IF(I.EQ.3)GO TO 20
          IF(I.EQ.1)PRNT(J)=CHAR(10)
          IF(I.EQ.2)PRNT(J)=CHAR(8)
          GO TO 20
     31   IF(SHADE(J).GT.66)GO TO 32
          IF(I.EQ.3)PRNT(J)=CHAR(4)
          IF(I.EQ.1)PRNT(J)=CHAR(6)
          IF(I.EQ.2)PRNT(J)=CHAR(10)
          GO TO 20
     32   IF(SHADE(J).GT.76)GO TO 33
          PRNT(J)=CHAR(11)
          IF(I.EQ.1)PRNT(J)=CHAR(10)
          IF(I.EQ.2)PRNT(J)=CHAR(9)
          GO TO 20
     33   PRNT(J)=CHAR(12)
          IF(I.EQ.1)PRNT(J)=CHAR(10)
          IF(I.EQ.2)PRNT(J)=CHAR(5)
    20    CONTINUE
          WRITE(6,500)(PRNT(L),L=1,100)
    500   FORMAT('+',1X,100A1)
    10    CONTINUE
          WRITE(6,501)
    501   FORMAT(1X)
          RETURN
          END
```

```
ORTRAN IV-PLUS V02-51E          13:23:37     21-APR-81           PAGE 3
PPER.FTN        /TR:BLOCKS/WR

ROGRAM SECTIONS

UMBER   NAME        SIZE                ATTRIBUTES

 1      $CODE1  001704   482            RW,I,CON,LCL
 3      $IDATA  000024    10            RW,D,CON,LCL
 4      $VARS   000170    60            RW,D,CON,LCL

NTRY POINTS

NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

LP           1-000000

ARIABLES

NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

I      I*2   4-000160   J      I*2   4-000164   K      I*2   4-000162   L      I*2   4-000166

RRAYS

NAME   TYPE  ADDRESS       SIZE          DIMENSIONS

CHAR   L*1   4-000000  000014     6      (12)
PRNT   L*1   4-000014  000144    50      (100)
SHADE  L*1   F-000002* 000001     0      (1)

ABELS

LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

10         **           20      1-001534        21      1-000216        22      1-000272        23      1-000422
24      1-000476        25      1-000552        26      1-000626        27      1-000702        28      1-001002
29      1-001102        30      1-001174        31      1-001266        32      1-001366        33      1-001460
43      1-000346        50         **           51'        **           500'    3-000000        501'    3-000010

OTAL SPACE ALLOCATED = 002120   552

LP.LST=UPPER
```

```
      C       THIS SUBROUTINE ACCEPTS THE X,Y COORDINATES OF
      C       A POINTON THE CACHE MAP SHEET, AND RETURNS
      C       THE Z VALUE AT THAT POINT
      C
      C8***********************************************
      C
      C       SUBROUTINE BY CYRUS C. TAYLOR
      C       COMPUTER SCIENCE SPECIALIST(TRAINEE)
      C       AUTO-CARTO BR
      C       TDL,USAETL
      C       FORT BELVOIR, VA
      C       23 AUG 1977
      C
      C***********************************************
      C
              SUBROUTINE ALT(X,Y,Z,SN,AZ,BZ,IASCD)
              INTEGER COEF1
              LOGICAL*1 COEF2
              DIMENSION COEF1(90,40),BUF(512),COEF2(90,40,3)
              DIMENSION FIT1(4),FIT2(4),FIT3(4),FIT4(4)
              COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
              COMMON /NEW/MRDR
              IF(X.LT.XMAXB .AND. Y.LT.YMAXB
           1  .AND. X.GT.XMINB .AND. Y.GT.YMINB)GO TO 2077
              Z=ELEVB
              AZ=0.
              BZ=0.
              RETURN
        2077  CONTINUE
              DATA IFIRST/0/
              IF(IFIRST.GT.0) GO TO 111
              IFIRST=1
              ISAVE=0
              JSAVE=0
              BORDER=((5.E-4)*100./2.54)*8.
              FIT=(BORDER/8.)*9.
              DO 112 K=1,90
              CALL DSKIRN(4,1,IERR,K,NRES,BUF,0,IDONE)
              IF(IERR.GT.0) GO TO 9999
              DO 114 J=1,40
              L=4*(J-1)+6+328
              COEF1(K,J)=BUF(L)
              COEF2(K,J,1)=BUF(L+1)
              COEF2(K,J,2)=BUF(L+2)
              COEF2(K,J,3)=BUF(L+3)
       114    CONTINUE
       112    CONTINUE
       111    CONTINUE
              J=INT((X-BORDER)/FIT)+1
              I=INT((Y-BORDER)/FIT)+1
              J1=J+1
              IMOD=I-82
              IF(I.EQ.ISAVE.AND.J.EQ.JSAVE) GO TO 80
              IF(IMOD.LT.1) GO TO 9998
              IF(J.GT.90) GO TO 9998
              IB=(IMOD-1)*4
```

```
                FIT1(1)=COEF1(J,IMOD)
                FIT2(1)=COEF1(J,IMOD+1)
                FIT3(1)=COEF1(J1,IMOD)
                FIT4(1)=COEF1(J1,IMOD+1)
                DO 75 IP=2,4
                I1=IB+IP
                I2=IB+4+IP
                FIT1(IP)=COEF2(J,IMOD,IP-1)
                FIT2(IP)=COEF2(J,IMOD+1 ,IP-1)
                FIT3(IP)=COEF2(J1,IMOD,IP-1)
                FIT4(IP)=COEF2(J1,IMOD+1,IP-1)
        75      CONTINUE
        80      CONTINUE
                XC=FLOAT(J-1)*FIT+BORDER
                XL=(X-XC)/FIT
                YC=FLOAT(I-1)*FIT+BORDER
                YL=(Y-YC)/FIT
                XLC=XL-1.
                YLC=YL-1.
                Z1=FIT1(1)+FIT1(2)*XL + (FIT1(3)+FIT1(4)*XL) * YL
                Z2=FIT2(1)+FIT2(2)*XL + (FIT2(3)+FIT2(4)*XL) * YLC
                Z3=FIT3(1)+FIT3(2)*XLC + (FIT3(3)+FIT3(4)*XLC) * YL
                Z4=FIT4(1)+FIT4(2)*XLC + (FIT4(3)+FIT4(4)*XLC) * YLC
                XL2=ABS(XL**(MRDR+1))
                XLC2=ABS(XLC**(MRDR+1))
                YL2=ABS(YL**(MRDR+1))
                YLC2=ABS(YLC**(MRDR+1))
                IF(IASCD.EQ.0)GO TO 200
                AZ1=FIT1(2)+FIT1(4)*YL
                AZ2=FIT2(2)+FIT2(4)*YLC
                AZ3=FIT3(2)+FIT3(4)*YL
                AZ4=FIT4(2)+FIT4(4)*YLC
                BZ1=FIT1(3)+FIT1(4)*XL
                BZ2=FIT2(3)+FIT2(4)*XL
                BZ3=FIT3(3)+FIT3(4)*XLC
                BZ4=FIT4(3)+FIT4(4)*XLC
        200      CONTINUE
                XLC=-XLC
                YLC=-YLC
                W1=(XLC2*(-2.*XLC+3.))*(YLC2*(-2.*YLC+3.))
                W2=(XLC2*(-2.*XLC+3.))*(YL2*(-2.*YL+3.))
                W3=(XL2*(-2.*XL+3.))*(YLC2*(-2.*YLC+3.))
                W4=(XL2*(-2.*XL+3.))*(YL2*(-2.*YL+3.))
                IF(IASCD.EQ.0)GO TO 205
                ADW1=-6.*XLC*(1.-XLC)*(YLC2*(-2.*YLC+3.))
                ADW3=6.*XL*(1.-XL)*(YLC2*(-2.*YLC+3.))
                ADW2=-6.*XLC*(1.-XLC)*(YL2*(-2.*YL+3.))
                ADW4=6.*XL*(1.-XL)*(YL2*(-2.*YL+3.))
                BDW1=-6.*YLC*(1.-YLC)*(XLC2*(-2.*XLC+3.))
                BDW2=6.*YL*(1.-YL)*(XLC2*(-2.*XLC+3.))
                BDW3=-6.*YLC*(1.-YLC)*(XL2*(-2.*XL+3.))
                BDW4=6.*YL*(1.-YL)*(XL2*(-2.*XL+3.))
                AZ=Z1*ADW1+Z2*ADW2+Z3*ADW3+Z4*ADW4
                AZ=AZ+AZ1*W1+AZ2*W2+AZ3*W3+AZ4*W4
                BZ=Z1*BDW1+Z2*BDW2+Z3*BDW3+Z4*BDW4
                BZ=BZ+BZ1*W1+BZ2*W2+BZ3*W3+BZ4*W4
```

```
        205    Z=Z1*W1+Z2*W2+Z3*W3+Z4*W4
        210    CONTINUE
C                WRITE(6,101)AZ,BZ,Z1,Z2,Z3,Z4,W1,W2,W3,W4,ADW1,ADW2,
C            1   ,ADW3,ADW4,,BDW1,BDW2,BDW3,BDW4,AZ1,AZ2,AZ3,AZ4,BZ1,BZ2,
C            1   BZ3,BZ4
        101      FORMAT(1X,2F9.4,2X,2(4F9.4,2X),/,2(1X,3(4F9.4,2X),/),//)
        100      FORMAT(1X,I5,I5,2F10.3,F10.3,/)
                 JSAVE=J
                 ISAVE=I
                 IUSVE=IUNIT
                 RETURN
       9999      WRITE(5,99991)IERR
      99991      FORMAT('1I/O FAILURE ON DB1: - IERR = ',I3)
                 STOP
       9998      Z=310.
                 RETURN
                 END
```

FORTRAN IV-PLUS V02-51E 13:24:20 21-APR-81 PAGE 4
ALSEP.FTN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	003074	798	RW,I,CON,LCL
2	$PDATA	000024	10	RW,D,CON,LCL
3	$IDATA	000066	27	RW,D,CON,LCL
4	$VARS	047504	10146	RW,D,CON,LCL
5	$TEMPS	000022	9	RW,D,CON,LCL
6	BOUNDS	000024	10	RW,D,OVR,GBL
7	NEW	000002	1	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALI		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ADW1	R*4	4-047440	ADW2	R*4	4-047450	ADW3	R*4	4-047444	ADW4	R*4	4-047454	AZ	R*4	F-000012*
AZ1	R*4	4-047360	AZ2	R*4	4-047364	AZ3	R*4	4-047370	AZ4	R*4	4-047374	BDW1	R*4	4-047460
BDW2	R*4	4-047464	BDW3	R*4	4-047470	BDW4	R*4	4-047474	BORDER	R*4	4-047226	BZ	R*4	F-000014*
BZ1	R*4	4-047400	BZ2	R*4	4-047404	BZ3	R*4	4-047410	BZ4	R*4	4-047414	ELEVB	R*4	6-000020
FI1	R*4	4-047232	I	I*2	4-047252	IASCD	I*2	F-000016*	IB	I*2	4-047260	IDONE	I*2	4-047244
IERR	I*2	4-047240	IFIRST	I*2	4-047220	IMOD	I*2	4-047256	IP	I*2	4-047262	ISAVE	I*2	4-047222
IUNIT	I*2	4-047502	IUSVE	I*2	4-047500	I1	I*2	4-047264	I2	I*2	4-047266	J	I*2	4-047246
JSAVE	I*2	4-047224	J1	I*2	4-047254	K	I*2	4-047236	L	I*2	4-047250	MRDR	I*2	7-000000
NRES	I*2	4-047242	SN	R*4	F-000010*	W1	R*4	4-047420	W2	R*4	4-047424	W3	R*4	4-047430
W4	R*4	4-047434	X	R*4	F-000002*	XC	R*4	4-047270	X1	R*4	4-047274	XLC	R*4	4-047310
XLC2	R*4	4-047344	XL2	R*4	4-047340	XMAXB	R*4	6-000000	XMINB	R*4	6-000010	Y	R*4	F-000004*
YC	R*4	4-047300	YL	R*4	4-047304	YLC	R*4	4-047314	YLC2	R*4	4-047354	YL2	R*4	4-047350
YMAXB	R*4	6-000004	YMINB	R*4	6-000014	Z	R*4	F-000006*	Z1	R*4	4-047320	Z2	R*4	4-047324
Z3	R*4	4-047330	Z4	R*4	4-047334									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
BUF	R*4	4-043120	004000	1024	(512)
COEF1	I*2	4-000000	016040	3600	(90,40)
COEF2	L*1	4-016040	025060	5400	(90,40,3)
FI11	R*4	4-047120	000020	8	(4)
FI12	R*4	4-047140	000020	8	(4)
FI13	R*4	4-047160	000020	8	(4)
FI14	R*4	4-047200	000020	8	(4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS

```
 75       **          80     1-001102     100'    **         101'    **          111     1-000444
 112      **          114      **         200     1-001774   205     1-052700    210       **
 2077   1-000124      9998   1-003052     9999    1-003000   99991'  3-000000
```

FUNCTIONS AND SUBROUTINES REFERENCED

 DSKTR$

TOTAL SPACE ALLOCATED = 052762 11001

,LP.LST=ALSLP

```
C       THIS SUBROUTINE CALCULATES THE X,Y POSITIONS OF
C       ALL POINTS ALONG A LOS PROFILE AT WHICH THE
C       ELEVATION IS TO BE CALCULATED
C
C*************************************************
C
C       PROGRAM BY CYRUS C. TAYLOR
C
C       AUTO-CARTO BR
C       TDL,USAETL
C       TDL,UASETL
C       23 AUG 1977
C        PROGRAM MODIFIED 19 JANUARY 1979
C        IN ORDER TO BE COMPATIBLE WITH DUAL
C        ALTITUDE/SLOPE COMPUTATION
C        SUBROUTINE ALSLP.FTN, TO BE USED
C        BY SHADED RELIEF PERSPECTIVE SFTWR.
C        C. C. TAYLOR
C
C*************************************************
C
C
        SUBROUTINE PTS(X1,Y1,X2,Y2,NPTS,B,DLN,IASCD)
        DIMENSION B(1)
        COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
        FNPTS1=FLOAT(NPTS-1)
        DLN=(SQRT((X2-X1)**2+(Y2-Y1)**2))/FNPTS1
        DX=(X2-X1)/FNPTS1
        DY=(Y2-Y1)/FNPTS1
        SN=DY/DLN
        DO 100 I=1,NPTS
        FIM1=FLOAT(I-1)
        X=X1+FIM1*DX
        Y=Y1+FIM1*DY
        IF(X.LT.XMAXB .AND. Y.LT.YMAXB
     1  .AND. X.GT.XMINB .AND. Y.GT.YMINB) GO TO 50
        B(I)=ELEVB
        GO TO 100
 50     CALL ALT(X,Y,Z,SN,AZ,BZ,IASCD)
        B(I)=Z
 100    CONTINUE
        RETURN
        END
```

FORTRAN IV-PLUS V02-51E 13:25:08 21-APR-81 PAGE 2
PTS.FTN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000456	151	RW,I,CON,LCL
3	$IDATA	000032	13	RW,D,CON,LCL
4	$VARS	000052	21	RW,D,CON,LCL
5	$TEMPS	000002	1	RW,D,CON,LCL
6	BOUNDS	000024	10	RW,D,OVR,GBL

ENTRY POINT

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
PTS		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
A2	R*4	4-000042	B2	R*4	4-000046	DEN	R*4	F-000016*	DX	R*4	4-000004	DY	R*4	4-000010
ELEVB	R*4	6-000020	FIM1	R*4	4-000022	FNPTS1	R*4	4-000000	I	I*2	4-000020	IASCD	I*2	F-000020*
NPTS	I*2	F-000012*	SN	R*4	4-000014	X	R*4	4-000026	XMAXB	R*4	6-000000	XMINB	R*4	6-000010
X1	R*4	F-000002*	X2	R*4	F-000006*	Y	R*4	4-000032	YMAXB	R*4	6-000004	YMINB	R*4	6-000014
Y1	R*4	F-000004*	Y2	R*4	F-000010*	Z	R*4	4-000036						

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
B	R*4	F-000014*	000004	2	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
50	1-000356	100	1-000432						

FUNCTIONS AND SUBROUTINES REFERENCED

ABT SSQRT

TOTAL SPACE ALLOCATED = 000610 196

,LP,LST=PTS

```
      SUBROUTINE RGSTR(NRAD,SHADE)
      LOGICAL*1 SHADE(1)
      COMMON /REG/NRD1,NRD2,NRD3,NRD4,NRG1,NRG2,DX
      IF(NRAD.LE.NRD2.AND.NRAD.GE.NRD1)GO TO 10
      IF(NRAD.LE.NRD4.AND.NRAD.GE.NRD3)GO TO 20
      GO TO 100
10    IF(NRAD.NE.NRD1)GO TO 11
      DO 12 J=NRG1,NRG2
12    SHADE(J)=127
11    SHADE(NRG2)=127
      GO TO 100
20    IF(NRAD.NE.NRD4)GO TO 21
      DO 22 J=NRG1,NRG2
22    SHADE(J)=127
21    SHADE(NRG2)=127
      GO TO 100
      ENTRY IRGSTR
      NRD4=INT(13./DX+0.5)
      NRD3=INT(12.5/DX+0.5)
      NRD1=INT(1./DX+0.5)
      NRD2=INT(1.5/DX+0.5)
      NRG1=INT(8.5/DX+0.5)
      NRG2=INT(9./DX+0.5)
100   RETURN
      END
```

```
FORTRAN IV-PLUS V02-51E          13:25:26    21-APR-81            PAGE 2
RGSTR.FTN        /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME        SIZE                 ATTRIBUTES

  1      $CODE1   000462   153            RW,I,CON,LCL
  3      $IDATA   000012     5            RW,D,CON,LCL
  4      $VARS    000002     1            RW,D,CON,LCL
  5      $TEMPS   000002     1            RW,D,CON,LCL
  6      REG      000020     8            RW,D,OVR,GBL

ENTRY POINTS

 NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS

 IRGSTR        1-000272    RGSTR         1-000000

VARIABLES

 NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS

 DX      R*4   6-000014    J       I*2   4-000000    NRAD    I*2   F-000002*   NRD1    I*2   6-000000    NRD2    I*2   6-000002
 NRD3    I*2   6-000004    NRD4    I*2   6-000006    NRG1    I*2   6-000010    NRG2    I*2   6-000012

ARRAYS

 NAME    TYPE  ADDRESS        SIZE         DIMENSIONS

 SHADE   L*1   F-000004* 000001      0     (1)

LABELS

 LABEL   ADDRESS           LABEL   ADDRESS           LABEL   ADDRESS           LABEL   ADDRESS           LABEL   ADDRESS

 10      1-000062          11      1-000142          12          **            20      1-000166          21      1-000246
 22          **            100     1-000432

TOTAL SPACE ALLOCATED = 000520   168

,LP.LST=RGSTR
```

```
      SUBROUTINE GRNDPT(J,Y,NFLAG,XI,YI,I
     1 ,YMAX,IASCD,AZ,BZ,Z,ITST)
C THIS SUBROUTINE IS DESIGNED AS A MORE
C EFFICIENT IMAGE/OBJECT CORRELATION ROUTINE
C THAN THE ORIGINAL GRNDPT.FTN OF AUG 78.
C SEE "COMPUTER GENERATION OF SHADED RELIEF IMAGES
C FOR CARTOGRAPHIC APPLICATIONS" FOR
C DOCUMENTATION.  ALGORITHM AND ROUTINE
C BY C. TAYLOR, 16 AUG 79
      COMMON /AARGH/X,Y0,THETAP,D,NPTS,DX,
     1 H,VSF,SCL,TOL,DELR,DIS,THETA,Y1ST
      COMMON /HGRAA/Z1,Z2,CSIP,SSIP,CST,SST
      DIMENSION Y(1)
      IF(NFLAG.EQ.1)GO TO 30
      YS=J*DX
      IF(I.NE.1)GO TO 10
      Z1=Y(1)
      Z2=Y(2)
      Y(1)=-100.
      Y(2)=H-(Y(2)-Y1ST)*VSF*SCL
      Y(2)=7.-(Y(2)*DIS)/(DELR*CST)
   10 CONTINUE
      IF(YS.GE.Y(I).AND.YS.LE.Y(I+1))GO TO 20
      I=I+1
      IF(I.NE.NPTS)GO TO 70
      NFLAG=1
      GO TO 30
   70 Z1=Z2
      Z2=Y(I+1)
      Y(I+1)=H-(Z2-Y1ST)*VSF*SCL
      Y(I+1)=7.-(Y(I+1)*DIS)/(FLOAT(I)*DELR*
     1 CST)
      GO TO 10
   20 CONTINUE
      ZU=Z2
      ZL=Z1
      A=VSF*SCL
      ZP=ZU-ZL
      R=FLOAT(I-1)*DELR
      DELTA=DELR
      RL=R
      RU=R+DELR
      ITRIN=0
   50 CONTINUE
C FOLLOWING LINE COMMENTED OUT AFTER DE-
C BUGGING. 17 AUG 79. C. TAYLOR
D     WRITE(6,100)I,J,DELR,DELTA,ZP,ZL,ZU,CST
  100 FORMAT(1X,2I5,6F10.5)
      ANUM=(-H*DIS-(Y1ST-ZL)*A*DIS-ZP*A*RL*DIS/DELTA)
      DEN=(YS-7.)*CST-ZP*A*DIS/DELTA
      RAD=ANUM/DEN
      ITRIN=ITRIN+1
      IF(ITRIN.EQ.25)GO TO 60
      XI=X+RAD*CSIP
      YI=Y0+RAD*SSIP
      ITST=ITST+1
```

```
      CALL ALT(X1,Y1,Z,SN,AZ,BZ,IASCD)
      YTRY=H-(Z-Y1ST)*VSF*SCL
      YTRY=7.-(YTRY*DIS)/(RAD*CST)
      YERR=YTRY-YS
      IF(ABS(YERR).LT.(TOL*DX))GO TO 30
C TWO POSSIBILITIES:CONCAVE AND CONVEX
      IF(YERR.GT.0)GO TO 40
      ZP=ZU-Z
      ZL=Z
      RL=RAD
      DELTA=RU-RL
      RL=RAD
      GO TO 50
   40 ZP=Z-ZL
      ZU=Z
      RU=RAD
      DELTA=RU-RL
      GO TO 50
   60 CONTINUE
   30 CONTINUE
      RETURN
      END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001412	389	RW,I,CON,LCL
3	$IDATA	000032	13	RW,D,CON,LCL
4	$VARS	000076	31	RW,D,CON,LCL
6	AARGH	000066	27	RW,D,OVR,GBL
7	HGRAA	000030	12	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
GRNDPT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
A	R*4	4-000014	ANUM	R*4	4-000046	AZ	R*4	F-000022*	BZ	R*4	F-000024*	CST	R*4	7-000020
CSTP	R*4	7-000010	D	R*4	6-000014	DELR	R*4	6-000046	DELTA	R*4	4-000030	DEN	R*4	4-000052
DIS	R*4	6-000052	DX	R*4	6-000022	H	R*4	6-000026	I	I*2	F-000014*	IASCD	I*2	F-000020*
ITRIN	I*2	4-000044	ITST	I*2	F-000030*	J	I*2	F-000002*	NFLAG	I*2	F-000006*	NPTS	I*2	6-000020
R	R*4	4-000024	RAD	R*4	4-000056	RL	R*4	4-000034	RU	R*4	4-000040	SCL	R*4	6-000036
SN	R*4	4-000062	SST	R*4	7-000024	SSTP	R*4	7-000014	THETA	R*4	6-000056	THETAP	R*4	6-000010
TOL	R*4	6-000042	VSF	R*4	6-000032	X	R*4	6-000000	XI	R*4	F-000010*	YERR	R*4	4-000072
YI	R*4	F-000012*	YMAX	R*4	F-000016*	YS	R*4	4-000000	YTRY	R*4	4-000066	YO	R*4	6-000004
YIST	R*4	6-000062	Z	R*4	F-000026*	ZL	R*4	4-000010	ZP	R*4	4-000020	ZU	R*4	4-000004
Z1	R*4	7-000000	Z2	R*4	7-000004									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
Y	R*4	F-000004*	000004	2	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000232	20	1-000504	30	1-001410	40	1-001322	50	1-000626
60	1-001402	70	1-000332	100'	**				

FUNCTIONS AND SUBROUTINES REFERENCED

ALT

TOTAL SPACE ALLOCATED = 001660 472

,LP.LST=GRNDPT

```
SSLPLP/PR:0/FP/CP/NOTR=SSLPLP,[300,300]DSKTRN
SLPS,ALSLP,LPPER,SXSY,QUANT,RCSLP
/
UNITS=21
ASG=TI:1,TI:6
ASG=TI:8
ASG=DB0:14,DB2:12
//
```

```
        C THIS PROGRAM WILL PRODUCE AN OUTPUT 9 TRACK
        C  MAGNETIC TAPE FILE CONTAINING THE OPTICAL
        C  DENSITIES OF AN ORTHONORMAL IMAGE OF A USER-
        C  SELECTED AREA OF THE CACHE, OK MAP SHEET.
        C  THE USER MAY SELECT BETWEEN TWO FORMULA FOR
        C  THE DETERMINATION OF THE GRAY SHADES, SPECIFY
        C  THE POSITION OF THE SUN, THE PIXEL DENSITY,
        C  AND A VERTICAL SCALING FACTOR.
        C
        C*********************************************
        C
        C  PROGRAM BY CYRUS TAYLOR
        C  AUTOMATED CARTOGRAPHY BRANCH
        C  ENGINEER TOPOGRAPHIC LABS
        C  FORT BELVOIR, VA
        C  27 JUNE 1978
        C
        C*********************************************
        C DIMENSION THE TWO ARAYS THAT WILL
        C  CONTAIN THE DATA PTS FOR TWO SUCCESIVE
        C  ELEVATION PROFILES.  THESE WILL BE
        C  USED TO DETERMINE THE LOCAL SLOPE OF
        C  THE TERRAIN, WHICH, TOGETHOR WITH OTHER
        C  USER DETERMINED FACTORS, DETERMINES
        C  THE INTENSITY OF THE REFLECTED LIGHT.
0001          DIMENSION AZA(1024),BZA(1024)
        C DEFINE THE BYTE ARRAY CONTAINING THE
        C  DENSITY INFORMATION; THIS ARRAY WILL BE
        C  WRITTEN TO TAPE.
        C *FOLLOWING STATEMENT ADDED R. ROSENTHAL 2/13/79
0002          DIMENSION IBYTES(10)
        C FOLLOWING LINE ADDED FOR TIME STATEMENT
        C C. TAYLOR, 26 AUG 1979
0003          DIMENSION ATIM(2)
0004          LOGICAL*1 SHADE(1024)
0005          COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
0006          COMMON /SEXY/DELTAG,COSEL,IVSUN,ANG,C1,C2,S1,S2
0007          COMMON /NEW/MRDR
0008          DATA XMAXB/17.696/,XMINB/.1772/,YMAXB/21.59/
0009          DATA YMINB/.1772/,ELEVB/310./,IUNIT/0/,KODE/-1/
0010          DATA MODE/0/,IPAR/0/,IDEN/0/,ICTL/128/
0011          DATA KOUNT/128/
0012          IERR=0
0013          MRDR=1
0014          PI=3.14159
        C  WE HAVE NOW INITIALIZED TAPTRN FOR LUN1
        C  NOW WE WILL OPEN THE DISKFILES
0015          CALL DSKFIL(4,-1,'DB0','CACHE1.DAT',DUM,1024
            1  ,100,IFLAG,IERR)
        C         OPEN(UNIT=4,TYPE='OLD',ACCESS='DIRECT',READONLY,
        C     1   RECORDSIZE=493,SHARED,ASSOCIATEVARIABLE=IV)
0016          IF(IERR.EQ.0)GO TO 6
        C ALL WRITE STATEMENTS MODIFIED TO
        C WRITE TO UNIT 1; I.E., ALL STATEMENTS
        C 'WRITE(5,' CHANGED TO WRITE(1, . . .
        C CHANGES MADE TO ALLOW DELAYED RUNNING
```

```
      C  OF SOFTWARE. C. TAYLOR. 26 AUG 1979
0017           WRITE(1,104)IERR,IFLAG
0018      104  FORMAT(' OPEN FAILURE ON DB1:CACHE1.DAT',2I3)
0019           STOP
      C6       CONTINUE
0020        6  CALL DSKFIL(2,-1,'DB0','SDRLF.DAT',DUM,512,
          1    1024,IFLAG,IERR)
      C        OPEN(UNIT=9,NAME='FOR005.DAT',TYPE='OLD',ACCESS=
      C     1  'DIRECT',READONLY,RECORDSIZE=405,
      C     1  SHARED,ASSOCIATEVARIABLE=IW)
0021           IF(IERR.EQ.0)GO TO 7
0022           WRITE(1,107)IERR
0023      107  FORMAT(' OPEN FAILURE ON DB1:SDRLF.DAT',I3)
0024           STOP
0025        7  CONTINUE
      C NOW SET PARITY AND DENSITY OF TAPE
      C NOW DEFINE PARAMETERS
      C *NEXT 7 LINES ADDED R. ROSENTHAL 2/13/79
0026           WRITE(1,*)' ENTER DEVICE AND FILE NAME:  ''DDN:FILE.EXT'''
0027           WRITE(1,*)' TO INPUT FROM TERMINAL ENTER ''TI:'''
      C NEXT TWO READ STATEMENTS ALTERED TO READ FROM
      C UNIT 4 IN ORDER TO ALLOW DELAYED RUNS
0028           READ(8,5555)IBYTES
0029   5555    FORMAT(10A2)
0030           WRITE(1,*)' NUMBER OF CHARS IN FILE NAME'
0031           READ(8,*)NBYTES
0032           CALL ASSIGN (4,IBYTES,NBYTES)
      C *NEXT 3 COMMENTS ENTERED R. ROSENTHAL 2/13/79
      C PROMPT INPUTS ARE NOW FROM UNIT 4
      C INPUT PARAMETERS MAY NOW BE READ FROM DISK FILES
      C CHANGES MADE TO 'READ(5,*)' NOW 'READ(4,*)
0033     20    CONTINUE
0034           WRITE(1,100)
0035      100  FORMAT(1X,'ENTER MAPSHEET COORDINATES OF LOWER LEFT-HAND
          1    ',/,1X,'  AREA OF INTEREST')
0036           READ(4,*)XLL,YLL
0037           WRITE(1,101)
0038      101  FORMAT(1X,'ENTER HEIGHT AND WIDTH OF AREA OF INTEREST')
0039           READ(4,*)H,W
0040     130   FORMAT(' ENTER ORDER OF WEIGHTING FUNCTION TO BE USED')
0041           WRITE(1,102)
0042      102  FORMAT(1X,'ENTER # PIXELS/HORIZONTAL LINE')
0043           READ(4,*)NPIX
0044           WRITE(1,103)
0045      103  FORMAT(1X,'ENTER VERTICAL SCALING FACTOR')
0046           READ(4,*)VSF
0047           WRITE(1,106)
0048      106  FORMAT(1X,'ENTER 0 FOR LAMBERTS LAW,',/,1X,
          1    '1 FOR THE LOMMEL-SEELIGER LAW')
0049           READ(4,*)LAW
0050           WRITE(1,105)
0051      105  FORMAT(' ENTER ANGLE OF ELEVATION OF LIGHT,',/,'
          1    DIRECTION OF LIGHT (0=FROM N,90 FROM E)',/,
          1    ' AZIMUTHAL VARIATION')
0052           READ(4,*)EL1,AN1,RVIEW1
0053           WRITE(1,131)
```

```
0054      131    FORMAT(1X,'RELIEF CONTOURS?Y=1,/,CONTOUR
              1  INTERVAL (M)')
0055             READ(4,*)IRELCN,ICON
0056             IVSUN=0
         C FOLLOWING 3 LINES ADDED FOR DCMNTN PURPOSES
         C C. TAYLOR, 26 AUG 1979
0057             CALL TIME(ATIM)
0058             WRITE(1,134)(ATIM(IJKL),IJKL=1,2)
0059      134    FORMAT(1X,2A4)
0060             IF(RVIEW1.GT.1.)IVSUN=1
0061             AN1=ABS(AN1)
0062             AN=90.-AN1
0063             EL=(EL1*3.14159)/180.
0064             AN=(AN*3.14159)/180.
0065             ANG=((135.*3.14159)/180.)-AN
0066             RVIEW=RVIEW1*PI/360.
0067             RVIEW=(PI-2.*RVIEW)/2.
         C NOW DEFINE PARAMETER-DEPENDANT VARIABLES
0068             DELTA=W/FLOAT(NPIX)
0069             CNVRT=0.0007874016
0070             CNVRT=1./CNVRT
0071             IF(IRELCN.EQ.1)VSF=TAN(EL)*CNVRT*(DELTA/ICON)*VSF
0072             DELTAP=DELTA/(((5.E-4)*100./2.54)*9.)
0073             DELTAG=(DELTA*50000.)*(2.54/100.)
0074             NPTS=NPIX+1
0075             XEND=XLL+W
0076             LSTRW=H/DELTA
         C *FOLLOWING LINE INSERTED R. ROSENTHAL 16-FEB-79
0077             NUMRWS=LSTRW+1
0078             DELSQR=DELTAG**2
0079             YCUR=YLL+H
         C HOW PRECISE ARE THE FOLLOWING?
0080             C1=COS(-RVIEW+3.*PI/4.)
0081             C2=COS(-RVIEW-PI/4.)
0082             S1=SIN(RVIEW+3.*PI/4.)
0083             S2=SIN(RVIEW-PI/4.)
0084             SY=COS(EL)*SIN(AN)
0085             SX=COS(EL)*COS(AN)
0086             SZ=SIN(EL)
0087             COSEL=COS(EL)
         C ESTABLISH DENSITY SCALING FACTOR
         C   ASSUME DMAX=127 CORRESPONDS TO
         C   TO DENSITY OF 2.3
         C       DSCL=127./2.3
0088             WRITE(1,111)
0089      111    FORMAT(' ENTER DSCL (55.2)')
0090             READ(4,*)DSCL
0091      115    FORMAT(' ENTER 1 TO DUMP ELEVATIONS')
0092             WRITE(6,112)XLL,YLL,H,W
0093      112    FORMAT('1XLL,YLL=(',F7.3,',',F7.3,'),[illegible]',F7.3,',',F7.3,')')
0094             WRITE(6,113)NPIX,VSF,LAW,MRDR
0095      113    FORMAT(' NPIX=',I3,' VSF=',F7.3[illegible]LAW=',[illegible],'MRDR= ',I3)
0096             WRITE(6,114)EL1,AN1,DSCL,RVIEW1,NUMRWS
0097      114    FORMAT(' EL=',F7.3,'AN=',F8.3,'DSCL=',F7.3,'RVIEW=',
                 1F8.2,'NUMBER OF ROWS=',I4,//)
         C *ABOVE 3 LINES, ADDITION TO WRITE STATEMENT ADDED R. ROSENTHAL
```

```
C ESTABLISH DO-LOOP TO PROCESS DATA
              DO 1000 IROW=0,LSTRW
              YCUR=YCUR-DELTA
              IF(IRELCN.EQ.1)GO TO 132
              CALL SLPS(XLL,YCUR,XEND,YCUR,NPIX,AZA,BZA,DLN)
              GO TO 133
  132         CALL RCSLP(XLL,YCUR,XEND,YCUR,NPIX,AZA,BZA
     1        ,DLN,ICON,DELTAP)
  133         CONTINUE
              IF(LAW.EQ.1)GO TO 11
              DO 13 IPIX=1,NPIX
              AZ=AZA(IPIX)*DELTAP
              BZ=BZA(IPIX)*DELTAP
              AZ=AZ*VSF
              BZ=BZ*VSF
              CALL SXSY(SX,SY,AZ,BZ)
              BEG=-SX*AZ*DELTAG-SY*DELTAG*BZ+DELSQR*SZ
              DIS=SQRT((AZ*DELTAG)**2+(DELTAG*BZ)**2+(DELSQR**2))
              COSI=BEG/DIS
C OPACITY=I/10,DENSITY=LOG(OPACITY),I=10COSI
C NOTE:DENSITY MAY HAVE TO BE CHANGED
              IF(COSI.LT.1.E-5)COSI=.00001
              SHAD=ALOG10(1./COSI)
              SHAD=SHAD*DSCL
              IF(SHAD.GT.127.)SHAD=127.
              SHADE(IPIX)=SHAD
   13         CONTINUE
              GO TO 1500
C NOW DEAL WITH THE LOMMEL SEELIGER CASE
   11         CONTINUE
C             DIFFERENTIATE BETWEEN EVEN AND ODD ROWS
              DO 19 IPIX=1,NPIX
              AZ=AZA(IPIX)*DELTAP
              BZ=BZA(IPIX)*DELTAP
              AZ=AZ*VSF
              BZ=BZ*VSF
              CALL SXSY(SX,SY,AZ,BZ)
              BEG=DELTAG*(-SX*AZ-SY*BZ)+DELSQR*SZ
              DIS=SQRT((AZ*DELTAG)**2+(BZ*DELTAG)**2+(DELSQR**2))
              COSE=DELSQR/DIS
              COSI=BEG/DIS
              IF(COSI.LT.1.E-5)COSI=.00001
              SHAD=1./(1.+(COSE/COSI))
              SHAD=(ALOG10(1./SHAD))*DSCL
              IF(SHAD.LT.0.)SHAD=0.
              IF(SHAD.GT.127.)SHAD=127.
              SHADE(IPIX)=SHAD
   19         CONTINUE
 1500         CONTINUE
C NOW WRITE IT OUT TO DISK
              JROW=IROW+1
              CALL DSKTRN(2,2,IERR,JROW,NRES,SHADE,
     1        0,IDONE)
              IF(IERR.EQ.0)GO TO 15
              WRITE(1,108)IERR
  108         FORMAT(' DSKTRN FAILURE:',I3)
```

```
                     STOP
          15         CONTINUE
        1000         CONTINUE
      C WRITE AN END OF FILE MARKER
                     IF (NPIX.GT.100)GO TO 1501
                     DO 1502 JFIL=1,JROW
                     CALL DSKTRN(2,1,IERR,JFIL,NRES,SHADE,
           1         0,IDONE)
                     IF(IERR.EQ.0)GO TO 16
                     WRITE(1,108)IERR
                     STOP
                     CONTINUE
                     CALL LP(SHADE)
        1502         CONTINUE
        1501         CONTINUE
                     CALL TIME(ATIM)
                     WRITE(1,134)(ATIM(IJKL),IJKL=1,2)
                     WRITE(1,109)
         109         FORMAT(' ANOTHER IMAGE:ENTER 1')
                     READ(4,110)IANS
         110         FORMAT(I4)
                     IF(IANS.EQ.1)GO TO 20
                     CALL DSKFIL(4,-2,'DB0','CACHE1.DAT',DUM,1024,100
           1         ,IFLAG,IERR)
                     CALL DSKFIL(2,-2,'DB0','SDRLF.DAT',DUM,512,1024
           1         ,IFLAG,IERR)
                     STOP
                     END
```

RUGRAM SECTIONS

UMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	004164	1082	RW,I,CON,LCL
2	$PDATA	000306	99	RW,D,CON,LCL
3	$IDATA	001574	446	RW,D,CON,LCL
4	$VARS	022312	4709	RW,D,CON,LCL
5	$TEMPS	000014	6	RW,D,CON,LCL
6	BOUNDS	000024	10	RW,D,OVR,GBL
7	SEXY	000036	15	RW,D,OVR,GBL
8	NEW	000002	1	RW,D,OVR,GBL

ARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
AN	R*4	4-022142	ANG	R*4	7-000012	AN1	R*4	4-022124	AZ	R*4	4-022244	BEG	R*4	4-022254
BZ	R*4	4-022250	CNVRT	R*4	4-022162	COSE	R*4	4-022274	COSEL	R*4	7-000004	COSI	R*4	4-022264
C1	R*4	7-000016	C2	R*4	7-000022	DELSQR	R*4	4-022204	DELTA	R*4	4-022156	DELTAG	R*4	7-000000
DELTAP	R*4	4-022166	DIS	R*4	4-022260	DLN	R*4	4-022236	DSCL	R*4	4-022230	DUM	R*4	4-022060
EL	R*4	4-022146	ELEVB	R*4	6-000020	EL1	R*4	4-022120	H	R*4	4-022100	IANS	I*2	4-022310
ICON	I*2	4-022136	ICTL	I*2	4-022046	IDEN	I*2	4-022044	IDONE	I*2	4-022304	IERR	I*2	4-022052
IFLAG	I*2	4-022064	IJKL	I*2	4-022140	IPAR	I*2	4-022042	IPIX	I*2	4-022242	IRELCN	I*2	4-022134
IROW	I*2	4-022234	IUNIT	I*2	4-022034	IVSUN	I*2	7-000010	JFIL	I*2	4-022306	JROW	I*2	4-022300
KODE	I*2	4-022036	KOUNT	I*2	4-022050	LAW	I*2	4-022116	LSTRW	I*2	4-022200	MODE	I*2	4-022040
MRDR	I*2	8-000000	NBYTES	I*2	4-022066	NPIX	I*2	4-022110	NPTS	I*2	4-022172	NRES	I*2	4-022302
NUMRWS	I*2	4-022202	PI	R*4	4-022054	RVIEW	R*4	4-022152	RVIEW1	R*4	4-022130	SHAD	R*4	4-022270
SX	R*4	4-022220	SY	R*4	4-022214	SZ	R*4	4-022224	S1	R*4	7-000026	S2	R*4	7-000032
VSF	R*4	4-022112	W	R*4	4-022104	XEND	R*4	4-022174	XLL	R*4	4-022070	XMAXB	R*4	6-000000
XMINB	R*4	6-000010	YCUR	R*4	4-022210	YLL	R*4	4-022074	YMAXB	R*4	6-000004	YMINB	R*4	6-000014

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
ATIM	R*4	4-020024	000010	4	(2)
AZA	R*4	4-000000	010000	2048	(1024)
BZA	R*4	4-010000	010000	2048	(1024)
IBYTES	I*2	4-020000	000024	10	(10)
SHADE	L*1	4-020034	002000	512	(1024)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
6	1-000136	7	1-000220	11	1-003060	13	**	15	1-003560
16	1-003714	19	**	20	1-000406	100'	3-000116	101'	3-000230
102'	3-000306	103'	3-000350	104'	3-000000	105'	3-000510	106'	3-000412
107'	3-000046	108'	3-001212	109'	3-001240	110'	3-001272	111'	3-000750
112'	3-000776	113'	3-001046	114'	3-001116	115'	**	130'	**
131'	3-000662	132	1-002462	133	1-002500	134'	3-000742	1000	**
1500	1-003464	1501	1-003746	1502	**	5555'	3-000112		

```
FUNCTIONS AND SUBROUTINES REFERENCED

 ASSIGN  DSKFIL  DSKTRN  LP      RCSLP   SLPS    SASY    TIME    $ALG10  $COS    $SIN    $SQRT   $TAN

TOTAL SPACE ALLOCATED = 030700  6368

,LP.LST=SSLPLP
```

```
C       THIS SUBROUTINE CALCULATES THE X,Y POSITIONS OF
C       ALL POINTS ALONG A LOS PROFILE AT WHICH THE
C       ELEVATION IS TO BE CALCULATED
C
C**************************************************
C
C       PROGRAM BY CYRUS C. TAYLOR
C
C       AUTO-CARTO BR
C       TDL,USAETL
C       TDL,UASETL
C       23 AUG 1977
C
C**************************************************
C
C
        SUBROUTINE SLPS(X1,Y1,X2,Y2,NPTS,AZA,BZA,DLN)
        DIMENSION AZA(1),BZA(1)
        COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
        FNPTS1=FLOAT(NPTS-1)
        DLN=(SQRT((X2-X1)**2+(Y2-Y1)**2))/FNPTS1
        DX=(X2-X1)/FNPTS1
        DY=(Y2-Y1)/FNPTS1
        SN=DY/DLN
        DO 100 I=1,NPTS
        FIM1=FLOAT(I-1)
        X=X1+FIM1*DX
        Y=Y1+FIM1*DY
        IF(X.LT.XMAXB .AND. Y.LT.YMAXB
     1 .AND. X.GT.XMINB .AND. Y.GT.YMINB) GO TO 50
        AZA(I)=0.
        BZA(I)=0.
        GO TO 100
 50     CALL ALT(X,Y,Z,SN,AZ,BZ,1)
        AZA(I)=AZ
        BZA(I)=BZ
 100    CONTINUE
        RETURN
        END
```

```
FORTRAN IV-PLUS V02-51E        13:28:00    21-APR-81          PAGE 2
SLPS.FTN        /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME       SIZE                 ATTRIBUTES

  1      $CODE1   000512    165          RW,I,CON,LCL
  2      $PDATA   000004      2          RW,D,CON,LCL
  3      $IDATA   000044     18          RW,D,CON,LCL
  4      $VARS    000052     21          RW,D,CON,LCL
  5      $TEMPS   000002      1          RW,D,CON,LCL
  6      BOUNDS   000024     10          RW,D,OVR,GBL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 SLPS         1-000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 A2     R*4   4-000042   B2     R*4   4-000046   DEN    R*4   F-000020*  DX     R*4   4-0000[illegible]4  [illegible]  R*4   4-[illegible]
 ELEVB  R*4   6-000020   FIM1   R*4   4-000022   FNFIS1 R*4   4-000000   I      I*2   4-000020   NPTS   I*2   F-[illegible]12*
 SN     R*4   4-000014   X      R*4   4-000026   XMAXB  R*4   6-000000   XMINB  R*4   6-00001[illegible]  X1     R*4   F-[illegible]2*
 X2     R*4   F-000036*  Y      R*4   4-000032   YMAXB  R*4   6-000004   YMINB  R*4   6-0000[illegible]4  Y1     R*4   F-[illegible]4*
 Y2     R*4   F-00001[illegible]*  [illegible]  R*4   4-0000[illegible]6

ARRAYS

 NAME   TYPE  ADDRESS         SIZE        DIMENSIONS

 A2A    R*4   F-0000[illegible]* 000004      2       (1)
 B2A    R*4   F-000016* 000004      2       (1)

LABELS

 LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

 5[illegible]      1-000[illegible]         [illegible]     1-000[illegible]

FUNCTIONS AND [illegible] REFERENCED

 ALI      [illegible]

TOTAL SPACE ALLOCATED = [illegible]

[illegible]
```

```
C       THIS SUBROUTINE ACCEPTS THE X,Y COORDINATES OF
C       A POINTON THE CACHE MAP SHEET, AND RETURNS
C       THE Z VALUE AT THAT POINT
C
C8*************************************************
C
C       SUBROUTINE BY CYRUS C. TAYLOR
C       COMPUTER SCIENCE SPECIALIST(TRAINEE)
C       AUTO-CARTO BR
C       TDL,USAETL
C       FORT BELVOIR, VA
C       23 AUG 1977
C
C************************************************
C
        SUBROUTINE ALT(X,Y,Z,SN,AZ,BZ,IASCD)
        INTEGER COEF1
        LOGICAL*1 COEF2
        DIMENSION COEF1(90,40),BUF(512),COEF2(90,40,3)
        DIMENSION FIT1(4),FIT2(4),FIT3(4),FIT4(4)
        COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
        COMMON /NEW/MRDR
        IF(X.LT.XMAXB .AND. Y.LT.YMAXB
     1  .AND. X.GT.XMINB .AND. Y.GT.YMINB)GO TO 2077
        Z=ELEVB
        AZ=0.
        BZ=0.
        RETURN
 2077   CONTINUE
        DATA IFIRST/0/
        IF(IFIRST.GT.0) GO TO 111
        IFIRST=1
        ISAVE=0
        JSAVE=0
        BORDER=((5.E-4)*100./2.54)*8.
        FIT=(BORDER/8.)*9.
        DO 112 K=1,90
        CALL DSKTRN(4,1,IERR,K,NRES,BUF,0,IDONE)
        IF(IERR.GT.0) GO TO 9999
        DO 114 J=1,40
        L=4*(J-1)+6+328
        COEF1(K,J)=BUF(L)
        COEF2(K,J,1)=BUF(L+1)
        COEF2(K,J,2)=BUF(L+2)
        COEF2(K,J,3)=BUF(L+3)
  114   CONTINUE
  112   CONTINUE
  111   CONTINUE
        J=INT((X-BORDER)/FIT)+1
        I=INT((Y-BORDER)/FIT)+1
        J1=J+1
        IMOD=I-82
        IF(I.EQ.ISAVE.AND.J.EQ.JSAVE) GO TO 80
        IF(IMOD.LT.1) GO TO 9998
        IF(J.GT.90) GO TO 9998
        IB=(IMOD-1)*4
```

```
      FIT1(1)=COEF1(J,IMOD)
      FIT2(1)=COEF1(J,IMOD+1)
      FIT3(1)=COEF1(J1,IMOD)
      FIT4(1)=COEF1(J1,IMOD+1)
      DO 75 IP=2,4
      I1=IB+IP
      I2=IB+4+IP
      FIT1(IP)=COEF2(J,IMOD,IP-1)
      FIT2(IP)=COEF2(J,IMOD+1 ,IP-1)
      FIT3(IP)=COEF2(J1,IMOD,IP-1)
      FIT4(IP)=COEF2(J1,IMOD+1,IP-1)
75    CONTINUE
80    CONTINUE
      XC=FLOAT(J-1)*FIT+BORDER
      XL=(X-XC)/FIT
      YC=FLOAT(I-1)*FIT+BORDER
      YL=(Y-YC)/FIT
      XLC=XL-1.
      YLC=YL-1.
      Z1=FIT1(1)+FIT1(2)*XL + (FIT1(3)+FIT1(4)*XL) * YL
      Z2=FIT2(1)+FIT2(2)*XL + (FIT2(3)+FIT2(4)*XL) * YLC
      Z3=FIT3(1)+FIT3(2)*XLC + (FIT3(3)+FIT3(4)*XLC) * YL
      Z4=FIT4(1)+FIT4(2)*XLC + (FIT4(3)+FIT4(4)*XLC) * YLC
      XL2=ABS(XL**(MRDR+1))
      XLC2=ABS(XLC**(MRDR+1))
      YL2=ABS(YL**(MRDR+1))
      YLC2=ABS(YLC**(MRDR+1))
      IF(IASCD.EQ.0)GO TO 200
      AZ1=FIT1(2)+FIT1(4)*YL
      AZ2=FIT2(2)+FIT2(4)*YLC
      AZ3=FIT3(2)+FIT3(4)*YL
      AZ4=FIT4(2)+FIT4(4)*YLC
      BZ1=FIT1(3)+FIT1(4)*XL
      BZ2=FIT2(3)+FIT2(4)*XL
      BZ3=FIT3(3)+FIT3(4)*XLC
      BZ4=FIT4(3)+FIT4(4)*XLC
200    CONTINUE
      XLC=-XLC
      YLC=-YLC
      W1=(XLC2*(-2.*XLC+3.))*(YLC2*(-2.*YLC+3.))
      W2=(XLC2*(-2.*XLC+3.))*(YL2*(-2.*YL+3.))
      W3=(XL2*(-2.*XL+3.))*(YLC2*(-2.*YLC+3.))
      W4=(XL2*(-2.*XL+3.))*(YL2*(-2.*YL+3.))
      IF(IASCD.EQ.0)GO TO 205
      ADW1=-6.*XLC*(1.-XLC)*(YLC2*(-2.*YLC+3.))
      ADW3=6.*XL*(1.-XL)*(YLC2*(-2.*YLC+3.))
      ADW2=-6.*XLC*(1.-XLC)*(YL2*(-2.*YL+3.))
      ADW4=6.*XL*(1.-XL)*(YL2*(-2.*YL+3.))
      BDW1=-6.*YLC*(1.-YLC)*(XLC2*(-2.*XLC+3.))
      BDW2=6.*YL*(1.-YL)*(XLC2*(-2.*XLC+3.))
      BDW3=-6.*YLC*(1.-YLC)*(XL2*(-2.*XL+3.))
      BDW4=6.*YL*(1.-YL)*(XL2*(-2.*XL+3.))
      AZ=Z1*ADW1+Z2*ADW2+Z3*ADW3+Z4*ADW4
      AZ=AZ+AZ1*W1+AZ2*W2+AZ3*W3+AZ4*W4
      BZ=Z1*BDW1+Z2*BDW2+Z3*BDW3+Z4*BDW4
      BZ=BZ+BZ1*W1+BZ2*W2+BZ3*W3+BZ4*W4
```

```
 205   Z=Z1*w1+Z2*w2+Z3*w3+Z4*w4
 210   CONTINUE
C       wRITE(6,101)AZ,BZ,Z1,Z2,Z3,Z4,w1,w2,w3,w4,ADw1,ADw2,
C     1 ,ADw3,ADw4,,BDw1,BDw2,BDw3,BDw4,AZ1,AZ2,AZ3,AZ4,BZ1,BZ2,
C     1 BZ3,BZ4
 101    FORMAT(1X,2F9.4,2X,2(4F9.4,2X),/,2(1X,3(4F9.4,2X),/),//)
 100    FORMAT(1X,I5,I5,2F10.3,F10.3,/)
        JSAVE=J
        ISAVE=I
        IUSVE=IUNIT
        RETURN
9999    wRITE(5,99991)IERR
99991   FORMAT('1I/O FAILURE ON DB1: - IERR = ',I3)
        STOP
9998    Z=310.
        RETURN
        END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	003074	798	RW,I,CON,LCL
2	$PDATA	000024	10	RW,D,CON,LCL
3	$IDATA	000066	27	RW,D,CON,LCL
4	$VARS	047504	10146	RW,D,CON,LCL
5	$TEMPS	000022	9	RW,D,CON,LCL
6	BOUNDS	000024	10	RW,D,OVR,GBL
7	NEW	000002	1	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ADW1	R*4	4-047440	ADW2	R*4	4-047450	ADW3	R*4	4-047444	ADW4	R*4	4-047454	A2	R*4	F-000012*
AZ1	R*4	4-047360	AZ2	R*4	4-047364	AZ3	R*4	4-047370	AZ4	R*4	4-047374	BDW1	R*4	4-047460
BDW2	R*4	4-047464	BDW3	R*4	4-047470	BDW4	R*4	4-047474	[illegible]	R*4	4-047226	B2	R*4	F-000014*
BZ1	R*4	4-047400	BZ2	R*4	4-047404	BZ3	R*4	4-047410	BZ4	R*4	4-047414	ELEVB	R*4	6-000020
FII	R*4	4-047232	I	I*2	4-047252	IASCD	I*2	F-000016*	IF	I*2	4-047260	ICONE	I*2	4-047244
IERR	I*2	4-047240	IFIRST	I*2	4-047220	IMOD	I*2	4-047256	IP	I*2	4-047262	ISAVE	I*2	4-047222
IUNIT	I*2	4-047502	IUSTE	I*2	4-047500	I1	I*2	4-047264	I2	I*2	4-047266	[illegible]	I*2	4-047246
JSAVE	I*2	4-047224	J1	I*2	4-047254	K	I*2	4-047236	L	I*2	4-047250	MRDF	I*2	7-000000
NRES	I*2	4-047242	SN	R*4	F-000010*	W1	R*4	4-047420	W2	R*4	4-047424	W3	R*4	4-047430
W4	R*4	4-047434	X	R*4	F-000002*	XC	R*4	4-047270	XD	R*4	4-047274	XLC	R*4	4-047310
XLC2	R*4	4-047344	XL2	R*4	4-047340	XMAX	R*4	6-000000	XSTEP	R*4	6-000010	Y	R*4	F-000004*
YC	R*4	4-047300	YL	R*4	4-047304	YLC	R*4	4-047314	YLC2	R*4	4-047354	YL2	R*4	4-047350
YMAXB	R*4	6-000004	YMINE	R*4	6-000014	Z	R*4	F-000006*	Z1	R*4	4-047320	Z2	R*4	4-047324
Z3	R*4	4-047330	Z4	R*4	4-047334									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
[illegible]	R*4	4-043120	[illegible]	1024	(512)
COEF1	I*2	4-000000	[illegible]	3600	(90,40)
COEF2	L*1	4-010040	025000	5400	(90,40,3)
FIT1	R*4	4-047120	000020	8	(4)
FIT2	R*4	4-047140	000020	8	(4)
FIT3	R*4	4-047160	000020	8	(4)
FIT4	R*4	4-047200	000020	8	(4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS

```
 75      **           80     1-001102     100*     **          101*     **          111     1-000444
 112     **           114    **           200      1-001774    205      1-002700    210     **
 2077    1-0.0124     9998   1-003052     9999     1-003006    99991*   3-000000
```

FUNCTIONS AND SUBROUTINES REFERENCED

DSKTRN

TOTAL SPACE ALLOCATED = 052762 11001

,LP.LST=ALSLP

```
C     FILE NAME - LPPER.FTN
C
            SUBROUTINE LP(SHADE)
            LOGICAL*1 SHADE(1),CHAR(12)
            LOGICAL*1 PRNT(100)
            DATA CHAR/' ', '.', '-', '1', '/', 'V', '*', '@', 'U',
     1                'A','4','W'/
C           WRITE(5,51)(SHADE(M),M=1,32)
  51        FORMAT(1X,32O3,/)
            DO 10 I=1,3
            DO 50 K=1,100
            IF(SHADE(K).LT.0.)SHADE(K)=127.
            PRNT(K)=CHAR(1)
   50       CONTINUE
            DO 20 J=1,100
            IF(SHADE(J).GT.6)GO TO 21
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(1)
            GO TO 20
   21       IF(SHADE(J).GT.9)GO TO 22
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(2)
            GO TO 20
    22      IF(SHADE(J).GT.13)GO TO 43
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(3)
            GO TO 20
  43        IF(SHADE(J).GT.15)GO TO 23
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(4)
            GO TO 20
   23       IF(SHADE(J).GT.19)GO TO 24
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(5)
            GO TO 20
    24      IF(SHADE(J).GT.24)GO TO 25
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(6)
            GO TO 20
    25      IF(SHADE(J).GT.28)GO TO 26
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(7)
            GO TO 20
   26       IF(SHADE(J).GT.33)GO TO 27
            IF(I.NE.1)GO TO 20
            PRNT(J)=CHAR(8)
            GO TO 20
    27      IF(SHADE(J).GT.38)GO TO 28
            IF(I.EQ.3)GO TO 20
            IF(I.EQ.1)PRNT(J)=CHAR(9)
            IF(I.EQ.2)PRNT(J)=CHAR(2)
            GO TO 20
   28     IF(SHADE(J).GT.44)GO TO 29
            IF(I.EQ.3)GO TO 20
            IF(I.EQ.1)PRNT(J)=CHAR(10)
            IF(I.EQ.2)PRNT(J)=CHAR(5)
```

```
              GO TO 20
      29      IF(SHADE(J).GT.51)GO TO 30
              IF(I.EQ.3)GO TO 20
              IF(I.EQ.1)PRNT(J)=CHAR(10)
              IF(I.EQ.2)PRNT(J)=CHAR(5)
              GO TO 20
      30      IF(SHADE(J).GT.58)GO TO 31
              IF(I.EQ.3)GO TO 20
              IF(I.EQ.1)PRNT(J)=CHAR(10)
              IF(I.EQ.2)PRNT(J)=CHAR(8)
              GO TO 20
      31      IF(SHADE(J).GT.66)GO TO 32
              IF(I.EQ.3)PRNT(J)=CHAR(4)
              IF(I.EQ.1)PRNT(J)=CHAR(6)
              IF(I.EQ.2)PRNT(J)=CHAR(10)
              GO TO 20
      32      IF(SHADE(J).GT.76)GO TO 33
              PRNT(J)=CHAR(11)
              IF(I.EQ.1)PRNT(J)=CHAR(10)
              IF(I.EQ.2)PRNT(J)=CHAR(9)
              GO TO 20
      33      PRNT(J)=CHAR(12)
              IF(I.EQ.1)PRNT(J)=CHAR(10)
              IF(I.EQ.2)PRNT(J)=CHAR(5)
      20      CONTINUE
              WRITE(6,500)(PRNT(L),L=1,100)
      500     FORMAT('+',1X,100A1)
      10      CONTINUE
              WRITE(6,501)
      501     FORMAT(1X)
              RETURN
              END
```

```
FORTRAN IV-PLUS V02-51E          13:29:55    21-APR-81          PAGE 3
UPPER.FTN        /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME       SIZE                 ATTRIBUTES

  1      $CODE1  001704   482            RW,I,CON,LCL
  3      $IDATA  000024    10            RW,D,CON,LCL
  4      $VARS   000170    60            RW,D,CON,LCL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 LP           1-000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 I      I*2   4-000160   J      I*2   4-000164   K      I*2   4-000[illegible]   L      I*2   4-000[illegible]

ARRAYS

 NAME   TYPE  ADDRESS       SIZE         DIMENSIONS

 CHAR   L*1   4-000000  000014     6     (12)
 PRNT   L*1   4-000014  000144    50     (100)
 SHADE  L*1   F-000002* 000001     0     (1)

LABELS

 LABEL   ADDRESS          LABEL   ADDRESS          LABEL   ADDRESS          LABEL   ADDRESS          LABEL   ADDRESS

 10         **            20      1-000134         21      1-000216         22      1-000272         23      1-000422
 24      1-000476         25      1-000552         26      1-000626         27      1-000702         28      1-001062
 29      1-001102         30      1-001177         31      1-001266         32      1-001366         33      1-0014[illegible]
 43      1-000346         50         **            51'        **            500'    3-000000         501'    [illegible]

TOTAL SPACE ALLOCATED = 002120   552

,LP,LST=UPPER
```

```
      C THIS SUBROUTINE IS DESIGNED TO VARY THE AZIMUTH OF THE
      C  'SUN' TO STRENGTHEN THE DETAIL IN SHADED RELIEF IMAGES
      C  CREATED BY PROGRAMS SHADE.FTN AND SHADLP.FTN.
      C REFERENCE: P. YOELI, 'SOME REMARKS ON THE COMPLETION
      C  OF THE ANALYTICAL HILL SADING PROJECT'
      C  (UNPUBLISHED?-AUTO CARTO FILES)
      C
      C        PROGRAM BY CYRUS C. TAYLOR
      C        AUTOMATED CARTOGRAPHY BRANCH
      C        USAETL, FORT BELVOIR, VA
      C        11 JULY 1978
      C
      C
0001           SUBROUTINE SXSY(SX,SY,AZ,BZ)
0002           COMMON /SEXY/DELTAG,COSEL,IVSUN,ANG,C1,C2,S1,S2
0003           REAL*4 NX,NY
0004           IF(IVSUN.EQ.0)GO TO 110
      C
0005           NX=-AZ*DELTAG
0006           NY=-BZ*DELTAG
0007           DISFI=SQRT((NX**2)+(NY**2))
0008           IF(DISFI.LT.1.E-6)GO TO 110
0009           C0=NX/DISFI
0010           S0=NY/DISFI
0011           C=C0*COS(ANG)-S0*SIN(ANG)
0012           S=S0*COS(ANG)+C0*SIN(ANG)
      C IN EFFECT, OUR X-Y COORDINATE SYSTEM HAS BEEN
      C  ROTATED THROUGH AN ANGLE OF ANG RADIANS.  NOTE
      C  THAT THE ANGULAR COORDINATE SYSTEM USED IN THIS
      C  ROUTINE AND IN SHADE OR SHADLP IS NOT THE
      C  AERONAUTICAL ANGULAR COORDINATE SYSTEM, BUT THE
      C  TRADITIONAL ALGEBRAIC COORDINATE SYSTEM.  SINCE
      C  OUR X-Y COORDINATE SYSTEM HAS BEEN ROTATED,
      C  SX AND SY MUST BE ROTATED BACK BEFORE RETURNING
      C  TO THE MAIN ROUTINE.
      C SECTOR I CASE
0013           IF(C.GT.(C1). OR .S.LT.(S1))GO TO 10
0014           SX=-C*COSEL
0015           SY=-S*COSEL
0016           GO TO 100
      C SECTOR III CASE
0017    10     IF(C.LT.(C2). OR .S.GT.(S2))GO TO 20
0018           SX=C*COSEL
0019           SY=S*COSEL
0020           GO TO 100
      C SECTOR II CASE: 2 SUBCASES
0021    20     IF((C.LT.0.AND.S.LT.0).OR.(C.LT.0.AND.S.LT.S1)
             1   .OR.(S.LT.0.AND.C.LT.C2))GO TO 30
0022           IF(C.GT.(0.7071))GO TO 25
0023           SX=0.
0024           SY=-COSEL
0025           GO TO 100
0026    25     SX=COSEL
0027           SY=0.
0028           GO TO 100
      C SECTOR IV CASE: 2 SUBCASES
```

```
  30    IF(C.LT.(-0.7071))GO TO 35
        SX=0.
        SY=-COSEL
        GO TO 100
  35    SX=COSEL
        SY=0.
 100    CONTINUE
C  WE MUST NOW ROTATE OUR LOCAL X-Y COORDINATE
C  SYSTEM BACK, SO THAT SX AND SY WILL BE CONSISTANT
C  WITH THE MAIN ROUTINES.
        Y=SY*COS(ANG)-SX*SIN(ANG)
        X=SX*COS(ANG)+SY*SIN(ANG)
        SX=-X
        SY=-Y
 110  CONTINUE
        RETURN
        END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001000	256	RW,I,CON,LCL
2	$PDATA	000014	6	RW,D,CON,LCL
4	$VARS	000044	18	RW,D,CON,LCL
5	$TEMPS	000004	2	RW,D,CON,LCL
6	SEXY	000036	15	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
SXSY		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ANG	R*4	6-000012	A2	R*4	F-000006*	B2	R*4	F-000010*	C	R*4	4-000024	COSEL	R*4	6-000004
C0	R*4	4-000014	C1	R*4	6-000016	C2	R*4	6-000022	DELTAG	R*4	6-000000	DIST1	R*4	4-000010
IVSUN	I*2	6-000010	NX	R*4	4-000000	NY	R*4	4-000004	S	R*4	4-000030	SX	R*4	F-000002*
SY	R*4	F-000004*	S0	R*4	4-000020	S1	R*4	6-000026	S2	R*4	6-000032	X	R*4	4-000040
Y	R*4	4-000034												

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000320	20	1-000412	25	1-000542	30	1-000572	35	1-000634
100	1-000660	110	1-000776						

FUNCTIONS AND SUBROUTINES REFERENCED

$COS $SIN $SQRT

TOTAL SPACE ALLOCATED = 001122 297

,LP,LST=SXSY

```
0001        SUBROUTINE QUANT(Z,ICON)
0002        Z=FLOAT(ICON*(INT(Z/CON)))
0003        RETURN
0004        END
```

AD-A101 422 ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA F/G 8/2
SHADED RELIEF IMAGES FOR CARTOGRAPHIC APPLICATIONS.(U)
APR 81 C C TAYLOR
UNCLASSIFIED ETL-0259 NL

3 of 3
AD A
101 422

END
DATE
FILMED
8-81
DTIC

FORTRAN IV-PLUS V02-51E 13:37:21 21-APR-81 PAGE 2
QUANT.FTN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000054	22	RW,I,CON,LCL
4	$VARS	000004	2	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
QUANT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CON	R*4	4-000000	ICON	I*2	F-000004*	Z	R*4	F-000002*						

TOTAL SPACE ALLOCATED = 000060 24

,LP,LST=QUANT

```
0001            SUBROUTINE RCSLP(XLL,YCUR,XEND,Y,NPIX,AZA,
              1   BZA,DLN,ICON,DELTAP)
        C
        C THIS SUBROUTINE CALCULATES ALTITUDES, QUANTIZES
        C   THE ELEVATION VALUES, AND CALCULATES SLOPES OVER
        C   A FINE GRID, IN ORDER TO SIMULATE TANAKA'S
        C   RELIEF CONTOUR REPRESENTATION OF TERRAIN.
        C
        C THIS SUBROUTINE IS RESTRICTED TO RECTANGULAR REGIONS
        C   WITH EDGES PARALLEL TO THE COORDINATE AXES OF THE
        C   POLYNOMIAL TERRAIN MODEL
        C
        C PROGRAM BY CYRUS TAYLOR
        C               AUTO-CARTO BR.
        C               USAETL
        C               FORT BELVOIR, VA
        C               11 JUNE 1979
        C
0002            DIMENSION AZA(1),BZA(1),ELVES(2,2)
0003            COMMON /BOUNDS/XMAXB,YMAXB,XMINB,YMINB,ELEVB
0004            FNPTS1=FLOAT(NPIX-1)
0005            DLN=(SQRT((XEND-XLL)**2+(YCUR-YEND)**2))/FNPTS1
0006            DY=(YEND-YCUR)/FNPTS1
0007            SN=DY/DLN
0008            IASCD=0
0009            DX=(XEND-XLL)/FNPTS1
0010            DX2=DX/2.
0011            X1=XLL-DX2
0012            X2=XLL+DX2
0013            Y1=YCUR-DX2
0014            Y2=YCUR+DX2
        C NOTE THAT WE ASSUME THAT YCUR=YEND, AS WILL BE THE CASE
        C   FOR ALL CALLS FROM SSLPLP.
        C
0015            CALL ALT(X1,Y1,Z,SN,AZ,BZ,IASCD)
0016            CALL QUANT(Z,ICON)
0017            ELVES(1,1)=Z
0018            CALL ALT(X2,Y1,Z,SN,AZ,BZ,IASCD)
0019            CALL QUANT(Z,ICON)
0020            ELVES(2,1)=Z
0021            CALL ALT(X1,Y2,Z,SN,AZ,BZ,IASCD)
0022            CALL QUANT(Z,ICON)
0023            ELVES(1,2)=Z
0024            CALL ALT(X2,Y2,Z,SN,AZ,BZ,IASCD)
0025            CALL QUANT(Z,ICON)
0026            ELVES(2,2)=Z
0027            DO 100 I=1,NPIX
0028            IF(I.EQ.2*(I/2))GO TO 101
0029            AZ=0.5*((ELVES(2,1)-ELVES(1,1))+(ELVES(2,2)-ELVES(1,2)))
0030            AZ=AZ/DELTAP
0031            AZA(I)=AZ
0032            BZ=0.5*((ELVES(1,2)-ELVES(1,1))+(ELVES(2,2)-ELVES(2,1)))
0033            BZ=BZ/DELTAP
0034            BZA(I)=BZ
0035            X1=X1+DX
0036            CALL ALT(X1,Y1,Z,SN,AZ,BZ,IASCD)
```

```
            CALL QUANT(Z,ICON)
            ELVES(1,1)=Z
            CALL ALT(X1,Y2,Z,SN,AZ,BZ,IASCD)
            CALL QUANT(Z,ICON)
            ELVES(1,2)=Z
            GO TO 100
      101   AZ=0.5*(ELVES(1,1)-ELVES(2,1)+ELVES(1,2)-ELVES(2,2))
            AZ=AZ/DELTAP
            AZA(I)=AZ
            BZ=0.5*(ELVES(1,2)-ELVES(1,1)+ELVES(2,2)-ELVES(2,1))
            BZ=BZ/DELTAP
            BZA(I)=BZ
            X1=X1+DX
            CALL ALT(X1,Y1,Z,SN,AZ,BZ,IASCD)
            CALL QUANT(Z,ICON)
            ELVES(2,1)=Z
            CALL ALT(X1,Y2,Z,SN,AZ,BZ,IASCD)
            CALL QUANT(Z,ICON)
            ELVES(2,2)=Z
      100   CONTINUE
            RETURN
            END
```

```
FORTRAN IV-PLUS V02-51E          13:30:03    21-APR-81            PAGE 3
RCSLP.FTN        /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME        SIZE                ATTRIBUTES

  1      $CODE1  001400    384           RW,I,CON,LCL
  3      $IDATA  000132     45           RW,D,CON,LCL
  4      $VARS   000110     36           RW,D,CON,LCL
  5      $TEMPS  000006      3           RW,D,CON,LCL
  6      BOUNDS  000024     10           RW,D,OVR,GBL

ENTRY POINTS

 NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS

 RCSLP         1-000000

VARIABLES

 NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS

 AZ      R*4   4-000076    BZ      R*4   4-000102    DELTAP  R*4   F-000024*   DLN     R*4   F-000020*   DX      R*4   4-000042
 DX2     R*4   4-000046    DY      R*4   4-000030    ELEVB   R*4   6-000020    FNPTS1  R*4   4-000020    I       I*2   4-000106
 IASCD   I*2   4-000040    ICON    I*2   F-000022*   NPIX    I*2   F-000012*   SN      R*4   4-000034    XEND    R*4   F-000006*
 XLL     R*4   F-000002*   XMAXB   R*4   6-000000    XMINB   R*4   6-000010    X1      R*4   4-000052    X2      R*4   4-000056
 Y       R*4   F-000010*   YCUR    R*4   F-000004*   YEND    R*4   4-000024    YMAXB   R*4   6-000004    YMINB   R*4   6-000014
 Y1      R*4   4-000062    Y2      R*4   4-000066    Z       R*4   4-000072

ARRAYS

 NAME    TYPE  ADDRESS         SIZE         DIMENSIONS

 AZA     R*4   F-000014* 000004      2      (1)
 BZA     R*4   F-000016* 000004      2      (1)
 ELVES   R*4   4-000000  000020      8      (2,2)

LABELS

 LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

 100     1-001350       101     1-001052

FUNCTIONS AND SUBROUTINES REFERENCED

 ALT     QUANT   $SQRT

TOTAL SPACE ALLOCATED = 001674   478

,LP,LST=RCSLP
```

DA

FILM